# SANDIA REPORT

SAND2020-4007 - Unlimited Release

Printed April 9, 2020

Sandia National Laboratories

# SIERRA Low Mach Module: Fuego Theory Manual – Version 4.56

SIERRA Thermal/Fluid Development Team

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185
Livermore, California 94550

Issued by Sandia National Laboratories, operated for the United States Department of Energy by National Technology & Engineering Solutions of Sandia, LLC.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@osti.gov
Online ordering: http://www.osti.gov/scitech

Available to the public from

U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Road
Alexandria, VA 22312

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.gov
Online order: https://classic.ntis.gov/help/order-methods

# ABSTRACT

The SIERRA Low Mach Module: Fuego, henceforth referred to as Fuego, is the key element of the ASC fire environment simulation project. The fire environment simulation project is directed at characterizing both open large-scale pool fires and building enclosure fires. Fuego represents the turbulent, buoyantly-driven incompressible flow, heat transfer, mass transfer, combustion, soot, and absorption coefficient model portion of the simulation software. Using MPMD coupling, Scefire and Nalu handle the participating-media thermal radiation mechanics. This project is an integral part of the SIERRA multi-mechanics software development project. Fuego depends heavily upon the core architecture developments provided by SIERRA for massively parallel computing, solution adaptivity, and mechanics coupling on unstructured grids.

# ACKNOWLEDGMENTS

The following Sandia staff have been involved with the Fuego project, either in code development, math model definitions, or requirements definition: Colin Aro, Amalia Black, Alex Brown, Shawn Burns, Bob Cochran, Stefan Domino, Greg Evans, David Glaze, Lou Gritzo, Michael Hansen, John Hewson, Bill Houf, Robert Knaus, Heeseok Kuo, Mario Martinez, Chris Moen, Elijah Newren, Vern Nicolette, Phil Sakievich, James Sutherland, Warren Tauber, Jeremy Templeton, Sheldon Tieszen, Tyler Voskuilen, and Greg Wagner.

# CONTENTS


**Nomenclature** 13
- English Character Symbols . . . 13
- Greek Character Symbols . . . 15
- Superscript Character Symbols . . . 16
- Subscript Character Symbols . . . 16
- Dimensionless Groups . . . 17

**1. Introduction** 19
- 1.1. Abnormal Thermal Environments . . . 19
- 1.2. Deliverables . . . 20
- 1.3. Document Organization . . . 23

**2. Math Models** 24
- 2.1. Low Mach Number Equations . . . 24
- 2.2. Laminar Flow Equations . . . 28
- 2.3. Radiation Transport Equation . . . 33
- 2.4. Turbulence Modeling Overview . . . 35
- 2.5. Turbulent Flow Equations, Favre-Averaged . . . 39
- 2.6. Turbulence Closure Models . . . 44
- 2.7. Wall Boundary Conditions for Turbulence Models . . . 54
- 2.8. Inlet Conditions for Turbulence Quantities . . . 64
- 2.9. EDC Turbulent Combustion Model . . . 65
- 2.10. Laminar Flamelet Turbulent Combustion Model . . . 88
- 2.11. Turbulent Reacting Mixing Models . . . 107
- 2.12. Soot Generation Model for Multicomponent Combustion . . . 108
- 2.13. Absorptivity Model . . . 117
- 2.14. Fuel Boundary Condition Submodel . . . 121
- 2.15. Fuel Spreading Submodel . . . 124
- 2.16. Volume of Fluid Model . . . 126

**3. MPMD Radiation** 132
- 3.1. Input Deck Conversion . . . 132
- 3.2. Spectral Radiation Transport . . . 137
- 3.3. Particle Radiation Terms . . . 139
- 3.4. Beam (Directed Flux) . . . 140
- 3.5. One-Dimensional Composite Fire Boundary Condition . . . 142
- 3.6. Non-Conformal DG Boundary Condition . . . 148

3.7. Porous-Fluid Coupling Algorithm ..... 150

**4. Particles** **159**
4.1. Introduction ..... 159
4.2. Particle Transport Model ..... 161
4.3. Coupling the Lagrangian and Eulerian Fields ..... 176
4.4. Verification of Particle Evolution Equations ..... 181
4.5. Summary ..... 198
4.6. Evaluating transport coefficients ..... 200
4.7. Lagrangian Particle Capabilities ..... 202

**5. Numerics** **211**
5.1. Flow Solver ..... 212
5.2. Smoothing algorithms defined ..... 216
5.3. Discrete system of equations ..... 220
5.4. Segregated Solution Procedure ..... 229
5.5. Discrete Transport Equations ..... 231
5.6. Discrete Boundary Conditions ..... 253
5.7. Conjugate Heat Transfer ..... 260
5.8. Element Topology and Shape Functions ..... 264
5.9. Interpolation Functions and Negative Coefficients ..... 284
5.10. H-Adaptivity Meshing ..... 293

**6. Implementation** **297**
6.1. SIERRA Frameworks ..... 297
6.2. Fuego Frameworks ..... 299

**7. Transport Processes** **333**
Appendix A. Multicomponent Transport ..... 333
Appendix B. Time-Averaging and Favre-Averaging ..... 334
Appendix C. Discrete 2D/Axisymmetric Transport Equations ..... 335

**8. Review of Control Volume Finite Element Methods** **344**

**9. Turbulence Modeling with v2-f Transport Equations** **352**
Appendix A. Introduction ..... 352
Appendix B. The v2-f Model ..... 354
Appendix C. Test Problem ..... 355
Appendix D. Numerical Implementation Issues and Details ..... 357
Appendix E. Plan ..... 359

**10. Buoyant Vorticity Generation Model** **362**

**11. Proposed Restart Fix for DT Scaling Algorithm** **366**

**12. Virtual Thermocouple Model** **369**
Appendix A. Theoretical Description of the Model . . . . . 369
Appendix B. Model Implementation . . . . . 371

# LIST OF FIGURES


Figure 2.0-1. Fire Math Model Coupling in Fuego . . . . . 25

Figure 2.3-1. Ordinate Direction Definition, $\mathbf{s} = \sin\theta_{zn}\sin\theta_{az}\mathbf{i} + \cos\theta_{zn}\mathbf{j} + \sin\theta_{zn}\cos\theta_{az}\mathbf{k}$ . . . . . 35

Figure 2.9-1. Model geometry for Magnussen's Eddy Dissipation Concept. The control volume is comprised of two zones; the properties of each zone are assumed to be adequately represented by a single set of values (i.e., lumped or perfectly stirred). The mass exchange between the zones is controlled by turbulent mixing. . . . . 66

Figure 2.9-2. Assumed flame surface geometry. $L$ is the integral turbulent length scale. The reaction zone thickness is characterized by the Kolmogorov dissipative turbulent length scale, $\eta$. . . . . 67

Figure 2.13-1. Pentagamma function and asymptotic expansion . . . . . 120

Figure 3.5-1. Representative mesh layout for 1-D composite fire boundary condition . . . . . 143

Figure 3.5-2. Mesh index definition for 1-D composite fire boundary condition . . . . . 146

Figure 4.4-1. The Fuego-computed particle velocity and trajectory are compared with predictions from 4.97 and 4.98 in the left panels. The error in each prediction is shown in the right panel along with the particle Reynolds number. Parameters for the particle evolution are from 4.4-1 with $d_p$ from top row to bottom row being $d_p = 3 \cdot 10^{-2}$, $d_p = 3 \cdot 10^{-3}$, and $d_p = 3 \cdot 10^{-4}$, respectively. In the right-hand panes the solid cures represent the position error, the dashed curves represent the velocity error and the dash-dot-dot curve shows the particle Reynolds number. . . . . 184

Figure 4.4-2. The mean square displacement of particles is shown as a function of time. Different curves show the statistical noise associated with different numbers of particles considered. . . . . 185

Figure 4.4-3. $d^2$ (left) and $T_p$ (right) as a function of time for evaporating water droplets . . . . . 188

Figure 4.4-4. The enthalpy over time in nonreacting channel flow with hot particles injected. The net input enthalpy includes the initial domain enthalpy and the enthalpy of all of the injected particles over time. Squares show the enthalpy associated with particles in the domain. Circles show the enthalpy of the particles that have left the domain. Diamonds show the enthalpy associated with fluid in the domain. Triangles show the excess enthalpy of fluid that has left the domain (difference between the outlet enthalpy and inlet that was accounted for in the net input category). The sum of the categories indicated by symbols is shown to agree with the net input enthalpy indicating overall conservation of enthalpy. . . . . 194

Figure 4.4-5. The mass deposition rate integrated over the outlet is shown as a function of time. The corresponding particle inlet mass flux is 0.1. . . . . 195

Figure 4.4-6. The momentum associated with the particle phase, the fluid phase, and the combined momentum shown. The right-hand panel shows the initial period in greater detail. . . . 196
Figure 4.4-7. Condensed-phase conduction is approximated based on the difference between the film and mean droplet temperatures and on an estimated heat transfer coefficient that describes a boundary layer thickness. Over this boundary layer thickness, the temperature difference $T_f - T_d$ is presumed to act. . . . 199
Figure 4.4-8. Terminal velocities for particles as a function of diameter and particle Reynolds numbers determined from 4.95 and 4.96. . . . 199
Figure 4.4-9. Water droplet evaporation and condensation with initial temperatures set to the wet bulb temperature. Left plot exhibits the linear $d^2$-law behavior while the right hand plot shows the droplet temperatures as constant (no heating). . . . 200
Figure 4.4-10. Aluminum particle evaporation with and without combustion with initial temperatures set to the wet bulb temperature showing the linear $d^2$-law behavior. . . . 201
Figure 4.4-11. The general configuration for verification of two-way coupling. The domain should be of sufficient length that the particles equilibrate with the gas-phase flow. . . . 202
Figure 4.7-1. Particle size (diameter) distribution for Lagrangian particle spray with and without diameter cutoffs set at $0.3$ and $0.65$ . . . 203
Figure 4.7-2. Lagrangian particle spray section of a Fuego input deck showing use of diameter cutoffs . . . 204
Figure 4.7-3. Alumina absorption coefficient for standard models Brewster and Kanopka along with a user-specified model . . . 205
Figure 4.7-4. For a Lagrangian particle spray, the number of particles contained within a parcel for three representative particle diameters using constant number, constant mass, and user defined number of particles per parcel. Circles represent parcels with the points inside representing the number of particles contained in the parcel. . . . 206
Figure 4.7-5. Examples of particle insertion types that can be used with particle insert from file mechanism . . . 208
Figure 4.7-6. Example of particle spread from a conical shaped particle spray nozzle at early times. This nonstandard spray form was generated through the particle creation from file data mechanism. Here particle temperatures are set to be a function of their position with the hottest particles leaving the nozzle near the circular base of the cone. . . . 209
Figure 4.7-7. Same simulation as Fig 4.7-6 but at late time. . . . 210

Figure 5.0-1. Control Volume is Centered about Finite-Element Node . . . 212
Figure 5.3-1. Cell-Peclet number blending function. . . . 223
Figure 5.3-2. Linear profile skew upwind scheme: a) all nodes on the intersected element face are upwind of the subface, b) omit nodes on intersected element face that are downwind of the subface. . . . 224
Figure 5.3-3. Boundary mass flux integration locations. . . . 228
Figure 5.6-1. Integration locations for a wall boundary. . . . 256
Figure 5.8-1. Quadrilateral element topology and numbering . . . 265
Figure 5.8-2. Triangular element topology and numbering . . . 267
Figure 5.8-3. Triangular natural coordinate system, shaded area corresponds to opposite node. . . . 268
Figure 5.8-4. Hexahedral element topology and numbering . . . 270

Figure 5.8-5. Hexahedron subcontrol points numbering . . . . . 274
Figure 5.8-6. Tetrahedral element topology and numbering . . . . . 276
Figure 5.8-7. Tetrahedron subcontrol points numbering . . . . . 278
Figure 5.8-8. Wedge subcontrol points numbering . . . . . 280
Figure 5.8-9. Pyramid subcontrol points numbering . . . . . 283
Figure 5.9-1. Control volume faces in a single element. Contributions to the the control volume centered about node 3. . . . . 285
Figure 5.9-2. Flux integration points (X) determine nodal (•) contributions to the coefficient stencil: a) mid-face rule of CVFEM, b) edge–operator, c) centroid–operator (one-point integration). . . . . 286
Figure 5.9-3. Integration point locations must be shifted out towards the element-edge with increasing element aspect ratio. . . . . 288
Figure 5.9-4. Triangular element geometry in defined by edge length and two vertex angles. . . . . 290
Figure 5.9-5. Limits of Edge-Length Ratio for Positive Coefficients in 3D CVFEM. . . . . 291
Figure 5.10-1. Control volume definition on an h-adapted mesh with hanging nodes. (Four-patch of parent elements with refinement in bottom-right element.) . . . . . 294
Figure 5.10-2. Control volume definition on an h-adapted mesh with transition control volumes about the hanging nodes. (Four-patch of parent elements with refinement in bottom-right element.) . . . . . 295
Figure 5.10-3. Control volume definition on a heterogeneous h-adapted mesh with no hanging nodes. (Four-patch of parent elements with refinement in bottom-right element and splitting in adjacent parent elements.) . . . . . 296

Figure 6.2-1. Fuego Procedure Class . . . . . 305
Figure 6.2-2. Fuego Region Class . . . . . 306
Figure 6.2-3. Fuego Element Mechanics Class . . . . . 306

Figure C-1. Near wall profiles of velocity, Re=13,800 (fuego-0.5.2). . . . . 356
Figure C-2. Profiles of velocity, Re=13,800 (fuego-0.5.2). . . . . 357
Figure C-3. Near wall profiles of turbulent kinetic energy, Re=13,800 (fuego-0.5.2). . . . . 358
Figure C-4. Profiles of turbulent kinetic energy, Re=13,800 (fuego-0.5.2). . . . . 359
Figure C-5. Profiles of turbulent dissipation, Re=13,800 (fuego-0.5.2). . . . . 360
Figure C-6. Near wall profiles of turbulent dissipation, Re=13,800 (fuego-0.5.2). . . . . 360
Figure C-7. Profiles of $v2$ using three meshes, Re=13,800 (fuego-0.5.2). . . . . 361

Figure A-1. Coordinate system, showing the vector $\mathbf{s}$ . . . . . 370

# LIST OF TABLES


Table 2.6-1. Constant parameters for $k-\epsilon$ turbulence models. . . . . 55
Table 2.6-2. Constant parameters for $k-\omega$ turbulence models. . . . . 56
Table 2.6-3. Constant parameters for LES turbulence models. . . . . 56
Table 2.6-4. Constant parameters for miscellaneous turbulence models. Default values may be changed using the $k-\epsilon$ model parameters input. . . . . 57
Table 2.10-1. Chemical mechanism data packaged into the standalone Spitfire executable . . . . . 104
Table 2.12-1. Soot model parameters (Tesner et al.(1971); Holen, et al.(1994)) . . . . . 112
Table 2.13-1. Parameters used in Leckner's gas phase emittance model. . . . . 121
Table 2.13-2. Species-specific parameters used in Equation 2.458. . . . . 122
Table 2.13-3. Coefficients Cij for calculating the scale total emittance of $CO_2$ from Equation 2.459 and Equation 2.460, (valid for $T > 400K$). . . . . 122
Table 2.13-4. Coefficients Cij for calculating the scale total emittance of $H_2O$ from Equation 2.459 and Equation 2.460, (valid for $T > 400K$). . . . . 122

Table 4.2-1. Input parameters related to the particle evolution provided through the input file . . 175
Table 4.2-2. Variables passed from the gas-phase Eulerian solver required to evolve the particles . . 176
Table 4.4-1. Input parameters related to the particle trajectory verification . . . . . 183
Table 4.4-2. Input parameters related to the particle dispersion verification . . . . . 185
Table 4.4-3. Input parameters related to the particle heating and cooling verification . . . . . 187
Table 4.4-4. Input parameters related to the verification of droplet evaporation. . . . . 188
Table 4.4-5. Particle initial conditions for verification of droplet evaporation. . . . . 188
Table 4.4-6. Input parameters related to the verification of energy conservation without evaporation . . . . . 193
Table 4.4-7. Input parameters related to the verification of momentum conservation . . . . . 197
Table 4.4-8. Input parameters related to the verification of energy conservation with evaporation. . . . . 198
Table 4.4-9. Verification sections and the equations tested and validated. . . . . 198

Table 5.8-1. Nodal shape functions and derivatives for quadrilateral elements . . . . . 265
Table 5.8-2. Element variable values and differentials at control-volume faces for quadrilateral elements. Face-to-edge number mapping. . . . . 266
Table 5.8-3. Element variable values and differentials at sub-control volume centers for quadrilateral elements. . . . . 267
Table 5.8-4. Nodal shape functions and derivatives for triangular elements. . . . . 268
Table 5.8-5. Element variable values and differentials at control-volume faces for triangular elements 269
Table 5.8-6. Element variable values and differentials at sub-control volume centers for triangular elements. . . . . 269
Table 5.8-7. Nodal shape functions and derivatives for hexahedral elements. Range is (-1,1). . . . . 271

Table 5.8-8. Element variable values and differentials at control-volume faces for hexahedral elements. Face-to-edge number mapping. . . . . 272
Table 5.8-9. Element variable values and differentials at sub-control volume centers for hexahedral elements. . . . . 273
Table 5.8-10. Subcontrol face definitions for exact surface area calculation on hexahedral elements. 275
Table 5.8-11. Subcontrol volume definitions for exact volume calculation on hexahedral elements. . 275
Table 5.8-12. Nodal shape functions and derivatives for tetrahedral elements. Range is (0,1). . . . . 276
Table 5.8-13. Element variable values and differentials at control-volume faces for tetrahedral elements. Face-to-edge number mapping. . . . . 277
Table 5.8-14. Element variable values and differentials at sub-control volume centers for tetrahedral elements. . . . . 277
Table 5.8-15. Subcontrol face definitions for exact surface area calculation on tetrahedral elements. 278
Table 5.8-16. Subcontrol volume definitions for exact volume calculation on tetrahedral elements. . 278
Table 5.8-17. Nodal shape functions and derivatives for wedge elements. Range is (0,1) and (-1,1). . . 279
Table 5.8-18. Element variable values and differentials at control-volume faces for wedge elements. Face-to-edge number mapping. . . . . 279
Table 5.8-19. Element variable values and differentials at sub-control volume centers for wedge elements. . . . . 280
Table 5.8-20. Subcontrol face definitions for exact surface area calculation on wedge elements. . . . . 281
Table 5.8-21. Subcontrol volume definitions for exact volume calculation on wedge elements. . . . . 281
Table 5.8-22. Nodal shape functions and derivatives for pyramid elements. Range is (-1,1) and (0,1). 281
Table 5.8-23. Element variable values and differentials at control-volume faces for pyramid elements. Face-to-edge number mapping. . . . . 282
Table 5.8-24. Element variable values and differentials at sub-control volume centers for pyramid elements. . . . . 282
Table 5.8-25. Subcontrol face definitions for exact surface area calculation on pyramid elements. . . 282
Table 5.8-26. Subcontrol volume definitions for exact volume calculation on pyramid elements. . . . 283

Table 6.2-1. Fuego Frameworks Classes . . . . . 307
Table 6.2-2. Fuego Procedure Control Data . . . . . 307
Table 6.2-3. Fuego Region Control Data . . . . . 308
Table 6.2-4. Fuego Element Mechanics Instance Control Data . . . . . 311
Table 6.2-5. Fuego BC Mechanics Instance Control Data . . . . . 311
Table 6.2-6. Fuego Sub-Mechanics Definitions . . . . . 311
Table 6.2-7. Fuego Element-Mechanics Workset Definitions . . . . . 312
Table 6.2-8. Fuego Dirichlet Boundary Condition Definitions . . . . . 313
Table 6.2-9. Fuego Master Element Definitions . . . . . 313
Table 6.2-10. User Subroutine Argument Lists . . . . . 313
Table 6.2-11. User Subroutine Signature Type . . . . . 314

# NOMENCLATURE

Einstein notation is used extensively throughout this report to imply summation over repeated indices, primarily for multiple directions in integral equations. Indices are also used to denote chemical species in a gas mixture. When dealing with notation for chemical species, Einstein notation is not implied. When summation over chemical species is required, we will use a summation operator.

## ENGLISH CHARACTER SYMBOLS

$C_p$ mixture specific heat at constant pressure

$D$ mass diffusion coefficient

$D_i$ mixture-averaged mass diffusion coefficient for species $i$

$D_{ij}$ mass diffusion coefficient between species $i$ and $j$ in a mixture

$E$ law of the wall parameter, turbulence model

$f_c$ mass fraction of "excess" carbon in a given species (over what may for $CO_2$ from the available oxygen in the species)

$G$ scalar radiative flux

$g$ magnitude of the gravity vector

$g_i$ component of the gravity vector in the $x_i$ direction

$h$ mixture enthalpy

$h$ fuel pool depth

$\mathrm{j}_{i,g}$ mass diffusion flux vector for species $g$ in the $x_i$ direction

$h_{f_g}$ fuel heat of vaporization

$K$ number of chemical species in a mixture

$k$ mixture thermal conductivity

$k$ turbulent kinetic energy

$L$ length scale

$L$ integral scale with respect to turbulence

$l$ characteristic length scale of the products

$\dot{m}$ mass flow rate

$M$ mass

$N$ concentration of soot particles per volume

$n$ concentration of radical nuclei per volume

$n_i$ unit normal vector component in the $x_i$ direction

$p$ pressure

$P_{th}$ thermodynamic pressure

$q_i$ heat flux vector component in the $x_i$ direction

$\dot{R}$ soot/radical-nuclei particle production/consumption rate per volume in a cell

$R$ universal gas constant

$\dot{r}$ species mass production/consumption rate per unit volume in cell

$r_i$ position vector

$s_i$ unit direction vector for radiation transport

$S$ ratio of air mass fraction to fuel mass fraction

$S_\phi$ source term for scalar variable $\phi$

$t$ time

$T$ temperature

$u_i$ velocity component in the $x_i$ direction

$u$ velocity component in the $x$-direction

$u^\tau$ friction velocity, turbulence model

$u_\parallel$ velocity parallel to the wall, turbulence model

$u^+$ dimensionless velocity, turbulence model

$v$ velocity component in the $y$-direction

$V$ volume of the computation cell (control volume)

$w$ velocity component in the $z$-direction

$W$ mixture molecular weight

$x_i$ Cartesian coordinate direction

$X_s$ mole fraction of species $s$

$Y_s$ mass fraction of species $s$

$y^+$ dimensionless distance from wall, turbulence model

# GREEK CHARACTER SYMBOLS

$\alpha$ absorptivity

$\beta$ concentration of radical nuclei per mixture mass

$\xi_c$ mole fraction of carbon available to produce soot

$\chi$ weighting function for the reacting portion of the fine structure

$\Delta$ scalar difference

$\delta_{ij}$ identity matrix

$\epsilon$ total normal emissivity

$\epsilon$ dissipation of turbulent kinetic energy

$\theta$ spherical direction angle for radiation transport

$\phi$ generic scalar quantity

$\Phi$ equivalence ratio

$\gamma$ volume fraction of turbulent fine structures

$\gamma$ coefficient of surface tension

$\eta$ Kolmogorov dissipative turbulent length scale

$\kappa$ emittance

$\kappa$ thermal conductivity

$\kappa$ von Karman constant, turbulence model

$\lambda$ Taylor turbulent length scale

$\mu$ viscosity

$\nu$ kinematic viscosity

$\rho$ mixture density

$\rho$ reflectivity

$\sigma$ Stefan-Boltzmann constant

$\sigma_{ij}$ deviatoric plus pressure stress tensor

$\tau$ characteristic time scale

$\tau$ transmissivity

$\tau_{ij}$ viscous stress tensor

$\upsilon$ Kolmogorov dissipative turbulent velocity scale

$\zeta$ stoichiometric coefficient

# SUPERSCRIPT CHARACTER SYMBOLS

$n$ iteration or time step number

$r$ indicial notation for reaction number

$'$ fluctuating quantity with respect to time average

$''$ fluctuating quantity with respect to Favre average

$\hat{}$ normalize by stoichiometric values

$\dot{}$ time rate of change of a variable

$\tilde{}$ Favre-averaged quantity

$*$ value for the turbulent fine structure in a cell

$\circ$ value for the surrounding structure in a cell

$\bar{}$ time-averaged quantity

# SUBSCRIPT CHARACTER SYMBOLS

**air** property associated with air

**az** azimuthal angle

**cell** property associated with a control volume

**co** stoichiometric reaction with CO and $H_2$ products

**co2** stoichiometric reaction with $CO_2$ and $H_2O$ products, also a property associated with $CO_2$

**D** property associated with diluents

**flame** property associated with flame zone

**fuel** property associated with fuel

$g$ indicial notation for gas-phase chemical species

**h2o** property associated with $H_2O$

$i$ indicial notation for component of a vector or tensor

**inc** incident quantity

$j$ indicial notation for component of a vector or tensor

$k$ indicial notation for chemical species

**min** minimum limiting value

**mix** mixture property

**n** number of hydrogen atoms in the fuel molecule

**n2** property associated with $N_2$

**oxy** property associated with $O_2$

**p** number of nitrogen atoms in the fuel molecule

**prod** property associated with products

**q** number of oxygen atoms in the fuel molecule

**rad** property associated with radiation

**reac** associated with a specific chemical reaction (??)

**res** fine structure residence

**soot** property associated with soot

**stoich** stoichiometric composition

**surr** property associated with the surroundings

**t** turbulent quantity

**w** wall value

**zn** zenith angle

# DIMENSIONLESS GROUPS

**Pr** Prandtl number, the ratio of viscous and thermal diffusivities

**Re** Reynolds number, the ratio of inertial and viscous forces

**Sc** Schmidt number, the ratio of viscous and mass diffusivities

# 1. INTRODUCTION

The SIERRA Low Mach Module: Fuego, henceforth referred to as Fuego, is the key element of the ASC fire environment simulation project. The fire environment simulation project is directed at characterizing both open large-scale pool fires and building enclosure fires. Fuego represents the turbulent, buoyantly-driven incompressible flow, heat transfer, mass transfer, combustion, soot, and absorption coefficient model portion of the simulation software. Using MPMD coupling, Scefire and Nalu handle the participating-media thermal radiation mechanics. This project is an integral part of the SIERRA multi-mechanics software development project. Fuego depends heavily upon the core architecture developments provided by SIERRA for massively parallel computing, solution adaptivity, and mechanics coupling on unstructured grids.

## 1.1. ABNORMAL THERMAL ENVIRONMENTS

Fuego is part of a suite of numerical simulation tools used to address abnormal thermal environments for nuclear weapon systems [1]. From manufacture to disassembly, a weapon will see three types of environments: normal, hostile, and abnormal. Abnormal environments result from natural phenomena, such as fires, floods, tornadoes, earthquakes, lightning strikes, meteor strikes, etc., and human phenomena, generally classified as “accidents". In general, these phenomena can present thermal, mechanical, and electrical hazards to a weapon system. Nuclear weapon systems must respond to these abnormal environments in a deterministically safe manner.

Fire phenomena in the context of the abnormal thermal environment weapons response issue is part of a three stage process leading from an accident to the system response. For certain scenarios, these stages are uncoupled and may be sequential in time; in others, the stages are tightly coupled and concurrent in time.

The first stage is the initial accident or environmental scenario that is defined typically through probabilistic studies such as historic data involving accident frequencies of a given type, ignition probabilities, etc. These are used to define scenarios for deterministic simulation tools that determine the state of integrity of the weapon system and the distribution of fuel. The weapon integrity is determined by the mechanical, transient-dynamic environment it sees during an accident. For accident scenario description, Fuego is intended to handle the distribution of liquid fuels, although initial implementation will be somewhat limited due to the very broad possibilities (e.g., fuel pools, spills, sprays, porous flows) and complexity involved in two-phase flow.

The second stage is the actual buoyant, turbulent, reacting, flow that is the source of the thermal hazard for the weapon system. Fuego and MPMD-coupled Nalu/Scefire are the primary tools that describe the fire phenomenology that links an accident description to thermal radiation and convection on a weapon

system. Fire involves a very complex, coupled set of physical phenomena over a very broad range of time and length scales. The key features are the turbulent, buoyant flows involving combustion of the fuel and air, and the formation of soot which results in participating media radiation (Nalu/Scefire), and a range of convection heat transfer conditions from free to forced convection (Fuego).

The third stage is the weapon thermal response. As with the fire itself, the response of the warhead to a fire is described by very complex, coupled set of physical phenomena. Simulation will require the coupling of several, separate effects codes for a complete description. Heat from the fire is conducted into the weapon and transmitted by surface-surface radiation. Materials such as foams decompose and result in pressurization. Conduction across engineered joints is pressure dependent as is the decomposition process. Materials such as aluminum can potentially melt and relocate. Energetic materials can decompose and react. Within this environment the engineered fail-safes in the weapon electrical system must operate with high reliability to ensure nuclear safety.

Because of the number of physical phenomena involved from the accident scenario to the weapon response for abnormal thermal environments, and the very disparate time and length scales over which these phenomena occur, it is necessary to have high-performance, massively-parallel, computers to even consider addressing a problem of this scale and complexity. Further, the key to integrating this suite of tools is flexibility of coupling and a common database architecture. Thus it is intended that all the simulation requirements identified above will ride on a common software architecture (SIERRA) with broad coupling flexibilities.

The principal value of the suite of numerical simulation tools is not the description of the accident to response process, but the ability to evaluate prevention and mitigation design strategies. Preventative strategies are primarily applied via administrative controls. Examples include design and maintenance to minimize fuel levels, separation of fuels from air and ignition sources, and/or weapons separate from the combination. Mitigation strategies include suppression (either manually through fire-fighters or by automated fire suppression equipment), design of thermally activated fail-safes, and containment design. In general, multiple barriers exist between fire and health consequences to the general public for nuclear weapons.

## 1.2. DELIVERABLES

The requirements for Fuego Version 1.0 are described in the Strategic Plan [2] and are summarized as follows:

- Customer Applications
  - Weapons Designers
    - Weapons Designers (all phases)
    - Weapons Safety Certification
  - Facilities Safety
    - Prevention Strategies Design/Assessment

        * Mitigation Strategies Design/Assessment
    - Nuclear Safety
        * Weapons Safety Assessments
        * Abnormal Thermal Environments Scenario Assessment
- Scenarios
    - All credible accident scenarios involving fire that can occur from creation to disassembly of any of our nuclear weapon systems.
- Priority Scenarios
    - open hydrocarbon pool fire without wind
    - open hydrocarbon pool fire with wind
    - facility/enclosure with a hydrocarbon fuel fire
- Required Output: Radiative and Convective Heat Flux
    - Resolution requirements
        * length scale: O(0.1 m)
        * time scale: O(10 s)
    - Uncertainty requirements
        * uncertainty estimates are a required part of an analysis
        * range from qualitative analysis to "as low as achievable"
        * tolerance: early phase design > late phase design > certification
- Math Model Requirements
    - Grid-Resolved Models: All Favre-averaged (RANS)
        * mass conservation, variable density
        * species conservation (7 gas equations, 2 soot equations)
        * momentum conservation (3 equations)
        * energy conservation (low Mach number approximation)
        * participating media radiation (number of equations ?)
        * turbulence model (2 equations)
    - Sub-Grid Models
        * wall functions for momentum and heat transfer
        * sub-grid turbulent mixing for combustion, soot, and radiation (EDC)

        * combustion chemistry and thermochemistry (EDC)
        * soot and precursor formation (EDC)
    - Material Models
        * radiative emission/absorption properties
        * transport properties for momentum, energy, and species
        * ideal gas law and thermally perfect thermodynamic properties
    - Fuel Sources
        * liquid hydrocarbon pools
- Computational Requirements
    - Compatibility with SIERRA Frameworks
        * coupled-mechanics (turbulent combustion, participating media radiation, heat conduction)
        * massively parallel
        * distributed memory
        * unstructured grid, O($10^8$) elements
    - Numerical Methods and Solvers
        * proven technology – guaranteed convergence (first-order accurate methods, time and space)
        * 3D, control volume, finite element method (CVFEM)
        * transient (but only for time scales long relative to turbulent fluctuation time scales)
        * flexible coupling between math models (linearization and segregation)
- Problem Solving Environment Requirements
    - Preprocessing for large data sets
    - Diagnostics/Postprocessing for large data sets
        * sensitivity coefficients
        * virtual measurement comparison; i.e., thermocouple
    - Version control
- Verification Requirements
    - Guidelines
    - Truncation error analysis for all operators
    - Regression testing during development

    - Unit testing for major program elements
    - Verification testing to establish correct implementation
- Certification Requirements
    - Analyst training program
    - Review and approval process
- Documentation Requirements
    - Implementation Plans for development
    - Theory Manual
        * math models
        * numerical methods
        * solution strategies
    - User Manual (input syntax and definitions)
    - Verification Suite
        * Truncation error
        * Regression tests
        * Unit tests
        * Verification tests

The following definitions describe the release schedule:

- Fuego $\alpha$ – math models are in place and a fire problem is demonstrated by the development team
- Fuego $\beta$ – code verification is sufficiently complete that the code can be released to a small group of "friendly" users; i.e., analysts working on simulation validation
- Fuego 1.0.0 – code is released with documentation and defect tracking

## 1.3. DOCUMENT ORGANIZATION

This document contains theory and implementation details for the Fuego code. A discussion of the physical models and governing transport equations (math models) is given in Chapter 2. A discussion of the numerical methods that we use to solve the governing transport equations is given in Chapter 5. Implementation details regarding the SIERRA Frameworks are described in Chapter 6. Future math model improvements are discussed in the appendices.

The Einstein notation of repeated indices is used extensively throughout this document. The only exception is for equations involving chemical species where an explicit summation operator is used to imply summation over all chemical species.

# 2. MATH MODELS

Fire simulation requires the solution of variable property, high Grashof number, turbulent, low Mach number flow including the effects of species and soot transport, radiation, and buoyancy. Figure 2.0-1 shows the relation and interconnectivity of the math models as a function of physical conservation law and length scale. Conservation laws include mass of the mixture, momentum, mass of the individual species, and energy. Length scales vary from molecular to convection dominated. For purposes of discussion, length scales are also categorized by the method of resolution.

The transport equations used to describe fire physics are based on two sets of approximations to the fundamental equations of fluid dynamics. Fast acoustic time scales are removed from the equations using low Mach number asymptotics, described in Section 2.1. Turbulent transport at high Grashof numbers is modeled using a Reynolds averaging approach, described in Section 2.4.1.

In what follows, we note that unless specifically stated otherwise all units in the equations and submodel expressions are cgs. For a more extensive treatment of units and unit conversions in Fuego, please see the "Units and Unit Conversions" section in the User's Manual. The numerical methods we use to solve the transport equations are of the finite volume class. Therefore, we generally write the transport equations in the integral form.

## 2.1. LOW MACH NUMBER EQUATIONS

The low Mach number equations are a subset of the full compressible Navier-Stokes (and continuity and energy) equations, admitting large variations in gas density while remaining acoustically incompressible. The low Mach number equations are preferred over the full compressible equations for our problems of interest. We avoid resolving fast-moving acoustic signals which have no bearing on the transport processes. Derivations of the low Mach number equations are found in Rehm and Baum [3], Paolucci [4], Majda and Sethian [5], and Merkle and Choi [6]. The equations are derived from the compressible equations using a perturbation expansion in terms of the lower limit of the Mach number squared; hence the name. The asymptotic expansion leads to a splitting of pressure into a spatially constant thermodynamic pressure and a locally varying dynamic pressure. The dynamic pressure is decoupled from the thermodynamic state and cannot propagate acoustic waves. The thermodynamic pressure is used in the equation of state and to determine thermophysical properties. The thermodynamic pressure can vary in time and can be calculated using a global energy balance.

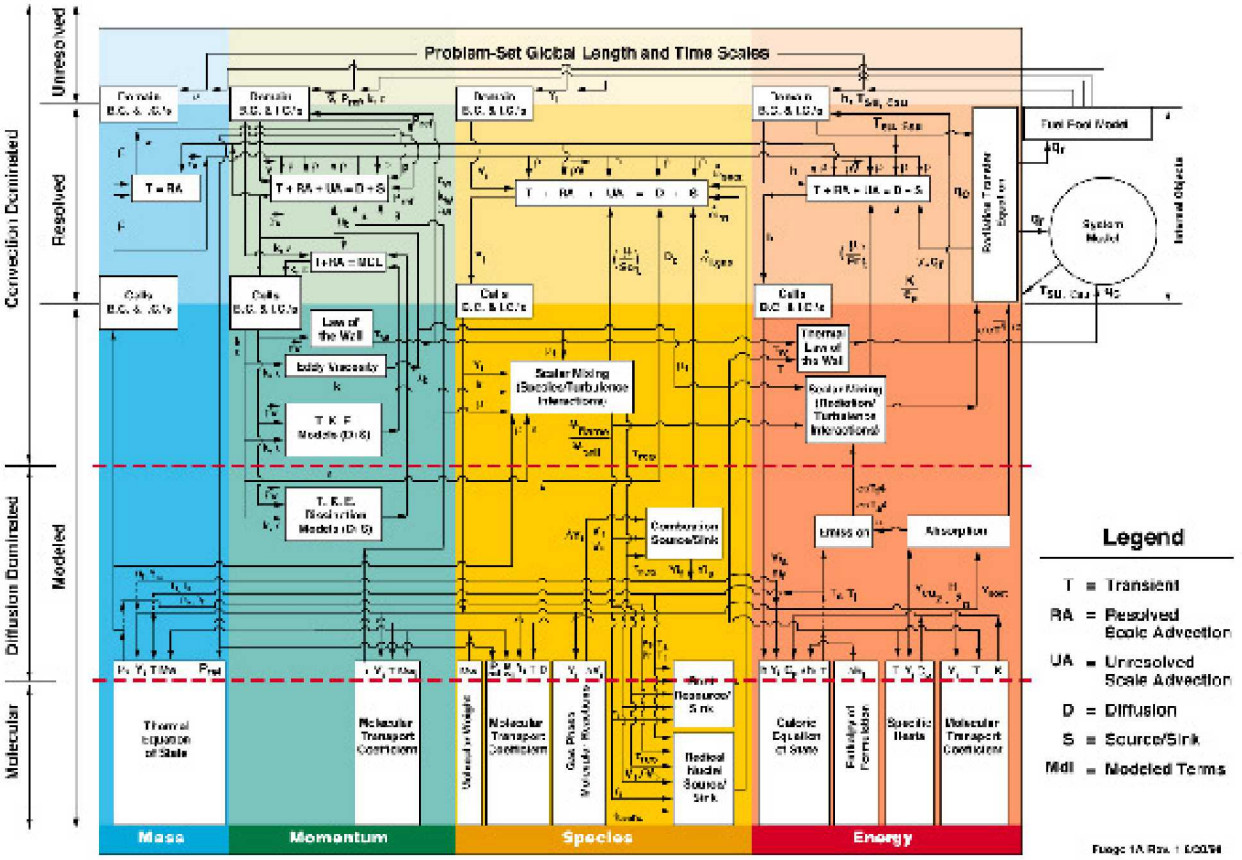


**Figure 2.0-1.. Fire Math Model Coupling in Fuego**

## 2.1.1. Asymptotic Expansion

The asymptotic expansion for the low Mach number equations begins with the full compressible equations in Cartesian coordinates. The equations are the minimum set required to propagate acoustic waves. The equations are written in divergence form using Einstein notation (summation over repeated indices):

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_j}{\partial x_j} = 0, \tag{2.1}$$

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_j u_i}{\partial x_j} + \frac{\partial P}{\partial x_i} = \frac{\partial \tau_{ij}}{\partial x_j} + \rho g_i, \tag{2.2}$$

$$\frac{\partial \rho E}{\partial t} + \frac{\partial \rho u_j H}{\partial x_j} = -\frac{\partial q_j}{\partial x_j} + \frac{\partial u_i \tau_{ij}}{\partial x_j} + \rho u_i g_i. \tag{2.3}$$

The primitive variables are the velocity components, $u_i$, the pressure, $P$, and the temperature $T$. The viscous shear stress tensor is $\tau_{ij}$, the heat conduction is $q_i$, the total enthalpy is $H$, the total internal energy is $E$, the density is $\rho$, and the gravity vector is $g_i$. The total internal energy and total enthalpy contain the kinetic energy contributions. The equations are closed using the following models and

definitions:

$$
\begin{aligned}
P &= \rho \frac{R}{W} T, && (2.4)\\
E &= H - P/\rho, && (2.5)\\
H &= h + \frac{1}{2} u_k u_k, && (2.6)\\
\tau_{ij} &= \mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \mu \frac{\partial u_k}{\partial x_k} \delta_{ij}, && (2.7)\\
q_i &= -k \frac{\partial T}{\partial x_i}. && (2.8)
\end{aligned}
$$

The mean molecular weight of the gas is $W$, the molecular viscosity is $\mu$, and the thermal conductivity is $k$. A Newtonian fluid is assumed along with the Stokes hypothesis for the stress tensor.

The equations are scaled so that the variables are all of order one. The velocities, lengths, and times are nondimensionalized by a characteristic velocity, $U_\infty$, and a length scale, $L$. The pressure, density, and temperature are nondimensionalized by $P_\infty$, $\rho_\infty$, and $T_\infty$. The enthalpy and energy are nondimensionalized by $C_{p,\infty} T_\infty$. Dimensionless variables are noted by overbars. The dimensionless equations are:

$$
\begin{aligned}
\frac{\partial \bar{\rho}}{\partial \bar{t}} + \frac{\partial \bar{\rho} \bar{u}_j}{\partial \bar{x}_j} &= 0, && (2.9)\\
\frac{\partial \bar{\rho} \bar{u}_i}{\partial \bar{t}} + \frac{\partial \bar{\rho} \bar{u}_j \bar{u}_i}{\partial \bar{x}_j} + \frac{1}{\gamma \mathrm{Ma}^2} \frac{\partial \bar{P}}{\partial \bar{x}_i} &= \frac{1}{\mathrm{Re}} \frac{\partial \bar{\tau}_{ij}}{\partial \bar{x}_j} + \frac{1}{\mathrm{Fr}_i} \bar{\rho}, && (2.10)\\
\frac{\partial \bar{\rho} \bar{h}}{\partial \bar{t}} + \frac{\partial \bar{\rho} \bar{u}_j \bar{h}}{\partial \bar{x}_j} &= -\frac{1}{\mathrm{Pr}} \frac{1}{\mathrm{Re}} \frac{\partial \bar{q}_j}{\partial \bar{x}_j} + \frac{\gamma - 1}{\gamma} \frac{\partial \bar{P}}{\partial \bar{t}} && (2.11)\\
&+ \frac{\gamma - 1}{\gamma} \frac{\mathrm{Ma}^2}{\mathrm{Re}} \frac{\partial \bar{u}_i \bar{\tau}_{ij}}{\partial \bar{x}_j} + \bar{\rho} \bar{u}_i \frac{\gamma - 1}{\gamma} \frac{\mathrm{Ma}^2}{\mathrm{Fr}_i}\\
&- \frac{\gamma - 1}{2} \mathrm{Ma}^2 \left( \frac{\partial \bar{\rho} \bar{u}_k \bar{u}_k}{\partial \bar{t}} + \frac{\partial \bar{\rho} \bar{u}_j \bar{u}_k \bar{u}_k}{\partial \bar{x}_j} \right).
\end{aligned}
$$

The groupings of characteristic scaling terms are:

$$
\begin{aligned}
\mathrm{Re} &= \frac{\rho_\infty U_\infty L}{\mu_\infty}, && \text{Reynolds number,} && (2.12)\\
\mathrm{Pr} &= \frac{C_{p,\infty} \mu_\infty}{k_\infty}, && \text{Prandtl number,} && (2.13)\\
\mathrm{Fr}_i &= \frac{u_\infty^2}{g_i L}, && \text{Froude number,} \quad g_i \neq 0, && (2.14)\\
\mathrm{Ma} &= \sqrt{\frac{u_\infty^2}{\gamma R T_\infty / W}}, && \text{Mach number,} && (2.15)
\end{aligned}
$$

where $\gamma$ is the ratio of specific heats.

For small Mach numbers, $\mathrm{Ma} \ll 1$, the kinetic energy, viscous work, and gravity work terms can be neglected in the energy equation since those terms are scaled by the square of the Mach number. The inverse of Mach number squared remains in the momentum equations, suggesting singular behavior. In order to explore the singularity, the pressure, velocity and temperature are expanded as asymptotic series in terms of the parameter $\epsilon$:

$$\bar{P} = \bar{P}_0 + \bar{P}_1\epsilon + \bar{P}_2\epsilon^2 \ldots \tag{2.16}$$

$$\bar{u}_i = \bar{u}_{i,0} + \bar{u}_{i,1}\epsilon + \bar{u}_{i,2}\epsilon^2 \ldots \tag{2.17}$$

$$\bar{T} = \bar{T}_0 + \bar{T}_1\epsilon + \bar{T}_2\epsilon^2 \ldots \tag{2.18}$$

The zeroeth-order terms are collected together in each of the equations. The form of the continuity equation stays the same. The gradient of the pressure in the zeroeth-order momentum equations can become singular since it is divided by the characteristic Mach number squared. In order for the zeroeth-order momentum equations to remain well-behaved, the spatial variation of the $\bar{P}_0$ term must be zero. If the magnitude of the expansion parameter is selected to be proportional to the square of the characteristic Mach number, $\epsilon = \gamma \mathrm{Ma}^2$, then the $\bar{P}_1$ term can be included in the zeroeth-order momentum equation.

$$\frac{1}{\gamma \mathrm{Ma}^2}\frac{\partial \bar{P}}{\partial x_i} = \frac{\partial}{\partial x_i}\left(\frac{1}{\gamma \mathrm{Ma}^2}\bar{P}_0 + \frac{\epsilon}{\gamma \mathrm{Ma}^2}\bar{P}_1 + \ldots\right) = \frac{\partial}{\partial x_i}\left(\bar{P}_1 + \epsilon\bar{P}_2 + \ldots \quad\right) \tag{2.19}$$

The form of the energy equation remains the same, less the kinetic energy, viscous work and gravity work terms. The $P_0$ term remains in the energy equation as a time derivative. The low Mach number equations are the zeroeth-order equations in the expansion including the $P_1$ term in the momentum equations. The expansion results in two different types of pressure and they are considered to be split into a thermodynamic component and a dynamic component. The thermodynamic pressure is constant in space, but can change in time. The thermodynamic pressure is used in the equation of state. The dynamic pressure only arises as a gradient term in the momentum equation and acts to enforce continuity. The unsplit dimensional pressure is

$$P = P_{th} + \gamma \mathrm{Ma}^2 P_1, \tag{2.20}$$

where the dynamic pressure, $p = P - P_{th}$, is related to a pressure coefficient

$$\bar{P}_1 = \frac{P - P_{th}}{\rho_\infty u_\infty^2} P_{th}. \tag{2.21}$$

The resulting unscaled low Mach number equations are:

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_j}{\partial x_j} = 0, \tag{2.22}$$

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_j u_i}{\partial x_j} + \frac{\partial p}{\partial x_i} = \frac{\partial \tau_{ij}}{\partial x_j} + (\rho - \rho_\circ)\, g_i, \tag{2.23}$$

$$\frac{\partial \rho h}{\partial t} + \frac{\partial \rho u_j h}{\partial x_j} = -\frac{\partial q_j}{\partial x_j} + \frac{\partial P_{th}}{\partial t}, \tag{2.24}$$

where the ideal gas law becomes

$$P_{th} = \rho \frac{R}{W} T. \tag{2.25}$$

The hydrostatic pressure gradient has been subtracted from the momentum equation, assuming an ambient density of $\rho_\circ$. The stress tensor and heat conduction remain the same as in the original equations.

### 2.1.2. Variable Thermodynamic Pressure

For a low Mach number set of equations, the time derivative of pressure can only be nonzero in a closed volume with energy addition or subtraction. Relaxing the low Mach number limit allows a time and spatially varying pressure to appear in the energy equation (see Section 2.2.3).

## 2.2. LAMINAR FLOW EQUATIONS

Laminar transport equations are not used for fire problems, but they are important for other classes of problems such as manufacturing. The low Mach number approximation is assumed (see Section 2.1).

### 2.2.1. Conservation of Mass

The mass conservation equation of a mixture of gases is given by

$$\int \frac{\partial \rho}{\partial t} \mathrm{d}V + \int \rho u_j n_j \mathrm{d}S = 0, \tag{2.26}$$

where $u_j$ is the mass average velocity of the mixture [7].

### 2.2.2. Conservation of Momentum

The conservation of momentum equations are given by

$$\int \frac{\partial \rho u_i}{\partial t} \mathrm{d}V + \int \rho u_i u_j n_j \mathrm{d}S + \int P n_i \mathrm{d}S = \int \tau_{ij} n_j \mathrm{d}S + \int \left(\rho - \rho_\circ\right) g_i \mathrm{d}V, \tag{2.27}$$

where the viscous stress tensor is

$$\tau_{ij} = \mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \mu \frac{\partial u_k}{\partial x_k} \delta_{ij}. \tag{2.28}$$

The pressure, $P$, in the momentum equations deserves a special note as this quantity can represent either the dynamic, i.e., the second term in the Mach number expansion in the case of the low Mach number assumption, or the static pressure in the case of formally compressibility. In either case, as shown above the hydrostatic pressure gradient has been removed which gives rise to the far-field density, $\rho_\circ$, in the buoyancy body force. Optionally, we allow for the following sets of buoyancy models:

1) a Boussinesq buoyancy approximation where the density difference is approximated as

$$(\rho - \rho_\circ) \approx -\frac{\rho_\circ}{T_\circ}(T - T_\circ), \tag{2.29}$$

2) a standard buoyant model in which case the pressure above does include the hydrostatic pressure and the buoyancy right-hand-side source term is,

$$\rho g_i, \tag{2.30}$$

3) A Boussinesq approximation for a binary mixture in which case the right-hand-side contribution is:

$$\rho MW^{ref}\left(\frac{1}{MW_1} - \frac{1}{MW_2}\right)\left[Y_1 - Y^{ref}\right] g_i. \tag{2.31}$$

The user is referred to the Fuego user manual for exact line commands for each of these buoyancy options.

Note that zero pressure is almost always a convenient initial condition for a low Mach fluid flow. However, in cases without buoyancy, it can be anything, as the value only defines the additive constant for the pressure solve. However, one must ensure that the value matches for both initial and boundary condition specifications.

For buoyant flow, specifying zero pressure is convenient in tandem with the "differential" buoyancy option. This buoyancy term subtracts off the hydrostatic contribution such that the source term is written as

$$\rho\left(\rho - \rho_{ref}\right) \tag{2.32}$$

One can see that using this term along with a zero pressure initial condition allows one to avoid specifying initial and boundary conditions as the hydrostatic pressure, i.e., as a function of height.

### 2.2.3. Conservation of Energy

The conservation of energy equation in terms of enthalpy (including a source term due to radiation absorption and emission) is

$$\begin{aligned}\int \frac{\partial \rho h}{\partial t}\mathrm{d}V + \int \rho h u_j n_j \mathrm{d}S &= -\int q_j n_j \mathrm{d}S - \int \frac{\partial q_i^r}{\partial x_i}\mathrm{d}V \\ &+ \int\left(\frac{\partial P}{\partial t} + u_j\frac{\partial P}{\partial x_j}\right)\mathrm{d}V + \int \tau_{ij}\frac{\partial u_i}{\partial x_j}\mathrm{d}V,\end{aligned} \tag{2.33}$$

where the energy diffusion flux vector is given by

$$q_j = -\kappa\frac{\partial T}{\partial x_j} + \sum_{k=1}^{K}\rho h_k Y_k \hat{u}_{j,k}, \tag{2.34}$$

and $\hat{u}_{j,k}$ is the diffusion velocity of species $k$ in the $j$ direction. This form of the energy equation is derived by starting with the energy equation and supplemental relationships of internal energy and

total enthalpy provided in Section 2.1.1. The time term and convection term due to kinetic energy are expanded using the chain rule and simplified by enforcing the continuity equation. The remaining kinetic energy terms and gravitational force term are removed by dotting velocity with the momentum equation (to obtain the mechanical energy equation) and subtracting it from the energy equation. This procedure provides the full material derivative of pressure and the expanded viscous dissipation term. The last two terms of Equation 2.33 are only active when formal compressibility (in an acoustic sense) are important (see the Fuego user manual for the appropriate command lines to activate the low speed compressible and high speed compressible form in Fuego).

For a low Mach number flow, the time derivative of the pressure appearing above is substituted by the thermodynamic reference pressure, $P_{th}$, that can only be nonzero in a closed volume with energy addition or subtraction. However, the low Mach number approximation mandates that the thermodynamic pressure is always spatially uniform.

The enthalpy of the mixture, $h$, is a mass-average of the component enthalpies, $h_k$, given by

$$h = \sum_{k=1}^{K} h_k Y_k. \tag{2.35}$$

The energy diffusion flux vector includes a scaled gradient of temperature whereas the independent field to be solved in Equation 2.33 is enthalpy. The form of the gradient of temperature is derived by first taking the gradient of Equation 2.35 and using the chain rule,

$$\frac{\partial h}{\partial x_j} = \sum_{k=1}^{K} Y_k \frac{\partial h_k}{\partial x_j} + \sum_{k=1}^{K} h_k \frac{\partial Y_k}{\partial x_j}. \tag{2.36}$$

Given the thermodynamic definition of specific heat, the above equation is given by,

$$\frac{\partial h}{\partial x_j} = \sum_{k=1}^{K} Y_k C_{p_k} \frac{\partial T}{\partial x_j} + \sum_{k=1}^{K} h_k \frac{\partial Y_k}{\partial x_j} \tag{2.37}$$

$$= C_p \frac{\partial T}{\partial x_j} + \sum_{k=1}^{K} h_k \frac{\partial Y_k}{\partial x_j}. \tag{2.38}$$

This equation is rearranged,

$$\frac{\partial T}{\partial x_j} = \frac{1}{C_p} \left( \frac{\partial h}{\partial x_j} - \sum_{k=1}^{K} h_k \frac{\partial Y_k}{\partial x_j} \right), \tag{2.39}$$

and substituted into the energy diffusion flux vector to obtain,

$$q_j = -\frac{\kappa}{C_p} \left( \frac{\partial h}{\partial x_j} - \sum_{k=1}^{K} h_k \frac{\partial Y_k}{\partial x_j} \right) + \sum_{k=1}^{K} \rho h_k Y_k \hat{u}_{j,k}. \tag{2.40}$$

Commonly, the last two terms in the above equation can be canceled when a simple diffusion model is assumed (see Section 2.2.4, Equation 2.46) in the limit where the ratio of thermal and mass diffusion is

equal (unity Lewis number, or equivalently speaking the Prandtl number equals the Schmidt number, i.e.,

$$Le^{unity} = \frac{Sc}{Pr} = \frac{\alpha}{D} = 1. \tag{2.41}$$

For completeness, the thermal diffusivity, Prandtl and Schmidt number are defined by,

$$\alpha = \frac{\kappa}{\rho C_p}, \tag{2.42}$$

$$Pr = \frac{C_p \mu}{\kappa} = \frac{\mu}{\rho \alpha}. \tag{2.43}$$

and

$$Sc = \frac{\mu}{\rho D_{ab}}. \tag{2.44}$$

### 2.2.4. Conservation of Species

The mass conservation equation for species $k$ in a mixture of $K$ gas phase species is

$$\int \frac{\partial \rho Y_k}{\partial t} \mathrm{d}V + \int \rho Y_k u_j n_j \mathrm{d}S = -\int \rho \hat{u}_{j,k} Y_k n_j \mathrm{d}S + \int \dot{\omega}_k \mathrm{d}V, \tag{2.45}$$

where $\dot{\omega}_k$ is the mass generation rate of species $k$ per unit volume by homogeneous chemical reactions. We allow several approximations for the diffusion velocity, $\hat{u}_{j,k}$, derived in Appendix 7. The simplest form is Fickian diffusion with the same value of mass diffusivity for all species,

$$\hat{u}_{i,k} = -D \frac{1}{Y_k} \frac{\partial Y_k}{\partial x_i}. \tag{2.46}$$

This form is used for the Reynolds-averaged form of the equations for turbulent flow. A more accurate approximation uses a mixture-averaged diffusion coefficient, $\bar{D}_k$, for each species diffusion velocity,

$$\hat{u}_{i,k} = -\bar{D}_k \frac{1}{X_k} \frac{\partial X_k}{\partial x_i} = -\bar{D}_k \left( \frac{1}{Y_k} \frac{\partial Y_k}{\partial x_i} + \frac{1}{W} \frac{\partial W}{\partial x_i} \right). \tag{2.47}$$

### 2.2.5. Conservation of Momentum, Axisymmetric with Swirl

Axisymmetric flows, with or without swirl, are described by two-dimensional equations in cylindrical coordinates. All azimuthal derivatives are zero (i.e., $\partial/\partial\theta = 0$). The axial coordinate is $x$, the radial coordinate is $r$, and the azimuthal coordinate is $\theta$. The radius is retained in the equations and the purpose will become more clear in the discussion of the discrete integral form. The axial velocity is $u$, the radial velocity is $v$, and the azimuthal velocity is $w$.

Axial-Momentum:

$$\frac{\partial \rho u r}{\partial t} + \frac{\partial}{\partial x}\left(\rho u^2 r\right) + \frac{\partial}{\partial r}\left(\rho u v r\right) + r\frac{\partial p}{\partial x} = \frac{\partial}{\partial x}\left(r\tau_{xx}\right) + \frac{\partial}{\partial r}\left(r\tau_{xr}\right) + \rho r g_x \tag{2.48}$$

Radial-Momentum:

$$\frac{\partial \rho v r}{\partial t} + \frac{\partial}{\partial x}\left(\rho u v r\right) + \frac{\partial}{\partial r}\left(\rho v^2 r\right) + r\frac{\partial p}{\partial r} - \rho w^2 = \frac{\partial}{\partial x}\left(r\tau_{rx}\right) + \frac{\partial}{\partial r}\left(r\tau_{rr}\right) - \tau_{\theta\theta} + \rho r g_r \tag{2.49}$$

Azimuthal-Momentum:

$$\frac{\partial \rho w r}{\partial t} + \frac{\partial}{\partial x}\left(\rho u w r\right) + \frac{\partial}{\partial r}\left(\rho v w r\right) + \rho v w = \frac{\partial}{\partial x}\left(r\tau_{\theta x}\right) + \frac{1}{r}\frac{\partial}{\partial r}\left(r^2\tau_{\theta r}\right) \tag{2.50}$$

The viscous stress terms for the cylindrical equations are

$$\tau_{xx} = \mu\left[2\frac{\partial u}{\partial x} - \frac{2}{3}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial r} + \frac{v}{r}\right)\right] \tag{2.51}$$

$$\tau_{rx} = \mu\left[\frac{\partial v}{\partial x} + \frac{\partial u}{\partial r}\right] \tag{2.52}$$

$$\tau_{rr} = \mu\left[2\frac{\partial v}{\partial r} - \frac{2}{3}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial r} + \frac{v}{r}\right)\right] \tag{2.53}$$

$$\tau_{\theta\theta} = \mu\left[2\frac{v}{r} - \frac{2}{3}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial r} + \frac{v}{r}\right)\right] \tag{2.54}$$

$$\tau_{r\theta} = \mu r\frac{\partial}{\partial r}\left(\frac{w}{r}\right) \tag{2.55}$$

$$\tau_{x\theta} = \mu\frac{\partial w}{\partial x} \tag{2.56}$$

The azimuthal equation can be simplified by relating the swirl velocity to the angular velocity, $w = r\omega$. The momentum equation, written in terms of the angular velocity, is

$$\frac{\partial \rho \omega r}{\partial t} + \frac{\partial}{\partial x}\left(\rho u \omega r\right) + \frac{\partial}{\partial r}\left(\rho v \omega r\right) + 2\rho v\omega = \frac{\partial}{\partial x}\left(r\mu\frac{\partial \omega}{\partial x}\right) + \frac{\partial}{\partial r}\left(r\mu\frac{\partial \omega}{\partial r}\right) + 2\mu\frac{\partial \omega}{\partial r}. \tag{2.57}$$

The production term that is used in the turbulence model is

$$\Phi = 2\left[\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial r}\right)^2 + \left(\frac{v}{r}\right)^2\right] + \left(\frac{\partial u}{\partial r} + \frac{\partial v}{\partial x}\right)^2 - \frac{2}{3}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial r} + \frac{v}{r}\right)^2. \tag{2.58}$$

### 2.2.6. Laminar Flow Boundary Conditions

The laminar flow math models require boundary conditions for velocity, pressure, temperature and enthalpy variables, and mixture composition.

#### 2.2.6.1. Inflow

There are three types of inflow boundary conditions. For velocity-specified inflow, Dirichlet conditions are applied to velocities in the momentum equations, temperature in the energy equation, and mass fractions in the species equations. The mass flow rate at the boundary is specified for the continuity equation. The pressure floats to a consistent value. Alternatively, a control volume balance is retained at the boundary nodes and the convection fluxes are specified.

For pressure-specified inflow, the outflow boundary condition is applied with the added condition that the flow must enter the domain normal to the mesh boundary. Transport equations are solved for the momentum, energy and species equations.

#### 2.2.6.2. Outflow

The pressure is specified at integration points on the outflow boundary. The specified pressure is used in the surface integration procedure for approximation nodal gradients. The pressure gradients are used to construct an interpolation for the mass flow rate at the boundary. Transport equations are solved for the momentum, energy and species equations. Upwind extrapolation is used for the scalars if the flow is leaving the domain. The boundary values of velocity and specified far-field values of scalars are used if the flow is entering the domain.

#### 2.2.6.3. Wall

It is assumed that there is no mass flow through the wall. The velocity is specified as a Dirichlet boundary condition in the momentum equations. The temperature is specified as a Dirichlet boundary condition in the energy if the wall is isothermal. We currently do not support heterogeneous chemical reactions at a surface, so there should be no boundary condition applied to the mass fractions.

#### 2.2.6.4. Symmetry Plane

There is no mass flow rate through the symmetry plane and there is no transport of scalar variables. The normal stress (pressure and viscous) at the symmetry plane is applied in the momentum equations.

## 2.3. RADIATION TRANSPORT EQUATION

For applications involving PMR, both the radiative heat flux and the divergence of the radiative heat flux are needed. The radiative heat flux vector provides the radiative flux to the boundary of the heat conduction region. The flux divergence provides one of the principal volumetric heat sources in the turbulent combustion region for fire applications.

### 2.3.1. Boltzmann Transport Equation

The spatial variation of the radiative intensity corresponding to a given direction and at a given wavelength within a radiatively participating material, $I(s)$, is governed by the Boltzmann transport equation. In general, the Boltzmann equation represents a balance between absorption, emission, out-scattering, and in-scattering of radiation at a point. For combustion applications, however, the steady form of the Boltzmann equation is appropriate since the transient term only becomes important on nanosecond time scales which is orders of magnitude shorter than the fastest chemical reaction [8].

Experimental data shows that the radiative properties for heavily sooting, fuel-rich hydrocarbon diffusion flames ($10^{-4}$% to $10^{-6}$% soot by volume) are dominated by the soot phase and to a lesser extent by the gas phase (Modest [9], pg. 425). Since soot emits and absorbs radiation in a relatively constant spectrum, it is common to ignore wavelength effects when modeling radiative transport in these environments. Additionally, scattering from soot particles commonly generated by hydrocarbon flames is several orders of magnitude smaller that the absorption effect and may be neglected [8]. With these assumptions in mind, the appropriate form of the Boltzmann radiative transport equation for heavily sooting hydrocarbon diffusion flames is

$$s_i \frac{\partial}{\partial x_i} I\left(s\right) + \mu_a I\left(s\right) = \frac{\mu_a \sigma T^4}{\pi}, \tag{2.59}$$

where $\mu_a$ is the absorption coefficient, $I(s)$ is the intensity along the direction $s_i$, and $T$ is the temperature.

The flux divergence (the last term on the right hand side of Equation 2.33) may be written as a difference between the radiative emission and mean incident radiation at a point,

$$\frac{\partial q_i^r}{\partial x_i} = \mu_a \left[4\sigma T^4 - G\right], \tag{2.60}$$

where $G$ is the scalar flux. The quantity, $G/4\pi$, is often referred to as the mean incident intensity [10].

The scalar flux and radiative flux vector represent angular moments of the directional radiative intensity at a point [9],

$$G = \int_0^{2\pi} \int_0^{\pi} I\left(s\right) \sin\theta_{zn} d\theta_{zn} d\theta_{az}, \tag{2.61}$$

$$q_i^r = \int_0^{2\pi} \int_0^{\pi} I\left(s\right) s_i \sin\theta_{zn} d\theta_{zn} d\theta_{az}, \tag{2.62}$$

where $\theta_{zn}$ and $\theta_{az}$ are the zenith and azimuthal angles respectively as shown in Figure 2.3-1.

### 2.3.2. Radiation Intensity Boundary Condition

The radiation intensity must be defined at all portions of the boundary along which $s_i n_i < 0$, where $n_i$ is the outward directed unit normal vector at the surface. The intensity is applied as a Dirichlet condition which must be determined from the surface properties and temperature. The diffuse surface

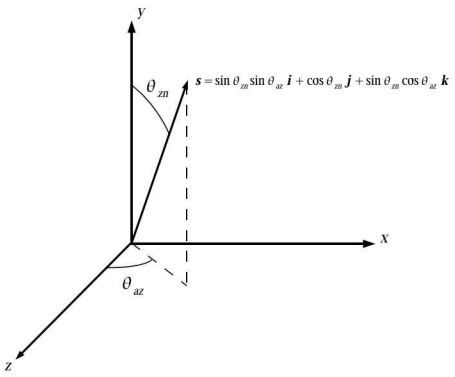


**Figure 2.3-1.. Ordinate Direction Definition,**
$\mathbf{s} = \sin\theta_{zn}\sin\theta_{az}\mathbf{i} + \cos\theta_{zn}\mathbf{j} + \sin\theta_{zn}\cos\theta_{az}\mathbf{k}$

assumption provides reasonable accuracy for many engineering combustion applications. The intensity leaving a diffuse surface in all directions is given by

$$I(s) = \frac{1}{\pi}\left[\tau\sigma T_{\infty}^{4} + \epsilon\sigma T_{w}^{4} + (1-\epsilon-\tau)\, q_{j}^{r,inc} n_{j}\right], \tag{2.63}$$

where $\epsilon$ is the total normal emissivity of the surface, $\tau$ is the transmissivity of the surface, $T_w$ is the temperature of the boundary, $T_\infty$ is the environmental temperature and $q_j^{r,inc}$ is the incident radiation, or irradiation for direction $j$. Recall that the relationship given by Kirchhoff's Law that relates emissivity, transmissivity and reflectivity, $\rho$, is

$$\rho + \tau + \epsilon = 1. \tag{2.64}$$

where it is implied that $\alpha = \epsilon$.

## 2.4. TURBULENCE MODELING OVERVIEW

Turbulent reacting flows involve a very large range of length and time scales, requiring massive computational resources to directly resolve all of the physical processes for even the most simple problem. To be able to solve complex problems of interest in a reasonable amount of time, modeling approximations must be made. A filtered form of the time-dependent Navier-Stokes, energy, and species mass conservation equations presented in Section 2.2 are used, and closure models are applied to the new terms that arise due to the filtering operation. Temporal filtering is used in the Reynolds-Averaged Navier-Stokes (RANS) method, and spatial filtering is used in the Large Eddy Simulation (LES) method. The form of the models are dependent on the type of filtering performed, and will be discussed for both the RANS and LES approaches in the following sections.

Figure 2.0-1 schematically illustrates the interaction between all of the transport equations across the full range of length scales. The transport equations are shown in shorthand with the notation T, RA, UA,

D, S being the transient term, the resolved advection term, the unresolved advection (Reynolds stresses) term, diffusion term and source term, respectively. Only one transport equation is shown for each conservation principle, but it is understood that three equations exist for momentum (u,v,w), and an equation exists for each species being transported (seven in the present model plus two for soot). The momentum transport equations are strongly interconnected while the species equations are coupled implicitly through their source terms, thermophysical properties, and conservation of mass of the mixture.

The length scales in Figure 2.0-1 between the smallest control volume dimension and the largest mesh dimension are defined as being "resolved", and the transport equations are used to solve the physics in this range. The effects of the resolved turbulent scales may be modeled for RANS closures or they may be directly solved for LES closures. Turbulence length scales can extend down many orders of magnitude beyond the smallest finite volume dimension to the Kolmogorov scales, and these subgrid scales must be modeled in either closure approach.

The output of the closure models is expressed as a source term in the conservation equations for the mean flow and as effective properties in the radiative transport equation. Hence, the output of the closure models can be interpreted as being cell-averaged values for the control volume for the appropriate time scale. For the RANS formulation used here, the time scale is long relative to the turbulence time scales (i.e., long time average). For LES, the time scale is the local advection time. For the current suite of models, the momentum closure model is of the lumped-parameter type; that is, it assumes homogeneity of the subgrid turbulence. The remaining closures, species and energy, are of the zone-model type; that is, they assume heterogeneity of the species and energy subgrid. Two zones (one combusting, one not) are used in the current zone models.

For length scales above the length scale of the mesh, the physics is modified via boundary and initial conditions. Momentum boundary conditions include specified velocity (wind, and mass sources), or constant pressure (inflow/outflow). Species boundary conditions include a mass source for the fuel (pool model). Thermal boundary conditions include flux and temperature conditions. The following sections provide details of the math models for conservation laws and fire physics models used in SIERRA/Fuego.

### 2.4.1. RANS Temporal Filtering

In many typical engineering applications, only time averages of physical quantities are of interest. Often, details of the turbulent fluctuations are of little concern. RANS formulations address this need by solving a temporally-filtered form of the transport equations, directly yielding the time-averaged variables of interest. For this reason, RANS approaches represent a relatively low-cost solution method at the expense of additional modeling complexity.

An independent variable $\phi$ can be temporally filtered to obtain its mean $\bar{\phi}$ with the mathematical form (Tennekes and Lumley [11])

$$\overline{\phi(\boldsymbol{x})} = \lim_{\tau\to\infty} \frac{1}{\tau} \int_{t_o}^{t_o+\tau} \phi(\boldsymbol{x}, t)\, \mathrm{d}t. \tag{2.65}$$

The original variable can be represented as the sum of its mean and fluctuating component, $\phi = \bar{\phi} + \phi'$, with the properties that $\bar{\bar{\phi}} = \bar{\phi}$ and $\overline{\phi'} = 0$. This is called the Reynolds decomposition of a variable.

In combustion problems, the overall exothermic process can result in large localized temperature increases and a correspondingly large density decrease in open systems where the molecular weight change from reactants to products is small. Allowing for turbulent fluctuations of density, the above temporal averaging procedure gives rise to additional terms involving time averages of products of density and other variable (e.g., velocity) fluctuations. An alternative approach to applying the Reynolds decomposition strictly to all independent variables is to consider a mass-weighted decomposition known as Favre averaging (Libby and Williams [12], p. 15; Kuo [13], p. 419). This simplifies all of the transport equations and eases modeling. A Favre-averaged variable $\tilde{\phi}$ is defined in terms of Reynolds averages as

$$\tilde{\phi} \equiv \frac{\overline{\rho\phi}}{\bar{\rho}}. \tag{2.66}$$

A variable can then be decomposed into its Favre-mean and fluctuating component as

$$\phi = \tilde{\phi} + \phi'', \tag{2.67}$$

where $\overline{\rho\phi''} = 0$. Note that $\overline{\phi''} \neq 0$. The relation between time averaged and Favre-averaged quantities is

$$\tilde{\phi} = \bar{\phi}\left(1 + \frac{\overline{\rho'\phi'}}{\bar{\rho}\bar{\phi}}\right). \tag{2.68}$$

Favre averaging is used for all turbulent transport equations solved in SIERRA/Fuego.

For the RANS formulation used here, the laminar conservation equations of Section 2.2 are first temporally filtered, revealing additional terms that can be simplified by substituting the Favre decomposition, resulting in the Favre-filtered equations that will be presented in Section 2.5. This procedure results in new terms in the equations that consist of time averages of products of fluctuating quantities, called Reynolds stresses. These moments must be modeled to close the system of equations.

The length of the time filter is typically much larger than the time scales of a turbulent flow, meaning that all time scales from the largest turbulence scale down to the minimum Kolmogorov scale are represented by these Reynolds stresses. In a strict sense, there can be no time dependence of a mean (time-averaged) quantity. However, if there are variations in mean quantities that occur on time intervals long compared to the averaging interval, then the transient terms for the mean quantities may be justified and required. For this reason, unsteady RANS simulations are possible with the present formulation. The available RANS turbulence closure models are discussed in Section 2.6.

### 2.4.2. LES Spatial Filtering

Unlike the RANS approach which models most or all of the turbulent fluctuations, LES directly solves for all resolved turbulent length scales and only models the smallest scales below the grid size. In this way, a majority of the problem-dependent, energy-containing turbulent structure is directly solved in a model-free fashion. The subgrid scales are closer to being isotropic than the resolved scales, and they

generally act to dissipate turbulent kinetic energy cascaded down from the larger scales in momentum-driven turbulent flows. Modeling of these small scales is generally more straightforward than RANS approaches, and overall solutions are usually more tolerant to LES modeling errors because the subgrid scales comprise such a small portion of the overall turbulent structure. While LES is generally accepted to be much more accurate than RANS approaches for complex turbulent flows, it is also significantly more expensive than equivalent RANS simulations due to the finer grid resolution required. Additionally, since LES results in a full unsteady solution, the simulation must be run for a long time to gather any desired time-averaged statistics. The trade-off between accuracy and cost must be weighed before choosing one method over the other.

The separation of turbulent length scales required for LES is obtained by using a spatial filter rather than the RANS temporal filter. This filter has the mathematical form

$$\overline{\phi(\boldsymbol{x},t)} \equiv \int_{-\infty}^{+\infty} \phi(\boldsymbol{x}',t) G(\boldsymbol{x}'-\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}', \tag{2.69}$$

which is a convolution integral over physical space $\boldsymbol{x}$ with the spatially-varying filter function $G$. The filter function has the normalization property $\int_{-\infty}^{+\infty} G(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = 1$, and it has a characteristic length scale $\Delta$ so that it filters out turbulent length scales smaller than this size. In the present formulation, a simple "box filter" is used for the filter function,

$$G(\boldsymbol{x}'-\boldsymbol{x}) = \begin{cases} 1/V & : \quad (\boldsymbol{x}'-\boldsymbol{x}) \in \mathcal{V} \\ 0 & : \quad \text{otherwise} \end{cases}, \tag{2.70}$$

where $V$ is the volume of control volume $\mathcal{V}$ whose central node is located at $\boldsymbol{x}$. This is essentially an unweighted average over the control volume. The length scale of this filter is approximated by $\Delta = V^{\frac{1}{3}}$. This is typically called the grid filter, as it filters out scales smaller than the computational grid size.

Similar to the RANS temporal filter, a variable can be represented in terms of its filtered and subgrid fluctuating components as

$$\phi = \bar{\phi} + \phi'. \tag{2.71}$$

For most forms of the filter function $G(\boldsymbol{x})$, repeated applications of the grid filter to a variable do not yield the same result. In other words, $\bar{\bar{\phi}} \neq \bar{\phi}$ and therefore $\overline{\phi'} \neq 0$, unlike with the RANS temporal averages.

As with the RANS formulation, modeling is much simplified in the presence of large density variations if a Favre-filtered approach is used. A Favre-filtered variable $\tilde{\phi}$ is defined as

$$\tilde{\phi} \equiv \frac{\overline{\rho\phi}}{\bar{\rho}} \tag{2.72}$$

and a variable can be decomposed in terms of its Favre-filtered and subgrid fluctuating component as

$$\phi = \tilde{\phi} + \phi''. \tag{2.73}$$

Again, note that the useful identities for the Favre-filtered RANS variables do not apply, so that $\tilde{\bar{\phi}} \neq \tilde{\phi}$ and $\overline{\phi''} \neq 0$. The Favre-filtered approach is used for all LES models in SIERRA/Fuego.

## 2.5. TURBULENT FLOW EQUATIONS, FAVRE-AVERAGED

The Favre-averaged turbulent transport equations are derived from the laminar equations of Section 2.2 by passing the equations through either the RANS temporal filter of Equation 2.65 or the LES spatial filter of Equation 2.69. The mathematical form of the equations are essentially identical between the two filtering methods, so only a single set of equations will be presented. Care should be taken to interpret the filters as either temporal or spatial, depending on the closure models selected. While it is the Favre-averaged form of the equations that are solved, a comparison of the simple Reynolds-averaged and the Favre-averaged form is given in Appendix B for reference.

The approach most commonly used in turbulence modeling is called the Boussinesq eddy viscosity approximation, which relates the turbulent stress tensor to the filtered strain rate tensor through a modeled turbulent eddy viscosity. This general modeling approach has shown remarkable success for a broad range of problems (Wilcox [14]), and is the approach used in SIERRA/Fuego. A similar approach is used for scalar transport, where the scalar flux vector is related to scalar gradients through a modeled diffusion coefficient.

The following subsections describe the turbulent transport equations expressed in terms of a turbulent eddy viscosity or turbulent diffusion coefficient through the Boussinesq approximation. The treatment of these coefficients is dependent upon which of the many closure models are selected, and will be described in Section 2.6.

### 2.5.1. Conservation of Mass

The integral form of the Favre-filtered continuity equation used for turbulent transport is

$$\int \frac{\partial \bar{\rho}}{\partial t} \mathrm{d}V + \int \bar{\rho} \tilde{u}_j n_j \mathrm{d}S = 0. \tag{2.74}$$

This equation is in closed form, and no additional modeling is required.

### 2.5.2. Conservation of Momentum

The integral form of the Favre-filtered momentum equations used for turbulent transport are

$$\int \frac{\partial \bar{\rho} \tilde{u}_i}{\partial t} \mathrm{d}V + \int \bar{\rho} \tilde{u}_i \tilde{u}_j n_j \mathrm{d}S + \int \bar{p} n_i \mathrm{d}S = \int \bar{\tau}_{ij} n_j \mathrm{d}S + \int \tau_{u_i u_j} n_j \mathrm{d}S + \int \left(\bar{\rho} - \rho_\circ\right) g_i \mathrm{d}V, \tag{2.75}$$

where the turbulent stress $\tau_{u_i u_j}$ is defined as

$$\tau_{u_i u_j} \equiv -\bar{\rho}(\widetilde{u_i u_j} - \tilde{u}_i \tilde{u}_j). \tag{2.76}$$

#### 2.5.2.1. RANS Modeling

For RANS simulations, $\tau_{u_i u_j}$ represents the Reynolds stress tensor and can be reduced to the form $\tau_{u_i u_j} = -\overline{\rho u_i'' u_j''}$ by substitution of the Favre decomposition $u_i \equiv \tilde{u}_i + u_i''$ of each variable and simplifying. The deviatoric (trace-free) part of the stress tensor is defined as

$$\begin{aligned} \tau^D_{u_i u_j} &\equiv \tau_{u_i u_j} - \frac{1}{3}\tau_{u_k u_k}\delta_{ij} \\ &= \tau_{u_i u_j} + \frac{2}{3}\bar{\rho}\tilde{k}\delta_{ij} \end{aligned} \tag{2.77}$$

where the turbulent kinetic energy is defined as $\tilde{k} \equiv \frac{1}{2}\widetilde{u_k'' u_k''}$. The deviatoric part of the Reynolds stress tensor is modeled by the Boussinesq approximation which relates the Reynolds stresses to the filtered strain rate tensor through a modeled turbulent viscosity $\mu_t$, resulting in

$$\begin{aligned} \tau^D_{u_i u_j} &= \mu_t \left( \frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} \right) - \frac{2}{3}\mu_t \frac{\partial \tilde{u}_k}{\partial x_k}\delta_{ij} \\ &= 2\mu_t \left( \tilde{S}_{ij} - \frac{1}{3}\tilde{S}_{kk}\delta_{ij} \right), \end{aligned} \tag{2.78}$$

where the filtered strain rate tensor is defined by

$$\tilde{S}_{ij} \equiv \frac{1}{2}\left( \frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} \right). \tag{2.79}$$

Substituting this into Equation 2.77 yields the modeled form of the full Reynolds stress tensor (Kuo [13], p. 445)

$$\tau_{u_i u_j} = 2\mu_t \left( \tilde{S}_{ij} - \frac{1}{3}\tilde{S}_{kk}\delta_{ij} \right) - \frac{2}{3}\bar{\rho}\tilde{k}\delta_{ij}. \tag{2.80}$$

The Favre-filtered momentum equations then become

$$\begin{aligned} &\int \frac{\partial \bar{\rho}\tilde{u}_i}{\partial t}\mathrm{d}V + \int \bar{\rho}\tilde{u}_i\tilde{u}_j n_j \mathrm{d}S + \int \left( \bar{p} + \frac{2}{3}\bar{\rho}\tilde{k} \right) n_i \mathrm{d}S = \\ &\quad \int 2(\mu + \mu_t)\left( \tilde{S}_{ij} - \frac{1}{3}\tilde{S}_{kk}\delta_{ij} \right) n_j \mathrm{d}S + \int (\bar{\rho} - \rho_\circ)\, g_i \mathrm{d}V, \end{aligned} \tag{2.81}$$

where RANS closure models for the turbulent viscosity $\mu_t$ are presented in Section 2.6.

#### 2.5.2.2. LES Modeling

For LES, $\tau_{u_i u_j}$ in Equation 2.75 represents the subgrid stress tensor. The deviatoric part of the subgrid stress tensor is defined as

$$\begin{aligned} \tau^D_{u_i u_j} &\equiv \tau_{u_i u_j} - \frac{1}{3}\tau_{u_k u_k}\delta_{ij} \\ &= \tau_{u_i u_j} + \frac{2}{3}\bar{\rho}q^2\delta_{ij}, \end{aligned} \tag{2.82}$$

where the subgrid turbulent kinetic energy is defined as $q^2 \equiv \frac{1}{2}(\widetilde{u_k u_k} - \tilde{u_k}\tilde{u_k})$. The deviatoric part of the subgrid stress tensor is then modeled similar to RANS closures as (Moin, et al. [15])

$$\tau^D_{u_i u_j} = 2\mu_t \left( \tilde{S}_{ij} - \frac{1}{3}\tilde{S}_{kk}\delta_{ij} \right). \tag{2.83}$$

Substituting this into Equation 2.82 yields the modeled form of the full subgrid stress tensor

$$\tau_{u_i u_j} = 2\mu_t \left( \tilde{S}_{ij} - \frac{1}{3}\tilde{S}_{kk}\delta_{ij} \right) - \frac{2}{3}\bar{\rho} q^2 \delta_{ij}. \tag{2.84}$$

For low Mach-number flows, a vast majority of the turbulent kinetic energy is contained at resolved scales (Erlebacher, et al. [16]). For this reason, the subgrid turbulent kinetic energy $q^2$ will not be directly treated and will instead be included in the pressure as an additional normal stress. The Favre-filtered momentum equations then become

$$\begin{aligned} \int \frac{\partial \bar{\rho}\tilde{u}_i}{\partial t} \mathrm{d}V + \int \bar{\rho}\tilde{u}_i\tilde{u}_j n_j \mathrm{d}S + \int \left( \bar{p} + \frac{2}{3}\bar{\rho}q^2 \right) n_i \mathrm{d}S = \\ \int 2(\mu + \mu_t) \left( \tilde{S}_{ij} - \frac{1}{3}\tilde{S}_{kk}\delta_{ij} \right) n_j \mathrm{d}S + \int (\bar{\rho} - \rho_\circ) g_i \mathrm{d}V, \end{aligned} \tag{2.85}$$

where LES closure models for the subgrid turbulent eddy viscosity $\mu_t$ are presented in Section 2.6.

### 2.5.3. Conservation of Energy

The integral form of the Favre-filtered energy equation used for turbulent transport is

$$\begin{aligned} \int \frac{\partial \bar{\rho}\tilde{h}}{\partial t} \mathrm{d}V + \int \bar{\rho}\tilde{h}\tilde{u}_j n_j \mathrm{d}S &= -\int \bar{q}_j n_j \mathrm{d}S - \int \tau_{hu_j} n_j \mathrm{d}S - \int \frac{\partial \bar{q}^r_i}{\partial x_i} \mathrm{d}V \\ &\quad + \int \left( \frac{\partial P}{\partial t} + \tilde{u}_j \frac{\partial P}{\partial x_j} \right) \mathrm{d}V + \int \overline{\tau_{ij} \frac{\partial u_i}{\partial x_j}} \mathrm{d}V. \end{aligned} \tag{2.86}$$

The simple Fickian diffusion velocity approximation, Equation 2.46, is assumed, so that the mean diffusive heat flux vector $\bar{q}_j$ is

$$\bar{q}_j = -\overline{\left[ \frac{\mu}{\mathrm{Pr}} \frac{\partial h}{\partial x_j} - \frac{\mu}{\mathrm{Pr}} \sum_{k=1}^{K} h_k \frac{\partial Y_k}{\partial x_j} \right]} - \overline{\frac{\mu}{\mathrm{Sc}} \sum_{k=1}^{K} h_k \frac{\partial Y_k}{\partial x_j}}. \tag{2.87}$$

If Sc = Pr, i.e., unity Lewis number (Le = 1), then the diffusive heat flux vector simplifies to $\bar{q}_j = -\frac{\mu}{\mathrm{Pr}} \frac{\partial \tilde{h}}{\partial x_j}$. The viscous dissipation term is closed by

$$\begin{aligned} \overline{\tau_{ij} \frac{\partial u_i}{\partial x_j}} &= \left( (\mu + \mu_t) \left( \frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} \right) - \frac{2}{3} \left( \bar{\rho}\tilde{k} + \mu_t \frac{\partial \tilde{u}_k}{\partial x_k} \right) \delta_{ij} \right) \frac{\partial \tilde{u}_i}{\partial x_j} \\ &= \left[ 2\mu \tilde{S}_{ij} + 2\mu_t \left( \tilde{S}_{ij} - \frac{1}{3}\tilde{S}_{kk}\delta_{ij} \right) - \frac{2}{3}\bar{\rho}\tilde{k}\delta_{ij} \right] \frac{\partial \tilde{u}_i}{\partial x_j}. \end{aligned} \tag{2.88}$$

The turbulent diffusive flux vector $\tau_{hu_j}$ in Equation 2.86 is defined as

$$\tau_{hu_j} \equiv \bar{\rho}\left(\widetilde{hu_j} - \tilde{h}\tilde{u}_j\right). \tag{2.89}$$

For RANS simulations, $\tau_{hu_j}$ represents the turbulent energy diffusive flux vector and is simplified to the form $\tau_{hu_j} = \overline{\rho h'' u_j''}$ by substitution of the Favre decomposition of each variable. It is then modeled by

$$\tau_{hu_j} = \overline{\rho h'' u_j''} = -\frac{\mu_t}{\mathrm{Pr}_t}\frac{\partial \tilde{h}}{\partial x_j}, \tag{2.90}$$

where $\mathrm{Pr}_t$ is the turbulent Prandtl number and $\mu_t$ is the modeled turbulent eddy viscosity from momentum closure. For LES, $\tau_{hu_j}$ represents the subgrid turbulent energy diffusive flux vector, and is modeled in the same way as

$$\tau_{hu_j} = -\frac{\mu_t}{\mathrm{Pr}_t}\frac{\partial \tilde{h}}{\partial x_j}, \tag{2.91}$$

where $\mathrm{Pr}_t$ is the subgrid turbulent Prandtl number and $\mu_t$ is the modeled subgrid turbulent eddy viscosity from momentum closure.

The resulting filtered and modeled turbulent energy equation for both RANS and LES is given in Libby and Williams [12], p. 25, as

$$\begin{aligned}\int \frac{\partial \bar{\rho}\tilde{h}}{\partial t}\mathrm{d}V + \int \bar{\rho}\tilde{h}\tilde{u}_j n_j \mathrm{d}S &= \int \left(\frac{\mu}{\mathrm{Pr}} + \frac{\mu_t}{\mathrm{Pr}_t}\right)\frac{\partial \tilde{h}}{\partial x_j} n_j \mathrm{d}S - \int \frac{\partial \bar{q}_i^r}{\partial x_i}\mathrm{d}V \\ &+ \int \left(\frac{\partial P}{\partial t} + \tilde{u}_j\frac{\partial P}{\partial x_j}\right)\mathrm{d}V + \int \overline{\tau_{ij}\frac{\partial u_j}{\partial x_j}}\mathrm{d}V.\end{aligned} \tag{2.92}$$

This equation is also given in Gran et al. [17] (without the transient and radiation source terms and the additional term for laminar transport). The turbulent Prandtl number must have the same value as the turbulent Schmidt number for species transport to maintain unity Lewis number.

### 2.5.4. Conservation of Species

The integral form of the Favre-filtered species equation used for turbulent transport is

$$\int \frac{\partial \bar{\rho}\tilde{Y}_k}{\partial t}\mathrm{d}V + \int \bar{\rho}\tilde{Y}_k\tilde{u}_j n_j \mathrm{d}S = -\int \tau_{Y_k u_j} n_j \mathrm{d}S - \int \overline{\rho Y_k \hat{u}_{j,k}} n_j \mathrm{d}S + \int \overline{\dot{\omega}_k}\mathrm{d}V, \tag{2.93}$$

where the form of diffusion velocities (see Equation 2.46) assumes the Fickian approximation with a constant value of diffusion velocity for consistency with the turbulent form of the energy equation, Equation 2.86.

The turbulent diffusive flux vector $\tau_{Y_k u_j}$ is defined as

$$\tau_{Y_k u_j} \equiv \bar{\rho}\left(\widetilde{Y_k u_j} - \tilde{Y}_k\tilde{u}_j\right). \tag{2.94}$$

For RANS simulations, $\tau_{Y_k u_j}$ represents the turbulent species diffusive flux vector and is simplified to the form $\tau_{Y_k u_j} = \overline{\rho Y_k'' u_j''}$ by substitution of the Favre decomposition of each variable. It is then modeled as

$$\tau_{Y_k u_j} = \overline{\rho Y_k'' u_i''} = -\frac{\mu_t}{\mathrm{Sc}_t}\frac{\partial \tilde{Y}_k}{\partial x_i}, \tag{2.95}$$

where $\mathrm{Sc}_t$ is the turbulent Schmidt number for all species and $\mu_t$ is the modeled turbulent eddy viscosity from momentum closure. For LES, $\tau_{Y_k u_j}$ represents the subgrid turbulent species diffusive flux vector, and is modeled identically as

$$\tau_{Y_k u_j} = -\frac{\mu_t}{\mathrm{Sc}_t}\frac{\partial \tilde{Y}_k}{\partial x_i}, \tag{2.96}$$

where $\mathrm{Sc}_t$ is the subgrid turbulent Schmidt number for all species and $\mu_t$ is the subgrid modeled turbulent eddy viscosity from momentum closure.

The Favre-filtered and modeled turbulent species transport equation for both RANS and LES then becomes (Gran et al. [17])

$$\int \frac{\partial \bar{\rho}\tilde{Y}_k}{\partial t}\mathrm{d}V + \int \bar{\rho}\tilde{Y}_k\tilde{u}_j n_j \mathrm{d}S = \int \left(\frac{\mu}{\mathrm{Sc}} + \frac{\mu_t}{\mathrm{Sc}_t}\right)\frac{\partial \tilde{Y}_k}{\partial x_j} n_j \mathrm{d}S + \int \overline{\dot{\omega}}_k \mathrm{d}V. \tag{2.97}$$

If transporting both energy and species equations, the laminar Prandtl number must be equal to the laminar Schmidt number and the turbulent Prandtl number must be equal to the turbulent Schmidt number to maintain unity Lewis number. Although there is a species conservation equation for each species in a mixture of $K$ species, only $K-1$ species equations need to be solved since the mass fractions sum to unity and

$$\tilde{Y}_k = 1 - \sum_{j \neq k}^{K} \tilde{Y}_j. \tag{2.98}$$

### 2.5.5. Radiation Transport

The Favre-averaged energy equation, Equation 2.93, requires the time-averaged radiative flux divergence. From Equation 2.60, the time-averaged radiative flux divergence is given by

$$\overline{\frac{\partial q_i^r}{\partial x_i}} = 4\sigma\overline{\mu_a T^4} - \overline{\mu_a G}. \tag{2.99}$$

For optically thin turbulent eddies, which is the case for many combustion applications, fluctuations in the absorption coefficient and the scalar flux are weakly correlated [8] so Equation 2.99 may be simplified to

$$\overline{\frac{\partial q_i^r}{\partial x_i}} = 4\sigma\overline{\mu_a T^4} - \bar{\mu}_a \bar{G}. \tag{2.100}$$

The time averaged scalar flux is obtained from the time averaged Boltzmann radiative transport equation

$$s_i \frac{\partial}{\partial x_i} \overline{I\left(s\right)} + \bar{\mu}_a \overline{I\left(s\right)} = \frac{\overline{\mu_a \sigma T^4}}{\pi}, \tag{2.101}$$

where the correlation between the turbulent fluctuations in the absorption coefficient and the intensity is assumed small to simplify the absorption term.

Both Equation 2.100 and Equation 2.101 include the time averaged emission term, $\overline{\alpha T^4}$, which may significantly increase the radiative emission from a turbulent flame above what would be estimated from the mean temperature and absorption coefficient values. The details of the closure used for this term are discussed in the turbulent combustion model section.

# 2.6. TURBULENCE CLOSURE MODELS

The Favre-filtered turbulent flow equations of the previous section have been modeled in terms of $\mu_t$, the turbulent eddy viscosity for RANS simulations and the subgrid turbulent eddy viscosity for LES. Evaluation of this eddy viscosity is dependent upon the closure model selected. All models supported by SIERRA/Fuego are described below.

## 2.6.1. Standard $k$-$\epsilon$ RANS Model

The standard $k$-$\epsilon$ closure model is a two-equation type of model, where transport equations for the turbulent kinetic energy and the turbulent dissipation rate are solved to obtain length-scale and time-scale estimates for the local turbulence field, to be used for modeling the turbulent eddy viscosity $\mu_t$. The turbulent kinetic energy, $k$, and the dissipation rate of turbulent kinetic energy, $\epsilon$, are given by (Gran et al. [17])

$$\int \frac{\partial \bar{\rho} k}{\partial t} \mathrm{d}V + \int \bar{\rho} k \tilde{u}_j n_j \mathrm{d}S = \int \frac{\mu_t}{\sigma_k} \frac{\partial k}{\partial x_j} n_j \mathrm{d}S + \int \left(P_k - \bar{\rho}\epsilon\right) \mathrm{d}V \tag{2.102}$$

$$\int \frac{\partial \bar{\rho} \epsilon}{\partial t} \mathrm{d}V + \int \bar{\rho} \epsilon \tilde{u}_j n_j \mathrm{d}S = \int \frac{\mu_t}{\sigma_\epsilon} \frac{\partial \epsilon}{\partial x_j} n_j \mathrm{d}S + \int \frac{\epsilon}{k} \left(C_{\epsilon 1} P_k - C_{\epsilon 2} \bar{\rho}\epsilon\right) \mathrm{d}V, \tag{2.103}$$

respectively, where the turbulence production rate, $P_k$, is defined as

$$P_k \equiv -\overline{\rho u_i'' u_j''} \frac{\partial \tilde{u}_i}{\partial x_j}, \tag{2.104}$$

and is modeled using the same Boussinesq approximation as in Equation 2.80,

$$\begin{aligned} P_k &= \mu_t \left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i}\right) \frac{\partial \tilde{u}_i}{\partial x_j} - \frac{2}{3} \left(\bar{\rho} k + \mu_t \frac{\partial \tilde{u}_k}{\partial x_k}\right) \frac{\partial \tilde{u}_m}{\partial x_m} \\ &= \left[2\mu_t \left(\tilde{S}_{ij} - \frac{1}{3} \tilde{S}_{kk} \delta_{ij}\right) - \frac{2}{3} \bar{\rho} \tilde{k} \delta_{ij}\right] \frac{\partial \tilde{u}_i}{\partial \tilde{x}_j}. \end{aligned} \tag{2.105}$$

The turbulent eddy viscosity is then given by the Prandtl-Kolmogorov relationship,

$$\mu_t = \mathrm{C}_\mu \bar{\rho} k \tau. \tag{2.106}$$

where $\tau = min(\frac{k}{\epsilon}, dt_f)$. The filter time, $dt_f$ is provided by the temporally filtered Navier Stokes model (Tieszen et al. [18]). The parameters $C_{\epsilon 1}$, $C_{\epsilon 2}$, $\sigma_k$, and $\sigma_\epsilon$ are adjustable constants.

Frequently, although not formally justified in high Reynolds flows, the diffusion coefficient for the turbulent kinetic energy and turbulence dissipation, Equations 2.102 and 2.103, may include the molecular viscosity. This option is supported within Fuego by entering the following command line in the Fuego region block, `include molecular viscosity in k-e diffusion term`.

### 2.6.2. Low Reynolds Number $k$-$\epsilon$ RANS Model

In the case of the low Reynolds number turbulent flows, the standard $k$-$\epsilon$ transport equations can be modified to contain additional damping functions to improve their accuracy. The low Reynolds number model of Launder and Sharma [19] are used here, which modify the turbulent kinetic energy equation, Equation 2.102, to includes an additional right-hand-side source term

$$S_k^{lr} = -2\mu \left( \frac{\partial k}{\partial x_j} \right)^2 \tag{2.107}$$

and the dissipation rate equation to include the non-isotropic dissipation source term

$$S_\epsilon^{lr} = -2\nu\nu^T \left( \frac{\partial^2 \tilde{u}_i}{\partial x_k \partial x_j} \right)^2. \tag{2.108}$$

The constants in the dissipation rate equation are modified by damping coefficients, $C_{\epsilon_1} = f_1 C_{\epsilon_1}$ and $C_{\epsilon_2} = C_{\epsilon_2} f_2$, where $f_1$ is unity and $f_2 = 1 - 0.3e^{-R_t^2}$.

The eddy viscosity is then given by

$$\mu_t = \mathrm{C}_\mu \bar{\rho} f_\mu k \tau. \tag{2.109}$$

Wall functions for momentum and turbulence quantities are not used with this model.

### 2.6.3. RNG $k$-$\epsilon$ RANS Model

The RNG $k$-$\epsilon$ model was derived using a rigorous statistical decomposition of the velocity field called renormalization group (RNG) theory. This model has several significant benefits over the standard $k$-$\epsilon$ model, including improved accuracy for rapidly strained flows, swirling flows, and low Reynolds number flows, without additional modifications. Additionally, values for the model constants are derived analytically rather than being evaluated empirically. Papageorgakis and Assanis [20] describe the version of the RNG $k$-$\epsilon$ model as implemented here.

The same turbulent kinetic energy equation as in the standard $k$-$\epsilon$ model, Equation 2.102, is used for the RNG $k$-$\epsilon$ equation. The turbulent kinetic energy dissipation rate equation is the same as Equation 2.103, with the addition of a single source term on the right-hand-side of the equation,

$$S_\epsilon^{\mathrm{RNG}} = -\frac{C_\mu \eta^3 (1 - \eta/\eta_o)}{1 + \beta\eta^3} \frac{\epsilon^2}{k}, \tag{2.110}$$

where $C_\mu$, $\beta$, and $\eta_o$ are model constants, and

$$\eta = (2\tilde{S}_{ij}\tilde{S}_{ij})^{\frac{1}{2}} \frac{k}{\epsilon}. \tag{2.111}$$

As with the standard $k$-$\epsilon$ model, the turbulent eddy viscosity is then given by the Prandtl-Kolmogorov relationship,

$$\mu_t = C_\mu \bar{\rho} k \tau. \tag{2.112}$$

### 2.6.4. $v^2$-$f$ RANS Model

Durbin [21] introduced a method for handling the wall region without using either wall functions or damping functions. In his method a fine grid is required near the wall (e.g., the first grid point is typically within one dimensionless unit of distance from the wall where the coordinate normal to the wall is nondimensionalized with the inner scale for a turbulent boundary layer, $y^+ = y u_\tau / \nu < 1$ at the first grid point, where $u_\tau$ is the friction velocity, $\sqrt{\tau_w/\bar{\rho}}$). The model employs two transport equations in addition to slightly modified $k$ and $\epsilon$ equations to account for the nonhomogeneous region near the wall. The eddy viscosity is formulated using the component of turbulent kinetic energy normal to the wall for velocity scaling (instead of using $\sqrt{k}$ as in the standard $k$-$\epsilon$ model).

The turbulent kinetic energy, $k$, is given by Equation 2.102 while the dissipation rate of turbulent kinetic energy, $\epsilon$, is given by

$$\int \frac{\partial \bar{\rho}\epsilon}{\partial t} \mathrm{d}V + \int \bar{\rho}\epsilon \tilde{u}_j n_j \mathrm{d}S = \int \frac{\mu_t}{\sigma_\epsilon} \frac{\partial \epsilon}{\partial x_j} n_j \mathrm{d}S + \int \frac{1}{T} \left( C'_{\epsilon 1} P_k - C_{\epsilon 2} \bar{\rho}\epsilon \right) \mathrm{d}V. \tag{2.113}$$

The time scale, $T$, is the usual time scale $k/\epsilon$, away from the wall region; however, near the wall, if $k/\epsilon$ becomes smaller than the Kolmogorov time scale $\sqrt{\nu/\epsilon}$, then the latter is used for $T$. This is formally stated by

$$T = \min \left[ T_1, \frac{\alpha}{2\sqrt{3}} \frac{k}{\overline{v^2} C_\mu \sqrt{\tilde{S}^2}} \right] \tag{2.114}$$

$$T_1 = \max \left[ \frac{k}{\epsilon}, 6\sqrt{\frac{\nu}{\epsilon}} \right], \tag{2.115}$$

where

$$\tilde{S}^2 = \tilde{S}_{ij}\tilde{S}_{ij} = \frac{1}{4} \left( \frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} \right) \left( \frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} \right) \tag{2.116}$$

and the modified constant, $C'_{\epsilon_1}$, is given by

$$C'_{\epsilon_1} = C_{\epsilon_1}\left(1 + 0.045\sqrt{k/\overline{v^2}}\right). \tag{2.117}$$

The model includes a transport equation for $\overline{v^2}$,

$$\frac{\partial \bar{\rho}\overline{v^2}}{\partial t} + \frac{\partial \bar{\rho}\tilde{u}_j\overline{v^2}}{\partial x_j} = \frac{\partial}{\partial x_j}\left[(\mu + \mu_t)\frac{\partial \overline{v^2}}{\partial x_j}\right] + \bar{\rho}kf - \frac{\bar{\rho}N\overline{v^2}}{T_1}. \tag{2.118}$$

An elliptic relaxation model equation is formulated to solve for the variable $f$ in the above equation. The purpose of the elliptic relaxation model is to account for nonlocal effects such as wall blocking; the equation is given by

$$f - L^2\frac{\partial}{\partial x_j}\left(\frac{\partial f}{\partial x_j}\right) = C_1\frac{\left(2/3 - \overline{v^2}/k\right)}{T_1} + C_2 2\nu_t\frac{\tilde{S}^2}{k} + (N-1)\frac{\overline{v^2}/k}{T_1}. \tag{2.119}$$

Finally, the turbulent eddy viscosity is given by

$$\mu_t = C_\mu\bar{\rho}\overline{v^2}\tau. \tag{2.120}$$

### 2.6.5. $k-\omega$ RANS Model

The $k-\omega$ turbulence model and its variants are similar in structure to the $k-\epsilon$ models. However, instead of computing the turbulent dissipation rate directly, the $k-\omega$ model models the transport the reciprocal of a turbulent timescale referred to as the turbulent frequency. This quantity, $\omega$, can be related to the turbulent dissipation by

$$\epsilon = \beta^* k\omega. \tag{2.121}$$

The the transport equations are given by the 2006 model, (Wilcox [22]),

$$\int \frac{\partial \bar{\rho}k}{\partial t}\mathrm{d}V + \int \bar{\rho}k\tilde{u}_j n_j \mathrm{d}S = \int (\mu + \sigma_k\frac{\bar{\rho}k}{\omega})\frac{\partial k}{\partial x_j}n_j \mathrm{d}V + \int \left(P_k^\omega - \beta^*\bar{\rho}k\omega\right)\mathrm{d}V, \tag{2.122}$$

$$\int \frac{\partial \bar{\rho}\omega}{\partial t}\mathrm{d}V + \int \bar{\rho}\omega\tilde{u}_j n_j \mathrm{d}S = \int (\mu + \sigma_\omega\frac{\bar{\rho}k}{\omega})\frac{\partial \omega}{\partial x_j}n_j \mathrm{d}V + \int \left(\gamma\frac{\omega}{k}P_k^\omega - \beta\bar{\rho}\omega^2 + \frac{\bar{\rho}\sigma_d}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial \omega}{\partial x_j}\right)\mathrm{d}V. \tag{2.123}$$

The user is to note the above standard for writing the effective diffusive flux coefficient. The model also has a number of adjustable parameters: $\beta_0 = 0.0708$, $\beta^* = 0.09$, $\gamma = \frac{13}{25}$, $C_{lim} = \frac{7}{8}$, $\sigma_k = 0.6$, and $\sigma_\omega = 0.5$. The constant $\beta$ is given by,

$$\beta = \beta_0 f_\beta \tag{2.124}$$

where

$$f_\beta = \frac{1 + 85\chi_\omega}{1 + 100\chi_\omega} \tag{2.125}$$

The value of $\chi_\omega$ is as follows:

$$\chi_\omega = |\frac{\Omega_{ij}\Omega_{jk}S_{ki}}{(\beta^*\omega)^3}| \tag{2.126}$$

The production term is the same as in $k - \epsilon$. Typically limiters are used to prevent it from exceeding the dissipation rate by too large an amount. Although the 2006 description does not speak of production limiters, other sources that use the 2006 model do, i.e.

$$P_k^\omega = \max\left(P_k, 10\bar{\rho}k\omega\right). \tag{2.127}$$

The value of 10 is expected to be a user specified quantity (see input file manual for more details). In general, this term is defaulted to a very high number.

The eddy viscosity is

$$\mu_T = \bar{\rho}\frac{k}{\hat{\omega}}. \tag{2.128}$$

where $\hat{\omega}$ is,

$$\hat{\omega} = \max(\omega, C_{lim}\sqrt{\frac{2S_{ij}S_{ij}}{\beta^*}}). \tag{2.129}$$

### 2.6.6. Shear Stress Transport (SST)

It has been observed that standard 1998 $k - \omega$ models display a strong sensitivity to the free stream value of $\omega$. To remedy, this, an alternative set of transport equations have been used that are based on smoothly blending the $k - \omega$ model near a wall with $k - \epsilon$ away from the wall (see Mentor [23]). Because of the relationship between $\omega$ and $\epsilon$, the transport equations for turbulent kinetic energy and dissipation can be transformed into equations involving $k$ and $\omega$. Aside from constants, the transport equation for $k$ is unchanged. However, an additional cross-diffusion term is present in the $\omega$ equation. Blending is introduced by using smoothing which is a function of the distance from the wall, $F(y)$. The transport equations for the Mentor 2003 model ( [23]) are provided by the following:

$$\int \frac{\partial \bar{\rho}k}{\partial t}\mathrm{d}V + \int \bar{\rho}k\tilde{u}_j n_j \mathrm{d}S = \int (\mu + \hat{\sigma}_k\mu_t)\frac{\partial k}{\partial x_j}n_j + \int \left(P_k^\omega - \beta^*\bar{\rho}k\omega\right)\mathrm{d}V, \tag{2.130}$$

$$\int \frac{\partial \bar{\rho}\omega}{\partial t}\mathrm{d}V + \int \bar{\rho}\omega\tilde{u}_j n_j \mathrm{d}S = \int (\mu + \hat{\sigma}_\omega\mu_t)\frac{\partial \omega}{\partial x_j}n_j + \int 2(1-F)\frac{\bar{\rho}\sigma_{\omega 2}}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial \omega}{\partial x_j}\mathrm{d}V + \int \left(\frac{\hat{\gamma}}{\nu_t}P_k^\omega - \hat{\beta}\bar{\rho}\omega^2\right)\mathrm{d}V. \tag{2.131}$$

The model coefficients, $\hat{\sigma}_k$, $\hat{\sigma}_\omega$, $\hat{\gamma}$ and $\hat{\beta}$ must also be blended, which is represented by

$$\hat{\phi} = F\phi_1 + (1-F)\phi_2. \tag{2.132}$$

where $\sigma_{k1} = 0.85$, $\sigma_{k2} = 1.0$, $\sigma_{\omega 1} = 0.5$, $\sigma_{\omega 2} = 0.856$, $\gamma_1 = \frac{5}{9}$, $\gamma_2 = 0.44$, $\beta_1 = 0.075$ and $\beta_2 = 0.0828$.

The blending function is given by

$$F = \tanh(arg_1^4), \tag{2.133}$$

where

$$arg_1 = \min\left(\max\left(\frac{\sqrt{k}}{\beta^*\omega y}, \frac{500\mu}{\bar{\rho}y^2\omega}\right), \frac{4\bar{\rho}\sigma_{\omega 2}k}{CD_{k\omega}y^2}\right). \tag{2.134}$$

The final parameter is

$$CD_{k\omega} = \max\left(2\bar{\rho}\sigma_{\omega 2}\frac{1}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial \omega}{\partial x_j}, 10^{-10}\right). \tag{2.135}$$

In the 2003 SST model description, the production term is expected to be limited:

$$P_k^\omega = \max\left(P_k, 10\bar{\rho}k\omega\right). \tag{2.136}$$

The value of 10 is expected to be a user specified quantity (see input file manual for more details). In general, this term is defaulted to a very high number.

An important component of the SST model is the different expression used for the eddy viscosity,

$$\mu_t = \frac{a_1\bar{\rho}k}{\max\left(a_1\omega, SF_2\right)}, \tag{2.137}$$

where $F_2$ is another blending function given by

$$F_2 = \tanh(arg_2^2). \tag{2.138}$$

The final parameter is

$$arg_2 = \max\left(\frac{2\sqrt{k}}{\beta^*\omega y}, \frac{500\mu}{\bar{\rho}\omega y^2}\right). \tag{2.139}$$

### 2.6.7. Standard Smagorinsky LES Model

The standard Smagorinsky LES closure model approximates the subgrid turbulent eddy viscosity using a mixing length-type model, where the LES grid filter size $\Delta$ provides a natural length scale. The subgrid eddy viscosity is modeled simply as (Smagorinsky [24])

$$\mu_t = \rho\left(C_s\Delta\right)^2|\tilde{S}|, \tag{2.140}$$

where the strain rate tensor magnitude is defined as $|\tilde{S}| \equiv (2\tilde{S}_{ij}\tilde{S}_{ij})^{\frac{1}{2}}$. The constant coefficient $C_s$ typically varies between $0.1$ and $0.24$ and should be carefully tuned to match the problem being solved (Rogallo and Moin [25]). It is assigned a value of $0.17$ here.

Although this model is desirable due to its simplicity and efficiency, care should be taken in its application. It is known to predict subgrid turbulent eddy viscosity proportional to the shear rate in the flow, independent of the local turbulence intensity. Non-zero subgrid turbulent eddy viscosity is even predicted in completely laminar regions of the flow, sometimes even preventing a natural transition to turbulence. Therefore, this model should only be used when this behavior will not adversely affect results.

### 2.6.8. Dynamic Smagorinsky LES Model

As mentioned in the previous section, the standard Smagorinsky model requires careful tuning of the constant model coefficient for the particular problem being simulated, and it is often overly-dissipative due to its inability to adapt to the local turbulent environment. Germano et al. [26] developed an improvement over the standard Smagorinsky model, where the coefficient $C_s$ is dynamically calculated based on the local turbulence field. A generalization of this method for variable-density flow is used here (Moin et al. [15]).

Similar to the standard Smagorinsky LES closure model, the subgrid eddy viscosity is modeled by the mixing length approximation

$$\mu_t = C_R \bar{\rho} \Delta^2 |\tilde{S}|, \tag{2.141}$$

where the strain rate tensor magnitude is defined as $|\tilde{S}| \equiv (2\tilde{S}_{ij}\tilde{S}_{ij})^{\frac{1}{2}}$. The coefficient $C_R$ is dynamically evaluated by taking advantage of scale similarity in the inertial range of the turbulence spectrum, near the minimum resolved scales. This is done by introducing a "test filter" which is identical to the grid filter defined in Equation 2.69 except for having a larger filter size denoted by $\hat{\Delta}$. The test filter of variable $\phi$ is denoted by $\hat{\phi}$.

The previously-defined subgrid stress tensor can be rewritten as

$$\begin{aligned} \tau_{u_i u_j} &\equiv -(\bar{\rho}\widetilde{u_i u_j} - \bar{\rho}\tilde{u}_i\tilde{u}_j) \\ &= -\left(\overline{\rho u_i u_j} - \frac{\overline{\rho u_i}\,\overline{\rho u_j}}{\bar{\rho}}\right) \end{aligned} \tag{2.142}$$

and an analogous larger-scale "subtest" stress $T_{u_i u_j}$ can be analogously defined as

$$T_{u_i u_j} \equiv -\left(\widehat{\overline{\rho u_i u_j}} - \frac{\widehat{\overline{\rho u_i}}\,\widehat{\overline{\rho u_j}}}{\hat{\bar{\rho}}}\right), \tag{2.143}$$

where the $\hat{\bar{()}}$ notation denotes resolved quantities that have been passed through the test filter. These two stresses can be related to each other through the algebraic identity of Germano [27],

$$L_{u_i u_j} \equiv T_{u_i u_j} - \widehat{\tau_{u_i u_j}} \tag{2.144}$$

$$= -\left(\widehat{\bar{\rho}\tilde{u}_i\tilde{u}_j} - \frac{\widehat{\bar{\rho}\tilde{u}_i}\,\widehat{\bar{\rho}\tilde{u}_j}}{\hat{\bar{\rho}}}\right). \tag{2.145}$$

Note that the right-hand side of Equation 2.145 is completely computable in terms of resolved quantities.

By modeling the two stresses in Equation 2.144 and equating them to Equation 2.145, the model coefficient $C_R$ can be dynamically evaluated. The subtest stress is modeled analogously to the subgrid stress, as

$$\tau_{u_i u_j} \approx 2C_R\bar{\rho}\Delta^2 |\tilde{S}| \left(\tilde{S}_{ij} - \frac{1}{3}\tilde{S}_{kk}\delta_{ij}\right) \tag{2.146}$$

$$T_{u_i u_j} \approx 2C_R\hat{\bar{\rho}}\hat{\Delta}^2 |\hat{\tilde{S}}| \left(\widehat{\tilde{S}_{ij}} - \frac{1}{3}\widehat{\tilde{S}_{kk}}\delta_{ij}\right), \tag{2.147}$$

where $C_R$ is assumed to be the same at both scales. The test-filtered strain rate tensor is defined similar to $|\tilde{S}|$ as

$$|\hat{\tilde{S}}| \equiv \left(2\widehat{\tilde{S}_{ij}}\widehat{\tilde{S}_{ij}}\right)^{\frac{1}{2}}. \tag{2.148}$$

Notice that when the modeled forms of $\tau_{u_iu_j}$ and $T_{u_iu_j}$ are substituted into Equation 2.144, $C_R$ appears inside a test filtering operation. Formally solving this system of equations for $C_R$ requires the expensive proposition of solving an additional set of coupled integro-differential equations (Ghosal et al. [28]). Alternatively, it is common practice to remove $C_R$ from the test filter with the assumption that it is varying slowly over distances on the order of the test filter size. This greatly simplifies calculations, although it yields a system of overdetermined equations for this single constant. The square of the error involved in this approximation is $Q = (L_{ij} - C_R M_{ij})^2$, where

$$L_{u_iu_j} = -\left(\widehat{\bar{\rho}\tilde{u}_i\tilde{u}_j} - \frac{\widehat{\bar{\rho}\tilde{u}_i}\,\widehat{\bar{\rho}\tilde{u}_j}}{\hat{\bar{\rho}}}\right) \tag{2.149}$$

$$M_{u_iu_j} = 2\hat{\bar{\rho}}\hat{\Delta}^2|\hat{\tilde{S}}|\left(\widehat{\tilde{S}_{ij}} - \frac{1}{3}\widehat{\tilde{S}_{kk}}\delta_{ij}\right) - \overline{2\bar{\rho}\Delta^2\,|\tilde{S}|\left(\tilde{S}_{ij} - \frac{1}{3}\tilde{S}_{kk}\delta_{ij}\right)}. \tag{2.150}$$

Minimizing this error in a least-squares fashion yields an expression for the modeled Smagorinsky coefficient (Lilly [29]),

$$C_R = \frac{L_{u_iu_j}M_{u_iu_j}}{M_{u_iu_j}M_{u_iu_j}}, \tag{2.151}$$

that can be used directly in Equation 2.141 for the subgrid turbulent eddy viscosity.

Due to the above simplifications, the model constant $C_R$ can sometimes fluctuate wildly to both large positive and negative values. These fluctuations can possibly lead to numerical instability, so they must be controlled. A common solution, and one that is taken here, is to pass the numerator and denominator of Equation 2.151 through the test filter, yielding

$$C_R = \frac{\widehat{L_{u_iu_j}M_{u_iu_j}}}{\widehat{M_{u_iu_j}M_{u_iu_j}}}. \tag{2.152}$$

This can be crudely justified by recognizing that $C_R$ was already assumed to vary slowly over distances equal to the test filter size, so that this filtering operation is simply enforcing that assumption.

This form of the dynamic Smagorinsky closure model allows energy backscatter, which is an intermittent transfer of turbulent kinetic energy from small scales to larger scales rather than the typical cascade from large to small scales. While backscatter can occur in real turbulent flows, the predicted negative eddy viscosities of the dynamic Smagorinsky model are more often attributable to model errors than to a real physical backscatter process. This can easily destabilize a simulation, so negative eddy viscosity is disallowed in the present formulation.

The only free parameter in the dynamic Smagorinsky closure model is the ratio between the test and grid filter sizes, $\alpha = \hat{\Delta}/\Delta$. Solutions are fairly insensitive to the choice of $\alpha$, although values of around $\alpha = 2$ are usually considered optimal (Germano et al. [26]). This ratio is dictated by the box filter formulation used in Fuego and the mesh topology selected by the user. The test filter volume for a

particular CVFEM node is defined as the volume of all surrounding finite elements that contain that node. (See Chapter 5 for more information about the CVFEM formulation.) On uniform hexahedral and uniform quadrilateral meshes, the test filter ratio will have a value of 2.0. The ratio will be around 1.59 for uniform tetrahedral meshes and around 1.73 for uniform triangular meshes, which are still reasonable values.

### 2.6.9. Subgrid-Scale Kinetic Energy One-Equation LES Model

The subgrid scale kinetic energy one-equation turbulence model, or $Ksgs$ model, represents a simple LES closure model. The transport equation for subgrid turbulent kinetic energy is given by

$$\int \frac{\partial \bar{\rho} k^{\text{sgs}}}{\partial t} \mathrm{d}V + \int \bar{\rho} k^{\text{sgs}} \tilde{u}_j n_j \mathrm{d}S = \int \frac{\mu_t}{\sigma_k} \frac{\partial k^{\text{sgs}}}{\partial x_j} n_j \mathrm{d}S + \int \left( P_k^{\text{sgs}} - D_k^{\text{sgs}} \right) \mathrm{d}V. \tag{2.153}$$

The production of subgrid turbulent kinetic energy, $P_k^{\text{sgs}}$, is modeled by Equation 2.105 while the dissipation of turbulent kinetic energy, $D_k^{\text{sgs}}$, is given by

$$D_k^{\text{sgs}} = C_\epsilon \bar{\rho} \frac{k^{\text{sgs}\,\frac{3}{2}}}{\Delta}, \tag{2.154}$$

where the grid filter length, $\Delta$, is given in terms of the grid cell volume by

$$\Delta = V^{\frac{1}{3}}. \tag{2.155}$$

The subgrid turbulent eddy viscosity is then provided by

$$\mu_t = C_{\mu_\epsilon} \Delta k^{\text{sgs}\,\frac{1}{2}}, \tag{2.156}$$

where the values of $C_\epsilon$ and $C_{\mu_\epsilon}$ are 0.845 and 0.0856, respectively.

Beta Capability:

### 2.6.10. Dynamic Subgrid-Scale Kinetic Energy One-Equation LES Model

Similar to the dynamic Smagorinsky model in Section 2.6.8, a dynamic version is developed for the subgrid kinetic energy model. The standard version with fixed coefficients over-predicts turbulent viscosity while the dynamic version is known to offer a better predictability. In Fuego, $C_\epsilon$ and $C_{\mu_\epsilon}$ are calculated dynamically which is considered to be a standard approach for the dynamic $Ksgs$ model [30]. The concept of "test filter" is identical to that of the dynamic Smagorinsky model in Section 2.6.8. Subgrid-scale kinetic energy for grid-filter and test-filter levels are

$$k^{\text{sgs}} = \frac{1}{2} \left( \widetilde{u_k u_k} - \tilde{u}_k \tilde{u}_k \right), \tag{2.157}$$

$$k^{\text{test}} = \frac{1}{2}\left(\widehat{\tilde{u}_k \tilde{u}_k} - \hat{\tilde{u}}_k \hat{\tilde{u}}_k\right). \tag{2.158}$$

Exact form of the dissipation $D_k^{\text{sgs}}$ in Equation 2.153 is

$$D_k^{\text{sgs}} = 2\bar{\mu}\left[\widetilde{S_{ij}^* D_{ij}^*} - \tilde{S}_{ij}^* \tilde{D}_{ij}^*\right] \tag{2.159}$$

where $S_{ij}^* = S_{ij} - \frac{1}{3} S_{kk}\delta_{ij}$, $D_{ij}^* = D_{ij} - \frac{1}{3} D_{kk}\delta_{ij}$, and $D_{ij} = \partial u_i / \partial x_j$. Meanwhile, $k^{\text{test}}$ dissipates by both molecular and turbulent viscosities of the grid-filtered level since the quantity is fully resolved in the test-filter level [30].

$$D_k^{\text{test}} = 2(\bar{\mu} + \mu_t)\left[\widehat{\tilde{S}_{ij}^* \tilde{D}_{ij}^*} - \widehat{\tilde{S}_{ij}^*}\,\widehat{\tilde{D}_{ij}^*}\right] \tag{2.160}$$

Using scale similarity, Equation 2.154 applies to the test-filter level as

$$D_k^{\text{test}} = C_\epsilon \hat{\bar{\rho}} \frac{k^{\text{test}\,\frac{3}{2}}}{\hat{\Delta}}, \tag{2.161}$$

and therefore, $C_\epsilon$ is calculated by

$$C_\epsilon = \frac{2(\bar{\mu} + \mu_t)\left[\widehat{\tilde{S}_{ij}^* \tilde{D}_{ij}^*} - \widehat{\tilde{S}_{ij}^*}\,\widehat{\tilde{D}_{ij}^*}\right]}{\widehat{\bar{\rho}} k^{\text{test}\,\frac{3}{2}} / \hat{\Delta}} \tag{2.162}$$

The other coefficient, $C_{\mu_\epsilon}$, is computed similarly to the Equation 2.151 as

$$C_{\mu_\epsilon} = \frac{L_{u_i u_j} M_{u_i u_j}}{M_{u_i u_j} M_{u_i u_j}}, \tag{2.163}$$

where $L_{u_i u_j}$ is defined identically to Equation 2.149 and $M_{u_i u_j}$ is simplified by

$$M_{u_i u_j} = 2\hat{\bar{\rho}}\hat{\Delta} k^{\text{test}\,\frac{1}{2}} \widehat{\tilde{S}_{ij}^*}. \tag{2.164}$$

Note that dynamic subgrid kinetic energy model does not require an additional filtering as in Equation 2.152.

### 2.6.11. Buoyancy Models for the Production Rate

There are three supported models that augment the production of turbulent kinetic energy via buoyancy contributions, buoyant vorticity generation [31], Rodi's [32], and de Ris' [33] buoyancy term.

The buoyant vorticity generation model has been developed and validated by Sandia National Laboratories group 9132 for use in large scale buoyant plumes. The model attempts to augment the

production of turbulent kinetic energy by adding a source term, $G_B$ to both the turbulent kinetic energy and dissipation rate equation that is related to the baroclinic torque,

$$G_B = \frac{C_{bvg}(\mu + \mu_t)||\frac{\partial \rho}{\partial x_j} X \frac{\partial P}{\partial x_j}||}{\rho^2}. \tag{2.165}$$

Please refer to Appendix 10 for a more detailed derivation of the model.

The buoyancy model of Rodi is given by

$$G_B = \beta \frac{\mu_t}{Pr_t} \frac{\partial T}{\partial x_j} g_j. \tag{2.166}$$

De Ris' buoyancy model offers two versions - flaming and non-flaming.

$$C_{deris}(\rho_\infty - \rho_f)\vec{g}k^{0.5} \qquad \text{flaming} \tag{2.167}$$

$$C_{deris}\Delta k^{0.5}(|\nabla \rho \times \vec{g}| - \nabla \rho \cdot \vec{g}) \qquad \text{non-flaming} \tag{2.168}$$

Default value of the user-defined coefficient $C_{deris}$ is 0.01. Note that ambient and flame density, rather than local density, matters on the flaming version.

In each model, derivatives are evaluated at the subcontrol volume center while the property values are lumped.

The right hand side of the turbulent kinetic energy equation for all model is $rhs+ = \int G_B dV$. For the dissipation rate equation, the source term is $rhs+ = \int C_{\epsilon 3} \frac{1}{T} G_B dV$ for the buoyant vorticity generation model while it is $rhs+ = \int C'_{\epsilon 1} C_{\epsilon 4} \frac{1}{T} G_B dV$ otherwise. Recall that the inverse time scale is determined by the turbulence model of choice, i.e., $\frac{\epsilon}{k}$ for the standard $k - \epsilon$ model and provided in Equation 2.114 for the $v^2$-$f$ model.

Note that the use of the buoyancy models hass not been evaluated with the $v^2$-$f$ model.

### 2.6.12. Turbulence closure model constants

For each of the afore-mentioned turbulence closure models, there are several constant coefficients which may be modified by the user in the input deck. Tables 2.6-1, 2.6-2, 2.6-3, and 2.6-4 list these parameters, their mapping to input deck names, and default values. Each of these default values may be modified by the user by specifying the respective `Turbulence Model Parameter` line in the `Global Constants` block under the Sierra domain.

## 2.7. WALL BOUNDARY CONDITIONS FOR TURBULENCE MODELS

### 2.7.1. Resolution of Boundary Layer; Momentum

The wall velocity boundary condition is the typical no-slip boundary; a specified value is expected.

**Table 2.6-1.. Constant parameters for $k-\epsilon$ turbulence models.**

| Turbulence Model | Symbol | User Input Name | Default Value |
|---|---|---|---|
| Standard $k-\epsilon$ | $C_\mu$ | Cmu | 0.09 |
| | $C_{\epsilon 1}$ | Ceps_1 | 1.44 |
| | $C_{\epsilon 2}$ | Ceps_2 | 1.92 |
| | $C_\chi$ | Cchi | 2.0 |
| | $\sigma_k$ | Sigma_K | 1.0 |
| | $\sigma_\epsilon$ | Sigma_E | 1.3 |
| Low Reynolds $k-\epsilon$ | $C_\mu$ | Cmu | 0.09 |
| | $C_{\epsilon 1}$ | Ceps_1 | 1.44 |
| | $C_{\epsilon 2}$ | Ceps_2 | 1.92 |
| | $\sigma_k$ | Sigma_K | 1.0 |
| | $\sigma_\epsilon$ | Sigma_E | 1.3 |
| | $A_\mu$ | Amu | 3.4 |
| RNG $k-\epsilon$ | $C_\mu$ | Cmu | 0.0837 |
| | $C_{\epsilon 1}$ | Ceps_1 | 1.42 |
| | $C_{\epsilon 2}$ | Ceps_2 | 1.68 |
| | $\sigma_k$ | Sigma_K | 0.7194 |
| | $\sigma_\epsilon$ | Sigma_E | 0.7194 |
| $v^2-f$ | $C_\mu$ | Cmu | 0.22 |
| | $C_{\epsilon 1}$ | Ceps_1 | 1.4 |
| | $C_{\epsilon 2}$ | Ceps_2 | 1.9 |
| | $\sigma_k$ | Sigma_K | 1.0 |
| | $\sigma_\epsilon$ | Sigma_E | 1.0 |
| | $C_1$ | CF_1 | 0.4 |
| | $C_2$ | CF_2 | 0.3 |
| | $\alpha$ | Alpha | 0.6 |
| | $C_T$ | Nseg | 6.0 |
| | $C_L$ | CL | 0.23 |
| | $C_\eta$ | Ceta | 70.0 |

## 2.7.2. Resolution of Boundary Layer; Turbulence Quantities

The resolution of the boundary layer is expected when the low Reynolds number or $v^2$-$f$ model is in use.

For the $v^2$-$f$ model, the wall turbulent kinetic energy and normal fluctuating stress component are each zero while the dissipation rate is given by

$$\epsilon_w = 2\nu \frac{\partial k^{1/2}{}^2}{\partial x_j} \ . \tag{2.169}$$

For the low Reynolds number, the wall turbulent kinetic energy is again zero while the dissipation rate, here considered to be the isotropic dissipation rate, is given as zero.

**Table 2.6-2.. Constant parameters for $k-\omega$ turbulence models.**

| Turbulence Model | Symbol | User Input Name | Default Value |
|---|---|---|---|
| $k-\omega$ | $\beta_0$ | Beta_Zero | 0.0708 |
| | $\beta^*$ | Beta_Star | 0.09 |
| | $\sigma_k$ | Sigma_K | $3/5$ |
| | $\sigma_\omega$ | Sigma_W | 0.5 |
| | $\gamma$ | Gamma | $13/25$ |
| | $C_{lim}$ | Clim | $7/8$ |
| SST | $A_1$ | A_One | 0.31 |
| | $\beta_1$ | Beta_One | 0.075 |
| | $\beta_2$ | Beta_Two | 0.0828 |
| | $\beta^*$ | Beta_Star | 0.09 |
| | $\gamma_1$ | Gamma_One | $5/9$ |
| | $\gamma_2$ | Gamma_Two | 0.44 |
| | $\sigma_{k1}$ | Sigma_K_One | 0.85 |
| | $\sigma_{k2}$ | Sigma_K_Two | 1.0 |
| | $\sigma_{\omega1}$ | Sigma_W_One | 0.5 |
| | $\sigma_{\omega2}$ | Sigma_W_Two | 0.856 |

**Table 2.6-3.. Constant parameters for LES turbulence models.**

| Turbulence Model | Symbol | User Input Name | Default Value |
|---|---|---|---|
| One-equation | $C_v$ | Cv | 0.5 |
| | $C_\epsilon$ | Ceps | 0.845 |
| | $C_{\mu_\epsilon}$ | Cmueps | 0.0856 |
| Standard Smagorinsky | $C_v$ | Cv | 0.5 |
| | $C_s$ | Cs | 0.17 |
| Dynamic Smagorinsky | $C_s$ | Cs | 0.17 |

## 2.7.3. Resolution of Boundary Layer; Enthalpy

The wall value of enthalpy is computed based on the specified temperature and either reference or local mass fractions. In the case of a heat flux boundary condition, the wall node value is computed based on the control volume balance.

## 2.7.4. Wall Functions for Turbulent Flow Boundary Conditions

Resolution of the near-wall turbulent boundary layer can require extensive mesh points. Adjacent to the wall exists an extremely thin viscous sublayer where these forces dominate and are relatively insensitive to free stream parameters. Following the viscous sublayer is a buffer layer, the so-called "log-layer" and, ultimately, the turbulent core. The Van Driest hypothesis of turbulent flow near solid boundaries can be used to derive the appropriate form of this log-law zone. In general, the use of wall

**Table 2.6-4.. Constant parameters for miscellaneous turbulence models. Default values may be changed using the $k-\epsilon$ model parameters input.**

| Model | Symbol | User Input Name | Default Value |
|---|---|---|---|
| Buoyant vorticity generation | $C_{BVG}$ | `Cbvg` | 0.35 |
| | $C_{\epsilon 3}$ | `Ceps_3` | 0.0 |
| Rodi's source term | $C_{\epsilon 4}$ | `C_eps4` | 0.0 |
| EDC laminar limit | $C_{\gamma,lam}$ | `Cgammalam` | 2.0 |
| | $C_{\tau,lam}$ | `Ctaulam` | 0.02 |
| | $C_{lam,trans}$ | `Clamtrans` | 40.0 |

functions eliminates the need to resolve the near wall layers by prescribing the wall shear stress and resulting force based on the law of the wall (Launder and Spalding [34]).

The primary assumptions of the law of the wall are

- local equilibrium of turbulent kinetic energy production and dissipation,
- constant shear stress within the log-law region,
- Couette flow (pure shear flow).

### 2.7.5. Wall Functions; Momentum

The wall shear stress enters the discretization of the momentum equations by the term

$$\int \tau_{ij} n_j dS = -F_{wi}. \tag{2.170}$$

Wall functions are used to prescribe the value of the wall shear stress rather than resolving the boundary layer within the near-wall domain. The fundamental momentum law of the wall formulation, assuming fully-developed turbulent flow near a no-slip wall, can be written as (Launder and Spalding [34])

$$u^+ = \frac{u_\parallel}{u_\tau} = \frac{1}{\kappa} \ln\left(E y^+\right), \tag{2.171}$$

where $u^+$ is defined by the the near-wall parallel velocity, $u_\parallel$, normalized by the wall friction velocity, $u_\tau$. The wall friction velocity is related to the turbulent kinetic energy by

$$u_\tau = C_\mu^{1/4} k^{1/2}. \tag{2.172}$$

by assuming that the production and dissipation of turbulence is in local equilibrium. Moreover, $y^+$ is defined as the normalized perpendicular distance from the point in question to the wall,

$$y^+ = \frac{\rho Y_p}{\mu} \left(\frac{\tau_w}{\rho}\right)^{1/2} = \frac{\rho Y_p u_\tau}{\mu} \tag{2.173}$$

The classical law of the wall is as follows:

$$u^+ = \frac{1}{\kappa}\ln(y^+) + C \tag{2.174}$$

where $\kappa$ is the von Karman constant and $C$ is the dimensionless integration constant that varies based on authorship and surface roughness. The above expression can be re-written as

$$u^+ = \frac{1}{\kappa}\ln(y^+) + \frac{1}{\kappa}\ln(\exp(\kappa C)) \tag{2.175}$$

or

$$u^+ = \frac{1}{\kappa}\left(\ln(y^+) + \ln(\exp(\kappa C))\right) \tag{2.176}$$

$$= \frac{1}{\kappa}\ln(Ey^+) \tag{2.177}$$

where $E$ is referred to in the text as the dimensionless wall roughness parameter and is described by

$$E = \exp(\kappa C) \tag{2.178}$$

In Fuego, $\kappa$ is set to the value of 0.42 while the value of $E$ is set to 9.8 for smooth walls[1]. The viscous sublayer is assumed to extend to a value of $y^+ = 11.63$.

The wall shear stress, $\tau_w$, can be expressed as

$$\tau_w = \rho u_\tau^2 = \rho u_\tau \frac{u_\parallel}{u^+} = \frac{\rho\kappa u_\tau}{\ln\left(Ey^+\right)}u_\parallel = \lambda_w u_\parallel, \tag{2.179}$$

where $\lambda_w$ is simply the grouping of the factors from the law of the wall. For values of $y^+$ less than 11.63, the wall shear stress is given by

$$\tau_w = \mu\frac{u_\parallel}{Y_p}. \tag{2.180}$$

The force imparted by the wall, for the $i_{th}$ component of velocity, can be written as

$$F_{w,i} = -\lambda_w A_w u_{i\parallel}, \tag{2.181}$$

where $A_w$ is the total area over which the shear stress acts.

The use of a general, non-orthogonal mesh adds a slight complexity to specifying the force imparted on the fluid by the wall. As shown in Equation 2.181, the velocity component parallel to the wall must be determined. Use of the unit normal vector, $n_j$, provides an easy way to determine the parallel velocity component by the following standard vector projection,

$$\Pi_{ij} = \left[\delta_{ij} - n_i n_j\right]. \tag{2.182}$$

Carrying out the projection of a general velocity, which is not necessarily parallel to the wall, yields the velocity vector parallel to the wall,

$$u_{i\parallel} = \Pi_{ij}u_j = u_i\left(1 - {n_i}^2\right) - \sum_{j=1;j\neq j}^{n} u_j n_i n_j. \tag{2.183}$$

[1] White [35] suggests values of $\kappa = 0.41$ and $E = 7.768$.

Note that the component that acts on the particular $i^{th}$ component of velocity,

$$-\lambda_w A_w \left(1 - n_i n_i\right) u_{i\|}, \tag{2.184}$$

provides a form that can be potentially treated implicitly; i.e., in a way to augment the diagonal dominance of the central coefficient of the $i^{th}$ component of velocity. The use of residual form adds a slight complexity to this implicit formulation only in that appropriate right-hand-side source terms must be added.

### 2.7.6. Wall Functions; Turbulent Kinetic Energy

The near wall turbulent kinetic energy can be obtained by two different procedures. The most common approach is to solve a transport equation for the near wall value of turbulent kinetic energy with a modified production and dissipation term on the right hand side of the turbulent kinetic energy equation, Equation 2.102. As will be shown below, the form of the near wall production and dissipation term are determined based on equilibrium arguments, i.e., $P_k = \rho\epsilon$.

Another common approach is to assign the value of turbulent kinetic energy that strictly results in the equality $P_k = \rho\epsilon$. In this formulation, it is assumed that the convection and diffusive flux is zero across the control volume.

Both procedures, which formally do not address the role of buoyancy production, begin with the determination of the near wall value of the production of turbulent kinetic energy. The turbulent kinetic energy production term is consistent with the law of the wall formulation and can be expressed as

$$P_{kw} = \tau_w \frac{\partial u_\|}{\partial y}. \tag{2.185}$$

The parallel velocity, $u_\|$, can be related to the wall shear stress by

$$\tau_w \frac{u^+}{y^+} = \mu \frac{u_\|}{Y_p}. \tag{2.186}$$

Taking the derivative of both sides of Equation 2.186, and substituting this relationship into Equation 2.185 yields,

$$P_{kw} = \frac{\tau_w^2}{\mu} \frac{\partial u^+}{\partial y^+}. \tag{2.187}$$

Applying the derivative of the law of the wall formulation, Equation 2.171, provides the functional form of $\partial u^+ / \partial y^+$,

$$\frac{\partial u^+}{\partial y^+} = \frac{\partial}{\partial y^+}\left[\frac{1}{\kappa}\ln\left(Ey^+\right)\right] = \frac{1}{\kappa y^+}. \tag{2.188}$$

Substituting Equation 2.188 within Equation 2.187 yields a commonly used form of the near wall production term,

$$P_{kw} = \frac{{\tau_w}^2}{\rho\kappa u_\tau Y_p}. \tag{2.189}$$

Assuming local equilibrium, $P_k = \rho\epsilon$, and using Equation 2.189 and Equation 2.172 provides the form of the near wall turbulence dissipation,

$$\epsilon = \frac{u_\tau^3}{\kappa Y_p} = \frac{C_\mu^{3/4} k^{3/2}}{\kappa Y_p}, \tag{2.190}$$

while the form of the wall shear stress is given by,

$$\tau_w = \rho C_\mu^{1/2} k \tag{2.191}$$

Under the above assumptions, the near wall value for turbulent kinetic energy, in the absence of convection, diffusion, or accumulation is given by,

$$k = \frac{u_\tau^2}{C_\mu^{1/2}}. \tag{2.192}$$

If the second method (Dirichlet condition on near wall turbulent kinetic energy) is to be used, the value of the wall friction velocity, $u_\tau$, can be obtained in an iterative manner (Sondak and Pletcher [36]) by use of Equation 2.171. This method has been used and shown to be satisfactory (Elkaim [37]) and strictly enforces the assumptions of the law of the wall that have already been outlined.

In the method that elects to solve a near wall turbulent kinetic energy transport equation, the production and dissipation terms in the turbulent kinetic energy transport equation are [potentially] given by Equation 2.189 and

$$-\rho\epsilon = -\rho\frac{C_\mu^{3/4} k^{3/2}}{\kappa Y_p}, \tag{2.193}$$

Unfortunately, there does not seem to be one universal description of the near wall turbulent kinetic energy production term and dissipation term, Equation 2.189 and 2.193, respectively. For example, in the law of the wall formulation, given by Launder and Spalding [38], the near wall production term is given by,

$$P_{kw} = \tau_w \frac{u_\parallel}{y_p}. \tag{2.194}$$

In this formulation, the wall shear stress is given by the law of the wall formulation, Equation 2.179, providing the value of $y^+$ is greater than 11.63 (otherwise, it is given by the laminar shear stress, Equation 2.180). The dissipation term, $-\rho\epsilon$ is given by

$$-\rho\epsilon = -\rho\frac{C_\mu^{3/4} k^{3/2}}{\kappa Y_p} \ln Ey^+. \tag{2.195}$$

Note that in the absence of convection, diffusion or accumulation, the above two forms of the near wall production and dissipation source terms revert to Equation 2.192. Therefore, if the modeled flow is consistent with the law of the wall formulations, all methods should yield similar limiting behavior. Under conditions of non equilibrium, i.e., a separated flow, or values of $y^+$ within the viscous sublayer, some models may perform better. However, it is important to note that if the flow to be simulated includes separation and reattachment, or the computation mesh is such that $y^+$ is within the viscous sublayer, the law of the wall formulation can provide non sensical results.

In Fuego, there are currently two general supported methods from which to choose when applying the near wall turbulent kinetic energy boundary condition. The first method, which can be activated by the command line `omit near wall turbulent ke transport equation`, is the form of Equation 2.192 that enforces a Dirichlet condition. The second method is to solve a full control volume balance for the near wall turbulent kinetic, with convection and diffusion terms, with a modified production and dissipation term given by either

- Equations 2.189 and 2.190.

- Equations 2.194 and 2.195

The use of Equations 2.189 and 2.190 can be activated by the command line (within the wall bc block) `use equilibrium production model` which is based on the ability to express the wall shear stress consistent with the assumptions of full equilibrium between production and dissipation, Equation 2.191. In all cases that do not set a Dirichlet condition for the turbulent kinetic energy, the assembled buoyancy source terms are not removed.

### 2.7.7. $k$-$\omega$ $SST$ Wall Functions; Turbulent Kinetic Energy

When a Dirichlet condition is not set for turbulent kinetic energy, the approach in modifying the near wall production and dissipation terms is followed.

In this approach, the equation for $k$ is solved near the wall to remove the assumptions of log layer flow one level. However, we invoke the log layer assumption to write,

$$P_k = \frac{\tau_w^2}{\rho \kappa u_\tau Y_p}. \tag{2.196}$$

Balancing production and dissipation in the $k-\omega$ model allows us to write,

$$P_k = \rho \frac{u_\tau^3}{\kappa Y_p} = \rho \frac{(\beta')^{3/4} k^{3/2}}{\kappa Y_p}. \tag{2.197}$$

The dissipation rate is also modified accordingly such that the production equality with dissipation is retained. An alternative method is to use the approximation of of Launder and Spaulding which prescribes production as,

$$P_k = \tau_w \frac{u_{||}}{Y_p}. \tag{2.198}$$

In practice, this formulation seems to be less stable since the production and dissipation terms are now in near-equilibrium.

## 2.7.8. Wall Functions; Turbulence Dissipation Transport

Consistently within the literature, the near wall turbulence dissipation is assigned the Dirichlet value given by Equation 2.190. Frequently, this expression is lagged by one subiteration in an effort to maintain consistency between the Dirichlet wall condition and the freezing of the $\epsilon/k$ ratio of the turbulence dissipation equation, Equation 2.103.

## 2.7.9. Wall Functions; Turbulent Frequency Transport

### *2.7.9.1. Low Reynolds Number Treatment*

The low Reynolds approach for $k-\omega$ uses a sequence of Dirichlet conditions similar to what is used for $k-\epsilon$. However, unlike the latter, $k-\omega$ requires no extra damping terms near the wall. When the wall is resolved, exact Dirichlet conditions are known for both the velocity and $k$:

$$\vec{u} = 0, \qquad k = 0. \tag{2.199}$$

A Dirichlet condition is also used on $\omega$. While the $k-\epsilon$ model is rendered less stable because $k$ appears in this boundary condition, the $\omega$ equation depends only on the near-wall grid spacing. The boundary condition is

$$\omega = \frac{6\nu}{\beta y^2}, \tag{2.200}$$

which is valid for $y^+ < 3$. Above, $\beta$ depends on the model type. If SST is in use, $\beta = \beta_1$ while if the Wilcox model is in use, $\beta = \beta_0$.

### *2.7.9.2. High Reynolds Number Treatment*

The high Reynolds approach is also quite similar to the $k-\epsilon$ model except $\omega$ is handled differently.

### *2.7.9.3. Automatic Wall Functions*

Because $\omega$ has analytic solutions in both the log layer and viscous sub-layer, an automatic treatment is developed that blends those two solutions to provide Dirichlet conditions for all $y$. Let $\omega_h$ be the high Reynolds number formulation and $\omega_l$ be the low Reynolds version. Then the Dirichlet condition on $\omega$ is

$$\omega = \omega_l \sqrt{1 + \left(\frac{\omega_h}{\omega_l}\right)^2}. \tag{2.201}$$

However, $u_\tau$ for the high Reynolds $\omega$ value is computed based on the parallel velocity: The velocity equation is augmented by a traction force based on the friction velocity $u_\tau$. This quantity may be solved for iteratively using the law of the wall. A Dirichlet condition is also used for $k$, assuming it is in the log region, which is similar to the $k-\epsilon$ model:

$$k = \frac{u_\tau^2}{\sqrt{\beta^*}}. \tag{2.202}$$

In the case of $\omega$, an analytic expression is known in the log layer:

$$\omega = \frac{u_\tau}{\sqrt{\beta^*}\kappa y}, \tag{2.203}$$

which is independent of $k$. Note that some implementations use a predefined constant instead of $\sqrt{\beta'}$, although the standard values are consistent with these expressions. Because all these expressions require $y$ to be in the log layer, they should absolutely not be used unless it can be guaranteed that $y^+ > 10$, and $y^+ > 25$ is preferable.

$$u_\tau = \sqrt{\nu \left| \frac{u_{||}}{y} \right|}. \tag{2.204}$$

The automatic wall function approach is obtained by removing the "omit near wall turbulent ke equation" line command and activating either the SST or KW turbulence models.

## 2.7.10. Wall Functions; Enthalpy Transport

For non-adiabatic boundaries, heat loss to the wall must be considered. The use of the Reynolds analogy provides a functional form of the energy transport similar to the that of the logarithmic law-of-the-wall momentum formulation. The thermal boundary layer is modeled either as a linear profile ($y^+ < 11.63$) where the thermal boundary layer is dominated by conduction or a logarithmic profile where the effects of turbulence dominate over thermal conduction, Versteeg and Malalasekera [39].

The law-of-the-wall used in Fuego has the following form,

$$q_w = \frac{\rho \left( h_w - h_p \right) u_\tau}{T^+}, \tag{2.205}$$

where

$$T^+ = \sigma_T \left[ u^+ + P \right]. \tag{2.206}$$

The role of $T^+$ is to account for the fact that the thickness of the thermal conduction layer is [practically] of a different size than that of the viscous sublayer (momentum).

In the above equation, $P$ is the universal "P function" (Jayatilleke [40]) and can be expressed as a function of the molecular and turbulent Prandtl number,

$$P = 9.24 \left[ \left( \frac{\sigma}{\sigma_T} \right)^{0.75} - 1 \right] \left( 1 + 0.28 exp \left[ -0.007 \frac{\sigma}{\sigma_T} \right] \right), \tag{2.207}$$

where $\sigma_T$ and $\sigma$ represent the turbulent and molecular Prandtl number, respectively.

Therefore, it is seen that the so-called "P function" is the parameter that functionally changes the thickness of the thermal conduction layer from that of the viscous sublayer. For example, if one were to model a high-Prandtl number fluid such as common vegetable oil, one would note that the thickness of the viscous sublayer is far greater than that of the thermal sublayer. However, for low-Prandtl number fluids, the opposite is true. The subsequent value of $T^+$ ensures this functionality.

In the case of a user defined heat flux at a wall boundary condition, the full quantity is assembled as a right-hand-side source term. As a post processing step, Equation 2.205 (in temperature form) is rearranged to provide the wall temperature. In practice, the heat flux boundary condition block is to be defined on an already defined wall boundary condition block (without temperature specification). In this manner, multiple boundary conditions are "painted" on a particular sideset.

### 2.7.11. Wall Functions; Scalar Transport

Wall functions for use in a convective diffusive problem, e.g., diffusional transport of fuel (through multicomponent evaporation) from a jet fuel pool, are not currently supported.

## 2.8. INLET CONDITIONS FOR TURBULENCE QUANTITIES

### 2.8.1. Turbulent Kinetic Energy

The inlet turbulent kinetic energy must be specified for any simulation that involves a velocity-specified inlet. If actual values of the inlet turbulent kinetic energy are not available, then a suitable value based on basic definitions is used. In general, the kinetic energy associated with the turbulent flow is defined by,

$$k = \frac{1}{2}\left(\overline{u'^2} + \overline{v'^2} + \overline{w'^2}\right). \tag{2.208}$$

The turbulence intensity $T_i$, is related to the kinetic energy by,

$$T_i = \frac{\left(\frac{2}{3}k\right)^{1/2}}{U_{ref}}. \tag{2.209}$$

Rearranging Equation 2.209 for the turbulent kinetic energy yields a working form for the specification of inlet turbulent kinetic energy based on a reference velocity, $U_{ref}$,

$$k = \frac{3}{2}\left(U_{ref}T_{in}\right)^2. \tag{2.210}$$

The value of $U_{ref}$ can typically be taken to be the magnitude of the velocity.

### 2.8.2. Turbulence Dissipation Rate

As with the turbulent kinetic energy inlet condition for specified velocity, the inlet value of the turbulence dissipation rate must also be specified. If values are known, for instance based on experimental data, then the available data should be used. Otherwise, the following assumed form of the turbulence dissipation rate is used,

$$\epsilon = C_\mu^{3/4}\frac{k^{3/2}}{l}, \tag{2.211}$$

where $l = 0.07L$; $L$ represents a characteristic length scale of an inlet eddy and $k$ represents the inlet turbulent kinetic energy as determined above.

# 2.9. EDC TURBULENT COMBUSTION MODEL

The combustion submodel is Magnussen's Eddy Dissipation Concept (EDC) and development details can be found in Magnussen, et al. [41], Magnussen [42], Byggstyøl and Magnussen [43], Magnussen [44], Lilleheie, et al. [45], and Gran and Magnussen [46].

## 2.9.1. Model Characteristics

The underlying assumption in the EDC model is that combustion in turbulent flows is controlled by turbulent mixing. The combustion model is an algebraic zone-type model and is influenced by local cell (control volume) values only. The model derivation assumes that the minimum cell dimension is large relative to the thickness of a flame (reaction zone) structure. This thickness varies with strain-rate, but the cell size should not be less than a few millimeters. The equations are not valid for laminar or near-laminar flow, but are based on fully developed turbulence arguments. The turbulent combustion model uses information from three sources: 1) thermochemistry, 2) species and state information from the cell values, and 3) turbulence kinetic energy and dissipation. From these data, the model creates source/sink terms for species equations and the energy equation (via radiative transport).

The model function is to provide an integral effect of combustion processes occurring within the control volume for the duration of a time-step. In this manner, reaction zone structures are not resolved, but the aggregate effect of turbulent combustion is modeled. To model the integral effect, two homogeneous zones are defined within each control volume for which there is combustion, as shown in Figure 2.9-1. The zones are termed the reaction zone (fine structures) and the surrounding zone. The size and mass exchange rate between these zones are influenced by the local turbulence properties and are the principal means by which turbulent fluctuations are accounted for within the model. The assumption that each zone is homogeneous is equivalent to assuming that the mixing within each zone is instantaneous. Since combustion occurs within (but is not limited to) the reaction zone, the assumptions for combustion correspond to those for a perfectly stirred reactor (PSR). Slower reactions can also occur in the surroundings, in which case, the assumptions for reaction in the surroundings are also consistent with PSR assumptions.

## 2.9.2. Physical Interpretation

Magnussen's EDC model is derived to be a general combustion model for premixed to non-premixed scalar fields and for high to moderate turbulence levels. It is not intended to be used for laminar combustion. Magnussen's physical interpretation of combustion is based on the concept that chemical reaction occurs in regions of the flow in which the dissipation of turbulent energy takes place, i.e., fine structure regions. These regions are concentrated in isolated volumes and represent a small fraction of

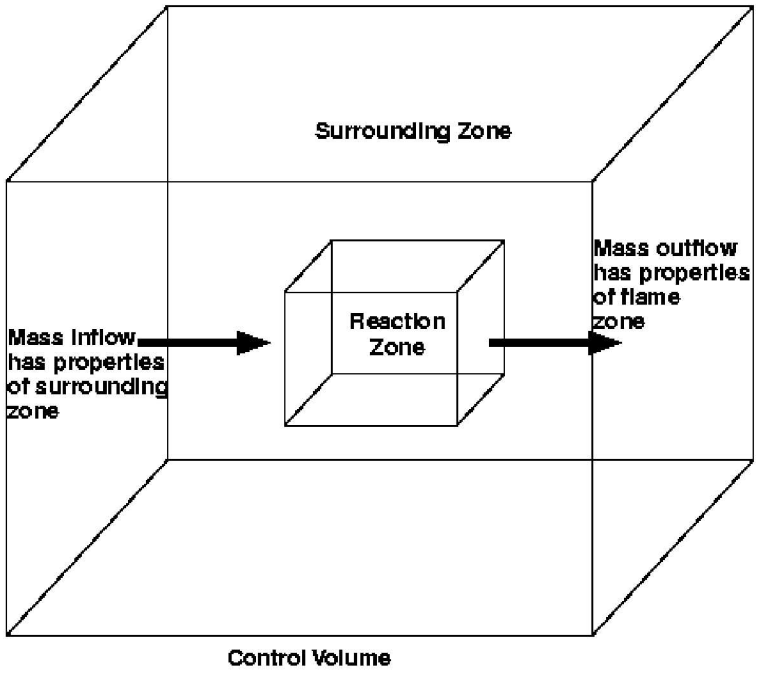


**Figure 2.9-1.. Model geometry for Magnussen's Eddy Dissipation Concept. The control volume is comprised of two zones; the properties of each zone are assumed to be adequately represented by a single set of values (i.e., lumped or perfectly stirred). The mass exchange between the zones is controlled by turbulent mixing.**

the flow. The regions have characteristic dimensions that are of the Kolmogorov length scale in one or two dimensions, but not the third.

Fires are buoyant flows. Turbulent fires tend to be large, having base diameters above a meter. The turbulent length scales are large and the flow velocities are relatively slow, on the order of meters to tens of meters per second. (Still photographs of reaction zone structure within large fires can be found in Tieszen, et al. [47]). Therefore, turbulence levels tend to be moderate. Near the base of a fire, the combustion zone can be characterized as a continuous wrinkled flame sheet that appears to wrap around larger turbulent structures. The basic combustion mode is that of a strained diffusion flame with large surface area due to the turbulence. At higher elevations in the fire, turbulence levels increase and the character may change. Premixed combustion is possible as unburned products in the smoke are re-entrained into the fire. While Magnussen's model was originally derived in terms of high turbulence levels resulting in fine structure regions (i.e., localized regions of high vorticity at dissipation scales), the model is appropriate for moderate turbulent intensities that occur in fires.

Figure 2.9-2 shows the physical geometry from which the combustion model will be derived for fires.

Turbulence controls the reaction and surrounding volume fractions and fuel mass transport per unit volume. In general, turbulent momentum exchange processes result in scalar stirring at all length scales down to molecular mixing processes which are diffusion controlled. Without length scale information below the grid scale of the computation, it is impossible to correctly represent the interactions between all the relevant physical processes at their relevant length scales.

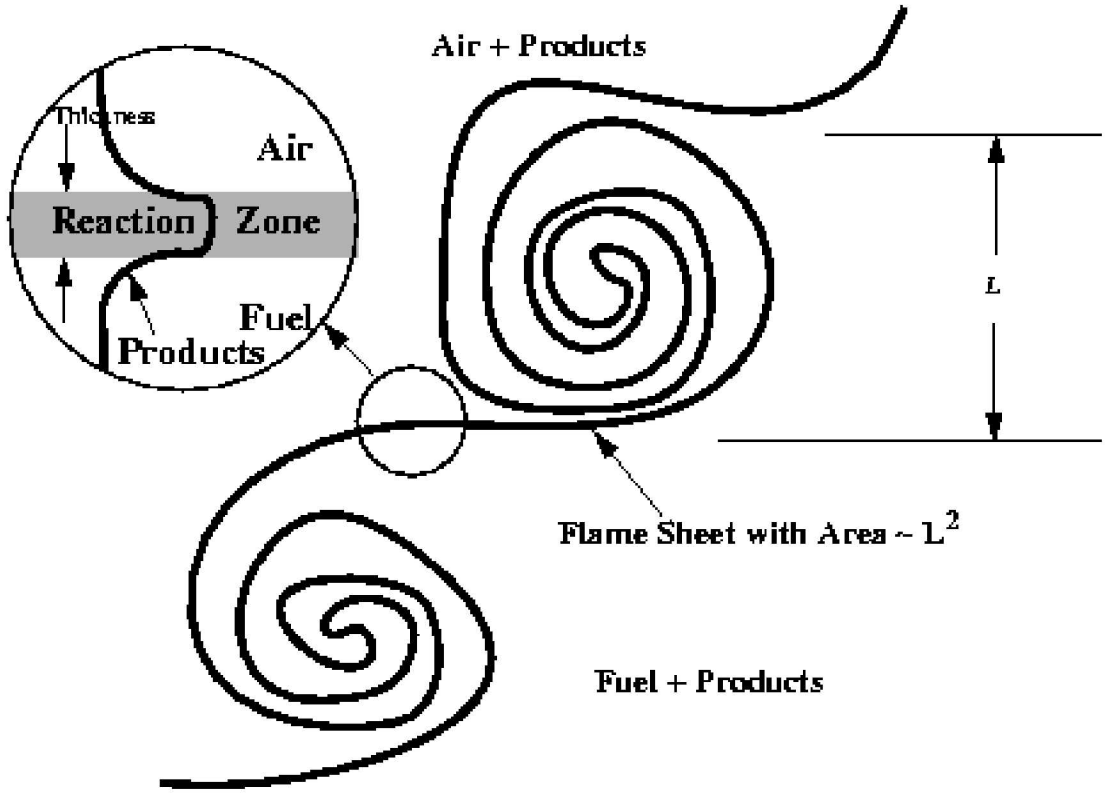


**Figure 2.9-2.. Assumed flame surface geometry. $L$ is the integral turbulent length scale. The reaction zone thickness is characterized by the Kolmogorov dissipative turbulent length scale, $\eta$.**

Magnussen's EDC model attempts to represent the mixing processes that are most important to the overall heat release from combustion. It it based on the assumption that the overall heat release rate is controlled by the mass transport into the reaction zone. Therefore, considerable effort is made to model turbulent momentum processes that affect mass transport into the reaction zone. In the surrounding gases, turbulent mixing occurs with (in all likelihood) a similar vigor, however, its effect on the combustion rate is considered less important since the turbulence is not directly contributing to mass transport into the reaction zone. For this reason, there are two different levels of mixing assumptions made within the model.

With respect to Figure 2.9-1, the turbulence level in each control volume is taken into account in the consideration of the mass exchange between the reaction zone and the surrounding zone. However,

within each zone, it is assumed that the properties are instantaneously homogeneous and uniform, i.e., perfectly stirred. This perfectly stirred assumption obviously over-predicts mixing within each zone for any real level of turbulence, and only begins to approximate reality at the highest levels of turbulence. On the other hand, the perfectly stirred assumption allows point calculations to be made in each zone for conveniently determining thermochemical properties. Without this assumption, it would be necessary to specify the gradients within each zone and integrate the specified gradients throughout the cell to obtain cell averaged property information. The approach here is to assume that over-predicting mixing within each zone via the perfectly stirred assumption has only a secondary effect on heat release rates within each cell.

### 2.9.3. Thermochemistry

Within the current strategy, chemical reaction can occur in both zones. However, in the simplest case, no reaction occurs within the surroundings due to the low temperature and unmixedness; all reaction occurs within the reaction zone. The notion of zones, perfect stirring within the zones, and type of chemistry involved are all independent assumptions, but have interrelated consequences. For example, finite-rate chemistry involving hundreds or thousands of species could be considered within the zones. From the perfectly stirred assumption within each zone, the finite-rate chemistry would be calculated as if it were occurring in a perfectly stirred reactor. In a real diffusion reaction, there are spatial variations in species concentrations for real turbulence levels so that the various chemical pathways, as well as heat, mass, and momentum transport, in a real strained diffusion flame can be quantitatively different than those calculated on the basis of perfect stirring. This effect is probably the strongest disadvantage of the perfectly stirred assumption. Only in the limit of infinitely-fast turbulent mixing does perfect stirring actually exist. In practice, the computation of detailed, finite-rate chemistry concurrently with a three-dimensional fluid mechanics calculation is expensive. Except in the limit where the turbulent strain rate is high enough that finite rate chemistry is warranted, it is adequate to use simpler descriptions of the chemistry. In the case of high strain rates, precalculation of the chemistry is usually done and the results tabulated in a look-up table to determine extinction limits.

For the current implementation, it is assumed that the chemistry can be represented as irreversible, “infinitely-fast” reactions that occur within each reactor. In classical combustion studies, the concept of “infinitely-fast” reactions is not usually invoked in the context of a perfectly stirred reactor. In the context of the current model, the meaning of an “infinitely-fast” reaction in the flame zone (a perfectly stirred reactor) is that the reactant stream entering the reaction zone is converted to products instantly as it enters the zone, and then the products are mixed instantly throughout the zone. The zone then reflects the thermodynamic properties of the combustion products at the adiabatic flame temperature for a given composition while the surrounding zone has the properties of reactants (and possibly previously combusted products) near the cell temperature.

In general, if the turbulent mass exchange rate between the zones (i.e. strain-rate) is sufficiently high that infinitely-fast chemistry assumptions do not apply, then finite-rate reactions within the perfectly stirred reactor can be used. Residence time scales that warrant finite-rate considerations tend to be at the sub-millisecond level. In the current implementation, the case of high turbulence levels leading to blow-out of a reactor is treated as a limits test. The test method is discussed in Section 2.9.9.

In principle, it is not necessary to assume irreversible chemistry within each zone. At long time scales (i.e., low turbulence levels), chemical equilibrium will result. The use of irreversible chemistry avoids the need to calculate the equilibrium state of the forward and reverse reactions for every combusting cell at every time step. For the current implementation, the time savings is deemed to be worth the cost in accuracy.

Regardless of the assumptions about chemistry employed in modeling the reaction zone, the actual reaction zones in a fire will very likely be similar to strained diffusion flames (wrinkled flame sheets wrapped into vortical structures). Perhaps higher in a fire with the re-entrainment of smoke, partially premixed combustion can occur. For diffusion reactions, combustion occurs within a region encompassing a stoichiometric surface between fuel and air. Therefore, the reaction zone is modeled as occurring with stoichiometric reactions. The reactants being transported into the reaction zone via turbulent mixing come from the surroundings zone and thus have the composition of the surroundings. There will be a limiting amount of one reactant if the combustion is to occur at off-stoichiometric conditions. The excess of the other reactant, prior products, and inerts do not participate in chemical reactions, but are transported in and out of the combustion zone by turbulent mixing. However, their presence affects the zone properties (for example, through their heat capacity).

Combustion products are transported into the surroundings at the same rate as the reactants are transported into the reaction zone (conservation of mass). However, the perfect stirring assumption for properties means that these products have uniform properties. In a diffusion reaction, products mix with fuel on one side of the reaction zone and air on the other. On the fuel side of the reaction zone, significant amounts of CO and soot can result from interaction between the inflowing fuel and outflowing products. The formation of CO is important not only from a toxic pollutant perspective but its formation results in significantly less heat release and lower temperatures. Given the limits of a two-zone model with perfect mixing within each zone, there is no simple way to model both stoichiometric combustion and the formation of CO on the fuel side of the reaction. In the current formulation, an ad hoc approach is used in which combustion in the reaction zone is assumed to occur in sequential steps, each of which is irreversible and infinitely fast. The first step is stoichiometric oxidation of the fuel species to CO and $H_2$ products. The second step is the oxidation of CO and $H_2$ to $CO_2$ and $H_2O$ provided there is excess $O_2$ in the reactant stream. If the overall stoichiometry in the control volume is fuel rich, significant amounts of CO and $H_2$ will be formed, while if it is lean only $CO_2$ and $H_2O$ will be formed.

### 2.9.4. Chemical Mechanism

For an arbitrary CHNO fuel, the stoichiometric, irreversible reaction to CO and $H_2$ products is given by

$$\begin{aligned} \mathrm{C}_m\mathrm{H}_n\mathrm{N}_p\mathrm{O}_q + \left(\frac{m-q}{2}\right)\mathrm{O}_2 \quad &+ \quad \sum(\zeta_D)\,\text{Diluent} \quad \Rightarrow \\ (m)\,\mathrm{CO} \quad &+ \quad \left(\frac{n}{2}\right)\mathrm{H}_2 + \left(\frac{p}{2}\right)\mathrm{N}_2 + \sum(\zeta_D)\,\text{Diluent}, \end{aligned} \tag{2.212}$$

where $m$, $n$, $p$, and $q$ are the numbers of carbon, hydrogen, nitrogen, and oxygen atoms within the fuel molecule, respectively, and the terms in parentheses are the stoichiometric coefficients. The summation

term for diluents includes all other species present in the reaction stream including nitrogen in air, combustion products in the surroundings from previous combustion processes, etc… Diluents, including the combustion products, are assumed to have no effect on the chemical reaction itself. However, diluents do have an effect on the temperature rise through their specific heats and the presence of products is used as an ignition criteria for the combustion model.

The assumption that combustion products act like diluents (i.e., have no effect on the reaction) is obviously a simplification. Product species include CO, $H_2$, $CO_2$, and $H_2O$. The presence of CO and $H_2$ in the reactant stream would affect equilibrium results; however, irreversible reactions have already been assumed in the model so the presence of these species does not represent an additional simplification. On the other hand, the presence of large amounts of $CO_2$ and $H_2O$ in the reactant stream may reduce the amount of $O_2$ consumed for a given amount of fuel due to partial oxidation of the products via the oxygen in the $CO_2$ and $H_2O$ in an overall fuel rich environment. However, this effect is partially compensated since the extra $O_2$ would be consumed by the second reaction.

The second reaction is the subsequent oxidation of CO and $H_2$ to $CO_2$ and $H_2O$. This reaction oxidizes both the CO and $H_2$ produced by the first reaction and any CO and $H_2$ that passed through the first reaction as products (i.e., diluent). The reaction is given by

$$\begin{aligned}(m)\,\mathrm{CO} + \left(\frac{n}{2}\right)\mathrm{H_2} \quad &+ \quad \left(\frac{p}{2}\right)\mathrm{N_2} + \left(\frac{m}{2}+\frac{n}{4}\right)\mathrm{O_2} + \sum(\zeta_d)\,\text{Diluent} \quad \Rightarrow \\ (m)\,\mathrm{CO_2} \quad &+ \quad \left(\frac{n}{2}\right)\mathrm{H_2O} + \left(\frac{p}{2}\right)\mathrm{N_2} + \sum(\zeta_d)\,\text{Diluent}. \end{aligned} \tag{2.213}$$

In the current implementation, soot is considered to be a trace species. As such, its mass and energetics are not considered part of the above chemical reactions. Soot has its own production terms and is considered to oxidize in proportion to the fuel oxidation in the first reaction. See the soot model in Section 2.12 for details.

### 2.9.5. Species Consumption/Production Limits

The reactants being transported into the reaction zone come from the surroundings and therefore have the same composition as the surroundings. As such, the reaction can only proceed within the limits of available fuel and oxygen from the reactant stream. For example, if there is insufficient oxygen in the reactant stream, then all of the oxygen will be consumed by Reaction 1, (Equation 2.212), and the excess fuel will be passed with products from Reaction 1 to Reaction 2, (Equation 2.213). Reaction 2 will not take place because all the oxygen was consumed in Reaction 1 (i.e., in both reactions, oxygen is limiting). If there is insufficient fuel in Reaction 1, then all the fuel will be consumed and excess oxygen will be passed to Reaction 2. Depending on the ratio of oxygen to CO and $H_2$, all the secondary fuels may be consumed or all the oxygen may be consumed.

To find the limiting mass, it is convenient to define an equivalence ratio. Equivalence ratios are normally defined in terms of molar ratios, but mass ratios yield the same result [48] and are preferred here since

mass fractions are used in the transport equations.

$$\Phi = \frac{\left(\frac{Y_{fuel}}{Y_{oxy}}\right)_{mix}}{\left(\frac{Y_{fuel}}{Y_{oxy}}\right)_{stoic}} \tag{2.214}$$

The numerator is the ratio of the actual mass of fuel to oxygen in the reactant stream,

$$\left(\frac{Y_{fuel}}{Y_{oxy}}\right)_{mix} = \left.\frac{\text{mass Fuel}}{\text{mass Oxygen}}\right|_{\text{mix}}. \tag{2.215}$$

The denominator is determined for each reaction. Generically, the first and second reactions have the following form

$$\sum\left(\zeta_{fuel}\right)\text{Fuel} + \zeta_{O_2}O_2 + \sum\left(\zeta_D\right)\text{Diluent} \rightarrow \sum\left(\zeta_{prod}\right)\text{Product} + \sum\left(\zeta_D\right)\text{Diluent}, \tag{2.216}$$

where $\zeta$ are stoichiometric coefficients on a molar basis. The stoichiometric fuel to oxygen mass ratio is

$$\left.\frac{Y_{fuel}}{Y_{oxy}}\right|_{stoic} = \frac{\sum W_{fuel}\left(\zeta_{fuel}\right)}{W_{O_2}\left(\zeta_{O_2}\right)}, \tag{2.217}$$

where $W$ is a molecular weight. Specifically for the first reaction, the stoichiometric mass ratio of $C_mH_nN_pO_q$ to $O_2$ is

$$\left.\frac{Y_{fuel}}{Y_{oxy}}\right|_{stoic} = \frac{(12m + n + 14p + 16q)}{32\left(\frac{m-q}{2}\right)}. \tag{2.218}$$

Therefore, the equivalence ratio for the first reaction which is based on carbon monoxide and hydrogen products is given by

$$\Phi_1 = \left(\frac{Y_{fuel}}{Y_{oxy}}\right)\frac{16\left(m-q\right)}{(12m + n + 14p + 16q)}, \tag{2.219}$$

and similarly, the equivalence ratio for the second reaction which is based on carbon dioxide and water products is given by

$$\Phi_2 = \left(\frac{Y_{co} + Y_{h2}}{Y_{oxy}}\right)\frac{16\left(m + \frac{n}{2}\right)}{(28m + n)}. \tag{2.220}$$

If either equivalence ratio is greater than unity, then the mass of oxygen will be completely consumed by its reaction. If either equivalence ratio is less than unity, then the mass of fuel will be completely consumed by its reaction. If either equivalence ratio is unity, then the mass of fuel and oxygen will both be completely consumed by that reaction. Note that $C_mH_nN_pO_q$ is not a fuel in the second reaction because if there is any of this fuel left, all the oxygen was consumed in the first reaction. Therefore, under these conditions the second reaction cannot proceed due to lack of oxygen. Also note that the expression for $\Phi_2$ does not identify which secondary fuel, CO or $H_2$, is limiting.

In order to determine the limiting reactant mass in a multi-fuel (or multi-oxidant) system, a more general approach based on equivalence ratios is required. Consider the reaction

$\zeta_A A + \zeta_B B \rightarrow \zeta_C C + \zeta_D D$ where $\zeta$ are stoichiometric coefficients. The stoichiometric mass ratio of reactant $B$ to $A$ is

$$\left.\frac{Y_B}{Y_A}\right|_{stoic} = \left.\frac{\text{mass}_B}{\text{mass}_A}\right|_{stoic} = \frac{W_B \zeta_B}{W_A \zeta_A}. \tag{2.221}$$

Further, $Y_A$ and $Y_B$ are the mass fractions of $A$ and $B$ in the mixture and

$$\frac{Y_B}{Y_A} = \left.\frac{\text{mass } B}{\text{mass } A}\right|_{mix}. \tag{2.222}$$

The ratio of these quantities is an equivalence ratio; i.e., if

$$\frac{Y_B}{Y_A} > \frac{W_B \zeta_B}{W_A \zeta_A}, \tag{2.223}$$

then $A$ is the limiting reactant, else $B$ is the limiting reactant. However, this inequality can be usefully rearranged. If

$$\frac{Y_A}{W_A \zeta_A} < \frac{Y_B}{W_B \zeta_B}, \tag{2.224}$$

then $A$ is the limiting reactant. The same procedure can be shown to apply to reactions where there are more than two reactants; i.e., if

$$\frac{Y_A}{W_A \zeta_A} < \frac{Y_B}{W_B \zeta_B} < \cdots < \frac{Y_n}{W_n \zeta_n}, \tag{2.225}$$

then $A$ is the limiting reactant of $n$ reactants. Therefore,

$$\text{First Reactant Depleted} = \min_n \left( \frac{Y_n}{W_n \zeta_n} \right). \tag{2.226}$$

Note that the units of $Y_n / W_n \zeta_n$ are $[(\text{mass } n)_{mix}/(\text{mass } n)_{stoic}]/(\text{mass})_{mix}$. Also note that diluents are not reactants and they are not depleted by the reaction. The $\min()$ function should only be applied to fuels and oxygen, not to all species.

To determine the change in mass fraction, $\Delta Y_k^m$, of reactant species $k$ due to reaction $m$, multiply the limiting mass expression by the stoichiometric mass of species $k$:

$$\Delta Y_k^m = -W_k \zeta_k^m \min_n \left( \frac{Y_n}{W_n \zeta_n} \right)^m. \tag{2.227}$$

This expression has units of $[(\text{mass } k)_{stoic}/(\text{mass } n)_{stoic}] \times [(\text{mass } n)_{mix}/(\text{mass})_{mix}]$. Since $n$ is the limiting reactant, the expression within the second set of square brackets is the change in mass fraction of species $n$ due to reaction $m$; this is because the limiting species $n$ is completely used up in the reaction (i.e., the mass fraction of species $n$ goes to zero). The expression within the first set of square brackets modifies the change in mass fraction of species $n$ to yield the change in mass fraction of species $k$ due to reaction $m$. The change in mass fraction of product species $k$ in reaction $m$ is similar but without the minus sign in the above expression.

Since the reactions are given priority, the "products" of Reaction 1 are the "reactants" of Reaction 2. The new mass fractions in the reactant stream for Reaction 2 are given by

$$(Y_k)_{\text{Reaction 2 reactants}} = (Y_k)_{\text{surr}} \pm \Delta Y_k^{\text{Reaction 1}}. \tag{2.228}$$

As noted above, the sign of the second term, $\pm\Delta Y_k^{\text{Reaction 1}}$, is positive for products and negative for reactants. Similarly, the product composition from Reaction 2 is given by

$$(Y_k)_{\text{Reaction 2 products}} = (Y_k)_{\text{Reaction 2 reactants}} \pm \Delta Y_k^{\text{Reaction 2}}. \tag{2.229}$$

Here again the positive sign on the second term is used for products and negative sign is used for reactants. Since the reactions are assumed to occur infinitely fast, the product composition for Reaction 2 is the composition of the reaction zone,

$$(Y_k)_{\text{flame}} = (Y_k)_{\text{Reaction 2 products}}\,. \tag{2.230}$$

### 2.9.6. Conservation Laws

For convenience we restate the Favre-averaged species mass conservation equation, Equation 2.97,

$$\int \frac{\partial \bar{\rho}\tilde{Y}_k}{\partial t}\mathrm{d}V + \int \bar{\rho}\tilde{Y}_k\tilde{u}_j n_j \mathrm{d}S = \int \left(\frac{\mu}{\mathrm{Sc}} + \frac{\mu_t}{\mathrm{Sc}_t}\right)\frac{\partial \tilde{Y}_k}{\partial x_j} n_j \mathrm{d}S + \int \overline{\dot{\omega}_k}\mathrm{d}V, \tag{2.231}$$

where $\bar{\rho}$ is the time averaged density of the mixture, $\tilde{Y}_k$ is the Favre-averaged mass fraction of species $k$, $\tilde{u}_i$ is the Favre-averaged velocity of the mixture, $\mu_t$ is the turbulent eddy viscosity, $\mathrm{Sc}_t$ is the turbulent Schmidt number, and $\overline{\dot{\omega}_k}$ is the time-averaged mass production rate of species $k$ per unit volume of the mixture. This equation is solved on a mesh, one control volume of which is shown in Figure 2.9-1. Within the control volume, the species $k$ mass consumption/production rate, $\dot{m}_{k,consumed/produced} = \overline{\dot{\omega}}_k V_{cell}$, is determined by the EDC model, assuming that the mass transfer process into and out of the reaction zone from the surroundings (cf. Figure 2.9-1) can be represented as a steady process,

$$(\dot{m}_k)_{consumed/produced} = (\dot{m}_k)_{flame} - (\dot{m}_k)_{surr}\,. \tag{2.232}$$

The mixture mass flow rate between the surroundings and the reaction zone is also assumed to be steady,

$$(\dot{m})_{flame} = (\dot{m})_{surr}\,. \tag{2.233}$$

Combining these two expressions yields

$$\begin{aligned}(\dot{m}_k)_{consumed/produced} &= \left[\frac{(\dot{m}_k)_{flame}}{(\dot{m})_{flame}} - \frac{(\dot{m}_k)_{surr}}{(\dot{m})_{surr}}\right](\dot{m})_{flame} \\ &= \left[(Y_k)_{flame} - (Y_k)_{surr}\right](\dot{m})_{flame}\,.\end{aligned} \tag{2.234}$$

It is convenient to normalize this equation with the mass of the control volume, or

$$\frac{(\dot{m}_k)_{consumed/produced}}{M_{cell}} = \left[(Y_k)_{flame} - (Y_k)_{surr}\right]\frac{(\dot{m})_{flame}}{M_{cell}}. \tag{2.235}$$

The term in the brackets is a function of thermochemistry only and is specified by the chemical processes derived in the previous section. The second term, the normalized mass transfer rate, is a function of the turbulent mass exchange rate between the reaction zone and its surroundings. The derivation of this term is the subject of the next subsection.

### 2.9.7. Effect Of Turbulence On Combustion Rates

Magnussen derived the effect of turbulence on combustion rates in terms of high turbulence levels. The derivation here will be for moderate turbulence levels for the flame geometry shown in Figure 2.9-2. The derivation herein does not include proportionality constants. Rather, dimensional reasoning is used to establish the relationship between reaction zone surface area, volume, and mass transfer rates with respect to the prevailing turbulence levels. Constants of proportionality, taken from Magnussen's original derivation, are added at the end.

Characteristic scales are needed for the mass transfer velocity into the reaction zone, the reaction zone surface area, and the reaction zone thickness. The mass transfer velocity into the reaction zone is a velocity appropriate to diffusional length scales that are being modified by the local strain field induced by the turbulent flow,

$$\text{Mass Transfer Velocity} \propto \upsilon. \tag{2.236}$$

An appropriate diffusional velocity is the Kolmogorov velocity, $\upsilon$, which is characteristic of dissipative length scales (i.e., those in which the local strain field is being dissipated by diffusional effects). From Kolmogorov's definition, $\upsilon$ is given by

$$\upsilon \equiv (\nu\epsilon)^{1/4}, \tag{2.237}$$

where $\nu$ is the molecular mixture kinematic viscosity (evaluated at the surrounding temperature), and $\epsilon$ is the rate of kinetic energy dissipation.

The reaction zone is characterized as a continuous flame sheet, highly wrinkled and wrapped around large eddies. The volume of a large eddy is characterized by

$$\text{Volume}_{eddy} \propto L^3, \tag{2.238}$$

where $L$ is the characteristic integral length scale of the turbulence. The reaction zone area is assumed to be proportional to both momentum and scalar influences. While all length scales of the turbulent cascade contribute to wrinkling and stretching the flame, it is assumed that large changes in surface area are associated with large length-scale fluctuations. Therefore, it is assumed that the square of the integral length scale is the most appropriate turbulent length scale for characterizing the reaction zone area.

Species concentrations also affect reaction zone area. Obviously, if no fuel is present, no reaction zone will be present regardless of level of turbulence present. The species influence are denoted by a function, $\chi$, the rationale of which will be described later. Based on these arguments,

$$\text{Area}_{flame} \propto \chi L^2. \tag{2.239}$$

To obtain property values for each zone in Figure 2.9-1, it is necessary to define the volume fractions of the reaction zone and surrounding zones. The reaction zone volume fraction is based on a reaction zone

area and a reaction zone thickness. Since the reaction zone is a strain modified diffusional zone, its thickness is best modeled with a diffusional length scale that is characteristic of the turbulence-induced strain field. Thus the reaction zone thickness is proportional to the Kolmogorov scale, $\eta$,

$$\text{Thickness}_{flame} \propto \eta. \tag{2.240}$$

Kolmogorov's definition of the diffusive length scale is

$$\eta \equiv \left(\frac{\nu^3}{\epsilon}\right)^{1/4}. \tag{2.241}$$

Since this is a characteristic scale analysis, the molecular mixture viscosity is evaluated at the surrounding temperature. The actual reaction zone thickness will be larger due to the volumetric expansion (i.e., lower density) in the reaction zone.

Based on these characteristic scales from the assumed reaction zone geometry in Figure 2.9-2, expressions can be obtained for the mass transfer rate per total mass. The mass exchange rate into the reaction zone per unit eddy mass is given by

$$\frac{\dot{m}_{flame}}{M_{cell}} = \left(\frac{\dot{m}_{flame}}{M_{eddy}}\right)\left(\frac{M_{eddy}}{M_{cell}}\right). \tag{2.242}$$

The first term on the right hand side is given by

$$\frac{\dot{m}_{flame}}{M_{eddy}} = \frac{(\text{SurroundingDensity})\,(\text{FlameArea})\,(\text{MassTransferVelocity})}{(\text{EddyDensity})\,(\text{EddyVolume})}. \tag{2.243}$$

The interpretation of the second term on the right hand side depends upon filtering used (i.e., averaging over scales). For LES, the length scale of the eddy being modeled is proportional to the length scale of the grid. In this case, the size of the eddy and the grid are the same. Therefore, the second term is unity. In RANS modeling, the eddy is much larger than the grid, as is the reaction zone surface being modeled. For RANS, it is assumed that averaged over a sufficient number of eddies, the mass exchange rate into the reaction zone per unit eddy (first term) is uniformly distributed (i.e., independent of length scale) up to the integral length scales. In this case the second term is irrelevant and is assigned a value of unity. For example, for an integral scale eddy with a length scale ten times the grid, the mass transfer into the reaction zone (averaged over many eddies) would be ten times the value for an eddy with a length scale that is just the size of the grid.

Conservation of mass requires that the mass exchange rate into and out of the reaction zone be identical so the properties can be evaluated at the thermodynamic state of either the reactant stream (surroundings) or the product stream (reaction zone). For convenience, they will be defined in terms of the reactant stream temperature and mass fractions. Using the characteristic length and velocity scale arguments given above yields

$$\frac{\dot{m}_{flame}}{M_{cell}} \propto \frac{(\rho_{surr})\,(L^2\chi)\,(\upsilon)}{(\rho_{cell})\,(L^3)} = \chi\frac{\upsilon}{L}\frac{\rho_{surr}}{\rho_{cell}}. \tag{2.244}$$

The standard integral scale estimate [11] of the rate of energy supply to diffusive scale eddies is

$$\epsilon \propto \frac{\text{TurbulentKineticEnergy}}{\text{EddyRollOverTime}} \propto \frac{u'^2}{L/u'} = \frac{u'^3}{L}. \tag{2.245}$$

The turbulence kinetic energy is given as

$$k = \frac{3}{2}u'^2. \tag{2.246}$$

Substituting and rearranging gives

$$L \propto \frac{k^{3/2}}{\epsilon}. \tag{2.247}$$

Ignoring the constant of proportionality and substituting the results into the definition for the Kolmogorov velocity gives

$$\upsilon \propto L\left(\frac{\epsilon}{k}\right)\left(\frac{\nu\epsilon}{k^2}\right)^{1/4}. \tag{2.248}$$

Substituting gives the mass exchange rate into the reaction zone per control volume in terms of standard turbulence parameters,

$$\frac{\dot{m}_{flame}}{M_{cell}} \propto \left(\frac{\nu\epsilon}{k^2}\right)^{1/4}\left(\frac{\epsilon}{k}\right)\chi\frac{\rho_{surr}}{\rho_{cell}}. \tag{2.249}$$

The function $\chi$ is a scalar correction to take into account species effects on the reaction zone area. The function is bounded between (0,1) with 1 representing optimal species concentrations which will maximize the reaction zone area and 0 representing prohibitive species concentrations which would prevent reaction zone formation. Two scalar properties are important, the reactant concentrations and the product concentration (which acts as an ignition source since ignition is not assumed). Therefore, the limiter is written as the product of two terms,

$$\chi = \chi_1\chi_2. \tag{2.250}$$

The function $\chi_1$ is intended to take into account the effect of the reactant mass fractions on the reaction zone surface area. Since the reaction zone surface occurs at stoichiometric concentrations of fuel and oxygen in a diffusion flame, stoichiometric concentrations of reactants in a control volume will result in the largest reaction zone area (controlled by the turbulence levels). In this case, $\chi_1$ is unity. On the other hand, if either fuel or oxygen is zero within a control volume, then $\chi_1$ is zero. Between these extremes, a functional form is assumed which has the correct limiting properties. The function is given by

$$\chi_1 = \frac{1}{\left(\frac{\hat{Y}_{oxy}+\hat{Y}_{prod}}{\hat{Y}_{min}+\hat{Y}_{prod}}\right)\left(\frac{\hat{Y}_{fuel}+\hat{Y}_{prod}}{\hat{Y}_{min}+\hat{Y}_{prod}}\right)}, \tag{2.251}$$

where the normalized mass fractions are defined below.

Overall reaction stoichiometry is determined from the sum of Reactions 1 and 2 in the chemical reaction section (Equations 2.212 and 2.213). The overall reaction is

$$\begin{aligned} \mathrm{C}_m\mathrm{H}_n\mathrm{N}_p\mathrm{O}_q + \left(m + \frac{n-2q}{4}\right)\mathrm{O}_2 \quad &+ \quad \sum(\zeta_d)\,\text{Diluent} \quad \Rightarrow \\ (m)\,\mathrm{CO}_2 \quad &+ \quad \left(\frac{n}{2}\right)\mathrm{H}_2\mathrm{O} + \left(\frac{p}{2}\right)\mathrm{N}_2 + \sum(\zeta_d)\,\text{Diluent} \end{aligned} \tag{2.252}$$

For the overall reaction, the mass ratio of oxygen to the mass ratio of fuel for stoichiometric reaction to $CO_2$ and $H_2O$ is given by

$$S = \left(\frac{Y_{oxy}}{Y_{fuel}}\right)_{Optimal} = \left(m + \frac{n}{4} - \frac{q}{2}\right)\left(\frac{32}{12m + n + 14p + 16q}\right). \tag{2.253}$$

Since mass is conserved in the reaction, $1 + S$ kilograms of product ($CO_2$ and $H_2O$) are produced for every kilogram of fuel consumed for a fuel/oxygen reaction. Note the mass of diluents, such as the nitrogen in the air does not change, as a result of the reaction. It is useful to produce normalized mass fractions based on the masses involved in the stoichiometric reaction.

$$\hat{Y}_{oxy} = \frac{Y_{oxy}}{S} \qquad \hat{Y}_{prod} = \frac{Y_{co2} + Y_{h2o}}{1 + S} \qquad \hat{Y}_{fuel} = \frac{Y_{fuel}}{1} \tag{2.254}$$

Note that the sum of these terms does not equal unity but one minus the mass fraction of diluent in the mixture.

The actual reaction may involve the secondary fuels, so a more general expression is required for the stoichiometric mass ratio of oxygen to fuel (and is used in the Vulcan code).

$$S = \frac{\mathrm{SO2FU} \cdot Y_{fuel} + \mathrm{SO2CO} \cdot Y_{co} + \mathrm{SO2H2} \cdot Y_{h2}}{Y_{fuel} + Y_{co} + Y_{h2}}, \tag{2.255}$$

$$\mathrm{SO2FU} = \left(m + \frac{n}{4} - \frac{q}{2}\right)\left(\frac{32}{12m + n + 14p + 16q}\right), \tag{2.256}$$

$$\mathrm{SO2CO} = \frac{32}{2 \cdot 28}, \tag{2.257}$$

$$\mathrm{SO2H2} = \frac{32}{2 \cdot 2}. \tag{2.258}$$

The product mass fractions are adjusted for the mass of nitrogen that accompanies the oxygen in air – the nitrogen is treated as a product species. The normalized mass fractions are

$$\hat{Y}_{oxy} = \frac{Y_{oxy}}{S} \qquad \hat{Y}_{prod} = \frac{3.39Y_{co2} + 3.92Y_{h2o}}{1 + 4.29S} \qquad \hat{Y}_{fuel} = \frac{Y_{fuel}}{1} \tag{2.259}$$

where

$$Y_{prod}|_{co2} = Y_{co2}\left(1 + 3.76\frac{MW_{n2}}{MW_{co2}}\right), Y_{prod}|_{h2o} = Y_{h2o}\left(1 + 1.88\frac{MW_{n2}}{MW_{h2o}}\right), \tag{2.260}$$

The molar ratio of nitrogen to oxygen in air is 3.76 and the mass ratio is 3.29. The production mass fraction, $Y_{prod}$, can be computed directly from the $CO_2$ and $H_2O$ mass fractions as long as the only source of product species in the flow field comes directly from combustion. If there is injection of product species into the domain from a diluent stream or from an ambient concentration, then a transport equation should be solved for the product mass fraction (see Section 2.9.12).

Since combustion always occurs at a stoichiometric surface in a diffusion flame, there is a limiting reactant mass fraction in a fuel/oxygen mixture within the control volume unless the ratio is stoichiometric. The limiting reactant mass fraction is given by

$$\hat{Y}_{min} = \min\left(\hat{Y}_{fuel}, \hat{Y}_{oxy}\right). \tag{2.261}$$

The function $\chi_1$ can be seen to approach the correct limits most clearly if the mass fraction of products, $Y_{prod}$, is set to zero. If the mixture is fuel lean, $\hat{Y}_{min} = \hat{Y}_{fuel}$ and $\chi_1$ is equal to the fuel to oxygen ratio which decreases to zero as the fuel mass fraction is decreased. If the mixture is fuel rich, $\hat{Y}_{min} = \hat{Y}_{oxy}$ and $\chi_1$ is equal to the oxygen to fuel ratio which decreases to zero as the fuel mass fraction is increased. At stoichiometric, $\chi_1$ is unity.

The function $\chi_2$ is intended to take into account the existence of reaction zone surface as a precondition for reaction zone surface propagation. A stoichiometric surface without reaction can exist in a flow field if there is no ignition source. An external source is required for ignition. However, once ignited, reaction zone propagation can be interpreted as new flame surface being ignited by existing adjacent reaction zone surface. A good indicator of existing flame surface is the presence of hot combustion products within the control volume and this fact is used to create the function $\chi_2$.

The value of $\chi_2$ is zero if no combustion products are present. If the product mass fraction was uniformly distributed, then the probability of ignition would increase with the ratio of product mass fraction to reactant mass fraction. However, the combustion products are not uniformly distributed but concentrated around the reaction zones, thereby increasing the probability of propagation of reaction zone surface for a given product mass fraction. The assumed functional form of $\chi_2$ that has these characteristics is

$$\begin{aligned} \chi_2 &= \frac{\left(\dfrac{\mathrm{ExistingProductMassFraction}}{\mathrm{MaxFlameVolume}}\right)}{\left(\dfrac{\mathrm{MaxPossibleProductMassFraction}}{\mathrm{CharacteristicProductVolume}}\right)} \\ &= \left(\frac{\mathrm{CharacteristicProductVolume}}{\mathrm{MaxFlameVolume}}\right)\left(\frac{\mathrm{ExistingProductMassFraction}}{\mathrm{MaxPossibleProductMassFraction}}\right). \end{aligned} \tag{2.262}$$

The maximum volume of the reaction zone is the thickness times its area,

$$\gamma = \frac{(Area \cdot Thickness)_{flame}}{Volume_{eddy}} \propto \frac{L^2\eta}{L^3} = \frac{\eta}{L}. \tag{2.263}$$

Using the definition for the Kolmogorov length scale and substituting the turbulence kinetic energy for length scale, $L$, gives

$$\gamma \propto \left(\frac{\nu\epsilon}{k^2}\right)^{3/4}, \tag{2.264}$$

which is the maximum reaction zone volume per eddy volume. The value of $\gamma^{1/3}$ is bounded by one since the length scale ratio of the flame volume to eddy volume cannot be larger than one.

The characteristic product volume can be defined by assuming the majority of combustion products are held up within a distance corresponding to the Taylor microscale from the reaction zone surface,

$$\gamma_\lambda = \frac{(Area \cdot Thickness)_{prod}}{Volume_{eddy}} \propto \frac{L^2\lambda}{L^3} = \frac{\lambda}{L}. \tag{2.265}$$

Note that this assumption is used only to establish an ignition probability. For actual property evaluation, it is assumed that the combustion products are well mixed with the surroundings. Taking

the ratio of the volumes gives,

$$\frac{\gamma_\lambda}{\gamma} \propto \frac{\lambda}{\eta}. \tag{2.266}$$

Using the standard definition of this ratio (Tennekes and Lumley [11]) gives,

$$\frac{\lambda}{\eta} = \mathrm{Re}_L^{1/4}. \tag{2.267}$$

The Reynolds number can be defined in terms of turbulent kinetic energy and its dissipation by,

$$\mathrm{Re}_L = \frac{k^2}{\nu\epsilon}. \tag{2.268}$$

Substituting gives,

$$\left(\frac{\mathrm{CharacteristicProductVolume}}{\mathrm{MaximumFlameVolume}}\right) = \left(\frac{1}{\gamma^{1/3}}\right). \tag{2.269}$$

The existing product mass fraction is given by $Y_{prod}$. The maximum possible product mass fraction is the sum of the existing products and the products that could be formed if all available reactants were to burn. Since combustion takes place at a stoichiometric surface, the limiting reactant mass fraction is given by $Y_{min}$. Therefore, $\chi_2$ becomes

$$\chi_2 = \left(\frac{1}{\gamma^{1/3}}\right)\left(\frac{\hat{Y}_{prod}}{\hat{Y}_{prod} + \hat{Y}_{min}}\right). \tag{2.270}$$

Functionally, $\chi_2$ can exceed unity but the product $\chi_1\chi_2$ is limited to the range (0,1). The function $\chi$ is now completely described in terms of species and turbulence properties.

Combining all previous results gives the following result for species consumption/production,

$$\frac{(\dot{m}_k)_{consumed/produced}}{M_{cell}} \propto \left[(Y_k)_{flame} - (Y_k)_{surr}\right]\left(\left(\frac{\nu\epsilon}{k^2}\right)^{1/4}\left(\frac{\epsilon}{k}\right)\chi\left(\frac{\rho_{surr}}{\rho_{cell}}\right)\right), \tag{2.271}$$

where $\chi$ is defined above in terms of $\chi_1$, $\chi_2$.

The above derivation is intended to provide a physical interpretation to Magnussen's EDC model for large fires typified by medium turbulence levels with diffusive combustion. Proportionality constants are needed to close the model. As always, constants can be tweaked for a given flow to produce the best result for that flow. However, we will use the constants as derived for more general flows (Ertesvåg and Magnussen [49]). With these constants, the model equations match those from the KAMELEON-II-FIRE code (Holen et al. [50]).

Using these constants, the maximum reaction zone volume fraction is given by

$$\gamma = 9.7\left(\frac{\nu\epsilon}{k^2}\right)^{3/4}. \tag{2.272}$$

Taking into account species limitations, the flame volume fraction is given by

$$\gamma\chi = 9.7\left(\frac{\nu\epsilon}{k^2}\right)^{3/4}\chi. \tag{2.273}$$

The reaction rate of fuel is given by

$$\frac{(\dot{m}_k)_{consumed/produced}}{M_{cell}} \propto \left[(Y_k)_{flame} - (Y_k)_{surr}\right] \left(23.6 \left(\frac{\nu\epsilon}{k^2}\right)^{1/4} \left(\frac{\epsilon}{k}\right) \chi \left(\frac{\rho_{surr}}{\rho_{cell}}\right) \chi_3\right). \quad (2.274)$$

The additional scalar function, $\chi_3$ , at the end of Equation 2.274 is multiplier on the combustion rate that Magnussen found necessary to maintain the mass transfer rate when the product concentration is high in premixed flames. Its necessity suggests that perhaps alternate scalings should be examined, but for consistency with the published model, it is implemented here as

$$\chi_3 = \min\left[\frac{\hat{Y}_{prod} + \hat{Y}_{min}}{\hat{Y}_{min}}, \frac{1}{\gamma^{1/3}}\right]. \quad (2.275)$$

### 2.9.8. Average Control Volume Properties

The volume and mass exchange process between the two zones is assumed to be constant over a time step. Consequently, cell averaged properties for the mean flow equations are a volume weighted sum of the properties in the two zones. Therefore, all control volume properties are given by

$$\phi_{cell} = \frac{\phi_{flame} V_{flame} + \phi_{surr} V_{surr}}{V_{cell}}. \quad (2.276)$$

The maximum volume fraction of the reaction zone, $\gamma$, was determined previously from momentum considerations. The actual volume fraction is the maximum volume fraction times the scalar function, $\chi$. The surroundings is the volume fraction that remains after the reaction zone volume has been removed. Therefore,

$$\phi_{cell} = \phi_{flame}(\gamma\chi) + \phi_{surr}(1 - \gamma\chi). \quad (2.277)$$

Volume averaged properties given by Equation 2.277 are desired. However, the estimates used to obtain $\gamma\chi$ (i.e., Equation 2.272) are based on uniform cell temperatures. Clearly, the flame zone will be hotter than the surroundings, so the volume fraction occupied by the flame will be larger than given by Equation 2.277 (and the surroundings fraction smaller).

A first order non-isothermal estimate is made to account for flame volume fraction. This estimate assumes that the non-homogeneous density field does not affect the local turbulence field (or alternately, that dilatation cancels the baroclinic generation) such that isothermal, isotropic, homogeneous turbulence estimates for the turbulent kinetic energy, $k$, and its dissipation, $\epsilon$, hold. (This assumption is made in virtually all models by the necessity that the fundamental research to quantify the actual coupling has not been done.)

A mass balance then gives a first order estimate for the actual flame volume fraction at the flame temperature. The actual flame volume at the flame temperature is given by its isothermal estimate times the cell mean density (used to obtain the isothermal estimate) divided by the actual flame density.

Thus, Equation 2.277 becomes,

$$\phi_{cell} = \phi_{flame}\left(\gamma\chi\right)\frac{\rho_{cell}}{\rho_{flame}} + \phi_{surr}\left(1-\gamma\chi\right)\frac{\rho_{cell}}{\rho_{surr}}. \tag{2.278}$$

Where the mean density is given by

$$\rho_{cell} = \left[\frac{\left(\gamma\chi\right)}{\rho_{flame}} + \frac{\left(1-\gamma\chi\right)}{\rho_{surr}}\right]^{-1}. \tag{2.279}$$

An interpretation of Equation 2.278 and Equation 2.279 is that $\gamma\chi$ is, therefore, not a volume fraction estimate but a mass fraction estimate. However, Reynolds, not Favre averaged properties are desired for source term closure estimates. In this case, Equation 2.279, is intended as a non-isothermal volume estimate, which the mass weighted isothermal volume estimate happens to be the best available estimator until turbulence coupling in reacting flows can be elucidated. All cell averaged properties are given by Equation 2.279. Equation 2.277 is intended for clarification only.

## 2.9.9. Limits Testing

Parameters in the EDC model take on limiting values in the presence of piloting conditions and extinction conditions. The limits are discussing in the following subsections.

### *2.9.9.1. Ignition Criteria*

Ignition will not occur in the above mechanism unless products are formed. An external ignition source (or pilot flame) is simulated by setting $\chi$ to be greater than zero (the product mass fraction is set to 0.2 times the maximum products that could be formed by the existing fuel in the current implementation) in a cell with fuel and oxygen present. This can be done on a cell by cell basis to represent point ignition sources, or in the whole domain if global ignition is required. If a pilot flame is to be simulated, the cells associated with it have $\chi$ set to be greater than zero for the duration of the calculation. If a transient ignition (e.g., spark) is to be simulated, the cells initially have $\chi$ set to be greater than zero. However, after a minimum temperature is reached within a cell, $T_{ign}$ (K) (a user input), $\chi$ is no longer specified but calculated from the species concentrations and turbulence levels as derived previously.

### *2.9.9.2. Extinction Criteria*

Extinction occurs when $\chi = 0$. This occurs automatically when the fuel and/or air is consumed. Local extinction can also be caused within a cell due to high turbulence levels. At high turbulence levels, the reaction zone can be appropriately modeled as a perfectly stirred reactor (PSR). A PSR blows out when

the residence time is less than a minimum value for a given composition. The residence time, $\tau_{res}$, in the reaction zone volume is given by

$$\tau_{res} = \frac{\text{Volume}_{flame}}{\text{VolumeFlowRate}_{flame}} = \frac{\left(\frac{\text{Volume}_{flame}}{\text{Volume}_{cell}}\right)}{\frac{\left(\frac{\dot{m}_{flame}}{\rho_{surr}}\right)}{\left(\frac{M_{cell}}{\rho_{cell}}\right)}}. \tag{2.280}$$

Simplifying gives

$$\tau_{res} = \frac{\chi\gamma}{\left(\frac{\dot{m}_{flame}}{M_{cell}}\right)\frac{\rho_{cell}}{\rho_{surr}}}. \tag{2.281}$$

Substituting prior relations gives

$$\tau_{res} \propto \frac{\chi\left(\frac{\nu\epsilon}{k^2}\right)^{3/4}}{\chi\left(\frac{\nu\epsilon}{k^2}\right)^{1/4}\left(\frac{\epsilon}{k}\right)}. \tag{2.282}$$

Simplifying and substituting Magnussen's constant of proportionality gives

$$\tau_{res} = \frac{1}{2.43}\left(\frac{\nu}{\epsilon}\right)^{1/2}. \tag{2.283}$$

Comparison of the calculated residence time with a user input minimum residence time (based on precalculation using a PSR and appropriate chemistry) determines whether or not combustion is allowed to continue. If so, heat release is calculated as derived herein, and finite-rate effects are not considered. However, if the calculated residence time is below the minimum value, $\chi$ is set equal to zero which causes combustion to cease within a cell.

#### 2.9.9.3. Laminar Values

As currently formulated, the model assumes the flow is fully turbulent and does not model laminar combustion. Minimum values for the reaction zone volume, $\gamma$, and mass transport into the reaction zone per mass in the cell, $\dot{m}_{flame}/M_{cell}$, are required in conditions with low turbulence levels to prevent singularities.

#### 2.9.9.4. Scalar Limits

The mass fractions of fuel, air, and products must remain bounded (0,1). This requires that the consumption rate for the species with the limiting concentration times the time step must be less than or equal to the mass of species.

## 2.9.10. Cell Value Information Used By Model

The combustion model requires inputs from the transport equations for cell averaged variables at the start of a time step. These variables include pressure, $P_{th}$ (dynes/cm$^2$), species mass fractions $Y_i$, density, $\rho_{cell}$ (g/cm$^3$), mixture molecular weight, $W_{mix}$, (g/mole), turbulent kinetic energy, $k$ (cm$^2$/sec$^2$), dissipation of the turbulent kinetic energy, $\epsilon$ (cm$^2$/sec$^3$), mixture kinematic viscosity, $\nu$ (cm$^2$/sec), individual (i.e., chemical plus sensible) enthalpies, $h_i$ (ergs/g), and mixture enthalpy, $h_{cell}$ (ergs/g).

## 2.9.11. Model Outputs

The two outputs of the combustion model are the species consumption rates and property estimates.

### *2.9.11.1. Species Consumption Estimates*

Noting the general relation between cell averaged values and surrounding values, Equation 2.277, the surrounding and cell mass fractions can be related to give

$$\left[(Y_k)_{flame} - (Y_k)_{surr}\right] = \frac{\left[(Y_k)_{flame} - (Y_k)_{cell}\right]}{(1-\gamma\chi)}. \tag{2.284}$$

Substituting this result and the definition of $\tau_{res}$ into the species consumption/production rate gives the source term in the species transport equation, Equation 2.231,

$$\overline{\dot{\omega}}_k = \frac{\left[(Y_k)_{flame} - (Y_k)_{cell}\right]}{\tau_{res}} \left(\frac{\gamma\chi}{1-\gamma\chi}\right) \chi_3, \tag{2.285}$$

for the species mass production/consumption rate in a control volume. The subscript $k$ is understood to be for each species, $C_mH_nN_pO_q$, $O_2$, $N_2$, CO, $H_2$, $CO_2$, $H_2O$, and any diluents in the system.

### *2.9.11.2. Property Estimates*

It is important in turbulent processes that nonlinear fluctuating quantities be appropriately represented. Properties for which nonlinear fluctuations are important include the radiative emissive power (proportional to the fourth power of temperature) and density.

To get the radiative emissive power, it is first necessary to get the temperature within each zone. This is accomplished by iterative estimate based on the species mass fractions within each zone. Since total (chemical plus sensible) enthalpy is used for each species, the total enthalpy per unit mass in the control volume does not change between the reaction zone or the surrounding zone. The partitioning of chemical and sensible enthalpy is different for the reaction and surrounding zone, but the specific total enthalpy is equal to the cell value defined at the beginning of each time step. (Note: this is not a statement of the energy equation, it is only a statement of property values within each zone and the cell. Obviously, the enthalpy does vary after radiation transport is allowed to occur and species are allowed to

advect between cells at the end of the time step as governed by the energy equation). The reaction zone temperature, $T_{flame}$, is obtained from iterative solution of

$$h_{flame} = \sum Y_k h_k \left(T\right)\Big|_{flame}, \tag{2.286}$$

and the surrounding temperature, $T_{surr}$, is obtained from iterative solution of

$$h_{surr} = \sum Y_k h_k \left(T\right)\Big|_{surr}. \tag{2.287}$$

The average emissive power is given by

$$\sigma\overline{\alpha T^4}_{rad} = \sigma\bar{\alpha}\left(T^4_{flame}\left(\gamma\chi\right)\frac{\bar{\rho}}{\rho_{flame}} + T^4_{surr}\left(1-\gamma\chi\right)\frac{\bar{\rho}}{\rho_{surr}}\right) \qquad (\mathrm{ergs/cm^2{-}s}). \tag{2.288}$$

An important assumption implied by the form of Equation 2.288 is that the turbulent fluctuations between the temperature and absorption coefficient are weakly correlated [8]. ( Note that the intent of the averaging form above is to volume-weight the emissive power from the flame and surrounding zones. This form implies that $\gamma\chi$ should be viewed as a mass fraction rather than a volume fraction as discussed for Equation 2.276. )

The density of each zone can be calculated according to the perfect gas law. For the reaction zone volume, the density is

$$\rho_{flame} = \frac{P_{th}W_{flame}}{RT_{flame}}, \tag{2.289}$$

where $R$ is the universal gas constant and $P_{th}$ is the constant thermodynamic pressure. For the surroundings, the density is

$$\rho_{surr} = \frac{P_{th}W_{surr}}{RT_{surr}}. \tag{2.290}$$

The soot model uses the temperatures, densities, and mass fractions of reaction zone and surroundings according to the above estimates.

## 2.9.12. Combustion Products Transport Equation

The product mass fraction represents the products formed by combustion ($CO_2$ and $H_2O$ for hydrocarbon fuels, and $H_2O$ for hydrogen fuel). If any of the product species are injected into the domain through either an initial condition or boundary condition to simulate a diluent stream or ambient concentration, their influence must be removed in order for the $\chi_2$ reaction limiter to function properly. A transport equation similar to Equation 2.231 is used where the reaction rate is given by

$$\dot{\omega}_{prod} = 3.392\dot{\omega}_{co2} + 3.924\dot{\omega}_{h2o}. \tag{2.291}$$

This transported product mass fraction can only be formed due to reaction within the domain and cannot be injected through either initial or boundary conditions. Therefore, the only boundary conditions that are required are at an outflow so that products may exit the domain, and a zero value at

any surfaces where a species Dirichlet condition is applied. All of these cases are handled automatically so that nothing needs to be specified by the user.

Note that a pilot stream will be unable to ignite a flame when using this model. It will be treated as an inert diluent stream, so that the normal ignition model will be required to ignite the flame. This model in its current form should not be used for piloted flames.

Also note that if the only source of products in the simulation is combustion, then the product mass fraction can be computed directly from the local species mass fractions and solving this transport equation is unnecessary.

## 2.9.13. Chemical Equilibrium Models

The EDC combustion model uses a two-step chemical reaction, where the fuel species is consumed by the reaction in Equation 2.212 to form CO and $H_2$, and then these intermediate species are consumed by the reaction in Equation 2.213 to form $CO_2$ and $H_2O$. If oxygen is present in excess, then none of the intermediate species will remain and only $CO_2$ and $H_2O$ will be produced. In reality, these reactions would not proceed to completion, but instead would reach an equilibrium where some of the intermediate species can persist. This can lead to a significantly different mixture composition and even a different mixture temperature than what the standard EDC model would predict, especially at higher temperatures.

Fuego includes two optional models that can include the effects of two independent chemical equilibrium reactions into the standard EDC model, to better predict high-temperature combustion species and temperatures.

### 2.9.13.1. $CO_2$ *Dissociation Model*

At high temperatures, the equilibrium reaction

$$CO_2 \Leftrightarrow CO + \frac{1}{2}O_2 \tag{2.292}$$

becomes active to dissociate $CO_2$ species back into CO and $O_2$, which has the effect to cool the gas mixture. Including the effects of this dissociation reaction will help to control nonphysically-high temperatures that might result otherwise.

This model will adjust the EDC-reacted mixture $(Y_k)_{\text{flame}}$ in Equation 2.285 to include the effects of equilibrium reaction 2.292. This equilibrium can be modeled by

$$K_p = \exp\left(\frac{-\Delta G_T^o}{R_u T}\right), \tag{2.293}$$

where $R_u$ is the ideal gas constant, $T$ is the temperature at which the equilibrium is being calculated, $K_p$ is the equilibrium constant for this dissociation reaction, and $\Delta G_T^o$ is the standard-state Gibbs function

change for this reaction. The equilibrium constant $K_p$ for Equation 2.292 is defined as

$$K_p = \frac{\left(\frac{P_{\mathrm{CO}}}{P^o}\right)\left(\frac{P_{\mathrm{O_2}}}{P^o}\right)^{\frac{1}{2}}}{\left(\frac{P_{\mathrm{CO_2}}}{P^o}\right)}, \tag{2.294}$$

where $P_{\mathrm{CO}}$, $P_{\mathrm{O_2}}$, and $P_{\mathrm{CO_2}}$ are the partial pressures of CO, $O_2$, and $CO_2$, respectively, and $P^o$ is the reference pressure taken as $1\ atm$. The standard-state Gibbs function change for this reaction can be evaluated in terms of the Gibbs function of formation for each species at temperature $T$,

$$\Delta G^o_T = \left(\bar{g}^o_{f,\mathrm{CO}} + \frac{1}{2}\bar{g}^o_{f,\mathrm{O_2}} - \bar{g}^o_{f,\mathrm{CO_2}}\right)_{T_{\mathrm{ref}}=T}. \tag{2.295}$$

The partial pressure of species $k$ can be computed by $P_k = X_k P$, where $P$ is the static pressure of the mixture and $X_k$ is the mole fraction of species $k$, defined as $X_k = n_k/n_{\mathrm{tot}}$ with the total number of moles of all species being defined as $n_{\mathrm{tot}} = \sum_i n_i$. After making these substitutions and simplifying, the equilibrium equation that needs to be solved, written in terms of moles of each species in a fixed-mass volume, is

$$\frac{n_{\mathrm{CO}}\, n_{\mathrm{O_2}}^{1/2}}{n_{\mathrm{CO_2}}\, n_{\mathrm{tot}}^{1/2}}\left(\frac{P}{P^o}\right)^{\frac{1}{2}} = \exp\left(\frac{-\Delta G^o_T}{R_u T}\right). \tag{2.296}$$

Additional equations may be written to enforce conservation of C and O atoms within the reaction volume,

$$N_C = n_{\mathrm{CO}} + n_{\mathrm{CO_2}} \tag{2.297}$$

$$N_O = 2\, n_{\mathrm{CO_2}} + n_{\mathrm{CO}} + 2\, n_{\mathrm{O_2}}, \tag{2.298}$$

where $N_C$ and $N_O$ are the fixed number of moles of carbon and oxygen atoms, respectively, during the equilibrium reaction. Equations 2.296, 2.297, and 2.298 represent a system of three equations that can be solved for the three unknowns $n_{\mathrm{CO_2}}$, $n_{\mathrm{CO}}$, and $n_{\mathrm{O_2}}$ at the equilibrium state.

The numerical solution procedure begins by approximating the number of moles of each species from the reacted mixture mass fraction vector $Y_i$ as $n_i = Y_i/W_i$, on a per-unit-mass-of-mixture basis. Eliminating $n_{\mathrm{CO_2}}$ and $n_{\mathrm{O_2}}$ from Equation 2.296 yields a nonlinear equation that can be solved directly for $n_{\mathrm{CO}}$ from the fixed atom balances at a fixed temperature $T$ and pressure $P$,

$$n_{\mathrm{CO}}^2\,(N_O - 2N_C + n_{\mathrm{CO}}) - \left(\frac{P^o}{P}\right)\exp\left(\frac{-2\Delta G^o_T}{R_u T}\right)(N_C - n_{\mathrm{CO}})^2\left(N_O + n_{\mathrm{CO}} + 2\sum_j^{N_{\mathrm{inert}}} n_j\right) = 0, \tag{2.299}$$

where $\sum_j^{N_{\mathrm{inert}}} n_j$ represents the summation of the number of moles of all species present in the mixture that do not participate directly in the equilibrium reaction, *i.e.* all species except for $CO_2$, CO, and $O_2$.

A standard Newton's method may be used to iteratively solve 2.299,

$$n_{\mathrm{CO}}^{n+1} = n_{\mathrm{CO}}^n - \frac{f(n_{\mathrm{CO}})}{f'(n_{\mathrm{CO}})}, \tag{2.300}$$

where the function $f(n_{\mathrm{CO}})$ is Equation 2.299 and the derivative function $f'(n_{\mathrm{CO}})$ is

$$f'(n_{\mathrm{CO}}) = 2\,n_{\mathrm{CO}}\left(N_O - 2N_C + n_{\mathrm{CO}}\right) + n_{\mathrm{CO}}^2 - \left(\frac{P^o}{P}\right)\exp\left(\frac{-2\Delta G_T^o}{R_u T}\right)\left[\left(N_C - n_{\mathrm{CO}}\right)^2 - 2\left(N_C - n_{\mathrm{CO}}\right)\left(N_O + n_{\mathrm{CO}} + 2\sum_j^{N_{\mathrm{inert}}} n_j\right)\right]. \tag{2.301}$$

Once this equation is solved for $n_{\mathrm{CO}}$, then the following equations may be used to evaluate the remaining equilibrium species moles,

$$n_{\mathrm{CO_2}} = N_C - n_{\mathrm{CO}} \tag{2.302}$$

$$n_{\mathrm{O_2}} = \frac{1}{2}\left(N_O - 2N_C + n_{\mathrm{CO}}\right). \tag{2.303}$$

With the new molar mixture defined for the equilibrium species, the mass fraction vector may be reconstructed by $Y_i = n_i W_i$. This new mixture composition will result in a different temperature since the enthalpy is fixed. After the new temperature is evaluated, this entire procedure may be repeated iteratively until the mixture temperature converges to within a specified tolerance.

### 2.9.13.2. $H_2$ Dissociation Model

Similar to the $CO_2$ dissociation model described in 2.9.13.1, at high temperatures the equilibrium reaction

$$\mathrm{H_2} \Leftrightarrow 2\,\mathrm{H} \tag{2.304}$$

becomes active to dissociate $H_2$ species into H atoms, which has the effect to cool the gas mixture. Including the effects of this dissociation reaction in addition to the $CO_2$ dissociation reaction will help to control nonphysically-high temperatures that might result otherwise.

This model will adjust the EDC-reacted mixture $(Y_k)_{\mathrm{flame}}$ in Equation 2.285 to include the effects of equilibrium reaction 2.304. This equilibrium can be modeled by Equation 2.293, with the equilibrium constant defined as

$$K_p = \frac{\left(\frac{P_{\mathrm{H}}}{P^o}\right)^2}{\left(\frac{P_{\mathrm{H_2}}}{P^o}\right)}, \tag{2.305}$$

where $P_{\mathrm{H}}$ and $P_{\mathrm{H_2}}$ are the partial pressures of H and $H_2$, respectively. The standard-state Gibbs function change for this reaction can be evaluated in terms of the Gibbs function of formation for each species at temperature $T$,

$$\Delta G_T^o = \left(2\,\bar{g}_{f,\mathrm{H}}^o - \bar{g}_{f,\mathrm{H_2}}^o\right)_{T_{\mathrm{ref}}=T}. \tag{2.306}$$

Simplifying this equilibrium expression and writing it in terms of the number of moles of each species in a fixed-mass volume results in the equilibrium equation

$$\frac{n_{\mathrm{H}}^2}{n_{\mathrm{H_2}}\,n_{\mathrm{tot}}}\left(\frac{P}{P^o}\right) = \exp\left(\frac{-\Delta G_T^o}{R_u T}\right). \tag{2.307}$$

An additional equation may be written to enforce conservation of H atoms within the reaction volume,

$$N_H = n_{\mathrm{H}} + 2\, n_{\mathrm{H_2}}, \tag{2.308}$$

where $N_H$ is the fixed number of moles of hydrogen atoms during the equilibrium reaction. Equations 2.307 and 2.308 represent a system of two equations that can be solved for the two unknowns $n_{\mathrm{H_2}}$ and $n_{\mathrm{H}}$ at the equilibrium state.

Similar to the $CO_2$ dissociation model, the numerical solution procedure begins by approximating the number of moles of each species from the reacted mixture mass fraction vector $Y_i$ as $n_i = Y_i/W_i$, on a per-unit-mass-of-mixture basis. Eliminating $n_{\mathrm{H_2}}$ from Equation 2.307 yields a nonlinear equation that can be solved directly for $n_{\mathrm{H}}$ from the fixed atom balance at a fixed temperature $T$ and pressure $P$,

$$n_{\mathrm{H}}^2 - \frac{1}{4}\left(\frac{P^o}{P}\right)\exp\left(\frac{-\Delta G_T^o}{R_u T}\right)(N_H - n_{\mathrm{h}})\left(N_H + n_{\mathrm{H}} + 2\sum_j^{N_{\mathrm{inert}}} n_j\right) = 0, \tag{2.309}$$

where $\sum_j^{N_{\mathrm{inert}}} n_j$ represents the summation of the number of moles of all species present in the mixture that do not participate directly in the equilibrium reaction, *i.e.* all species except for $H_2$ and H.

A standard Newton's method may be used to iteratively solve 2.309,

$$n_{\mathrm{H}}^{n+1} = n_{\mathrm{H}}^n - \frac{f(n_{\mathrm{H}})}{f'(n_{\mathrm{H}})}, \tag{2.310}$$

where the function $f(n_{\mathrm{H}})$ is Equation 2.309 and the derivative function $f'(n_{\mathrm{H}})$ is

$$f'(n_{\mathrm{H}}) = 2\, n_{\mathrm{H}} - \frac{1}{4}\left(\frac{P^o}{P}\right)\exp\left(\frac{-\Delta G_T^o}{R_u T}\right)\left[(N_H - n_{\mathrm{H}}) - \left(N_H + n_{\mathrm{H}} + 2\sum_j^{N_{\mathrm{inert}}} n_j\right)\right]. \tag{2.311}$$

Once this equation is solved for $n_{\mathrm{H}}$, then the following equations may be used to evaluate the remaining equilibrium species moles,

$$n_{\mathrm{H_2}} = \frac{1}{2}(N_H - n_{\mathrm{H}}) \tag{2.312}$$

With the new molar mixture defined for the equilibrium species, the mass fraction vector may be reconstructed by $Y_i = n_i W_i$. This new mixture composition will result in a different temperature since the enthalpy is fixed. After the new temperature is evaluated, this entire procedure may be repeated iteratively until the mixture temperature converges to within a specified tolerance.

## 2.10. LAMINAR FLAMELET TURBULENT COMBUSTION MODEL

Laminar flamelet models for non-premixed turbulent combustion treat turbulent flames as an ensemble of laminar diffusion flames. [51] Nonequilibrium chemistry effects may be included in the model by

accounting for localized fluid strain, resulting in what is classically called the Strained Laminar flamelet Model (SLFM). Nonadiabatic effects may also be included by accounting for losses to the surroundings in the ensemble of flamelets.

The fundamental assumption is that the chemical time scales of the important reactions are fast enough that they occur only in a thin layer around stoichiometry, thinner (ideally) than the smallest scales of the turbulence. Defining a small quantity $\epsilon = \ell_{\text{reaction zone}}/\ell_{\text{mixing layer}} \ll 1$, we can examine the governing equations in that thin region using a multiscale asymptotic expansion as

$$Y_i = Y_i(\zeta, \tau, x, t) + \epsilon Y_i^1(\zeta, \tau, x, t) + \ldots, \qquad \zeta = \frac{Z(x,t) - Z_{\text{st}}}{\epsilon} \text{ and } \tau = t/\epsilon^2. \tag{2.313}$$

Collecting the dominant terms, making some standard simplifications, and assuming that the chemical reaction scales as $\epsilon^{-2}$, the state of the gas depends on the flow scale $Z$ and $\chi = 2D|\nabla Z|^2$:

$$\begin{aligned}
&\rho\frac{\partial Y_i}{\partial t} - \frac{\rho\chi}{2}\frac{1}{\text{Le}_i}\frac{\partial^2 Y_i}{\partial Z^2} - \dot{\omega}_i(\vec{\Phi}) = 0 \\
&\rho\frac{\partial T}{\partial t} - \frac{\rho\chi}{2}\left(\frac{\partial^2 T}{\partial Z^2} + \frac{1}{c_p}\frac{\partial c_p}{\partial Z}\frac{\partial T}{\partial Z}\right) - \dot{\omega}_T(\vec{\Phi}) = 0 \\
&\text{and } T(Z=0,t) = T_{\text{ox.}}, T(Z=1,t) = T_{\text{fuel}}, Y_i(Z=0,t) = Y_{\text{i,ox.}}, Y_i(Z=1,t) = Y_{\text{i,fuel}}, \\
&\text{with } \rho = \rho(\vec{\Phi}) \text{ and } c_p = c_p(\vec{\Phi}),
\end{aligned} \tag{2.314}$$

where $\vec{\Phi}$ is the state vector $\vec{\Phi} = (P_{\text{th}}, T, Y_0, Y_1, \ldots, Y_N)$. The approximation allows us to resolve the chemical scales in the phase space of the mixture fraction instead of on a three-dimensional grid, granting dramatic computational savings. If we make the additional assumption that the chemistry is quasi-steady on the scale of the flow, then the chemical structure in mixture fraction space can be pre-computed offline from the simulation for a range of flow parameters $\chi$ and tabulated (using `fuego_tabular_props`). During the flow simulation, the solution of the flamelet simulation can be queried to determine required flow properties, e.g. $\rho = \rho(Z, \chi)$. Note that the flamelet formulation in Eq. 2.314 is specifically for a "two-stream" problem, with constant Lewis numbers, where the boundary and initial conditions of the simulation can be completely described by linear combinations of two constant state vectors. Additional "streams" and boundary heat losses will require additional transport equations to be solved.

This section summarizes the basic formulation and implementation details of both the adiabatic and nonadiabatic flamelet model and SLF model, including both the property table generation procedure in `fuego_tabular_props` and the usage of the property table in `fuego` to evaluate turbulent filtered quantities of interest for both adiabatic and nonadiabatic configurations.

## 2.10.1. Adiabatic Property Table Generation

### *2.10.1.1. Laminar Flamelet Generation*

Unstrained flamelet libraries, where nonequilibrium chemistry effects may be neglected with respect to fluid strain rates, can be generated directly with the `fuego_tabular_props` application. These

libraries should be used either in laminar flow or in turbulent flow where the turbulence/chemistry interactions may be neglected.

Equilibrium chemistry, Burke-Schumann chemistry, or nonreacting flow scenarios are supported in configurations where there are two or more streams that may be mixed and potentially reacted. The stream composition is parameterized by the mixture fraction vector $Z_m$, where each of the $M$ component represents the fraction of mass that originated in that stream, where there are $N$ streams and $M = N - 1$. The mixture fraction for the final stream may be evaluated as $Z_N = 1 - \sum_{m=1}^{M} Z_m$.

The resulting flamelet data can then be assembled into a sequence of multi-dimensional tables of dependent variable $\phi$ as a function of the mixture fraction vector, $\phi(Z_m)$, and can be used directly for laminar simulations. Adding turbulence interactions, nonequilibrium effects, and nonadiabatic effects will increase the dimensionality of this lookup table and require additional processing. See the following sections for more information.

#### 2.10.1.2. Strained Laminar Flamelet Libraries

Strained laminar flamelet data may be generated for use in Fuego with the Spitfire code. This data is two-dimensional in nature, determined by the mixture fraction and a reference scalar dissipation rate $\chi_o$ at a reference mixture fraction $Z_o$. The instantaneous laminar scalar dissipation rate is defined as

$$\chi = 2D \frac{\partial Z}{\partial x_i} \frac{\partial Z}{\partial x_i}, \tag{2.315}$$

with D being the molecular mass diffusion coefficient. The reference value $\chi_o$ is arbitrary, although typical choices include the stoichiometric value $\chi_{\text{st}} = \chi(Z_{\text{st}})$ or the maximum value $\chi_{\text{max}} = \chi(Z = 0.5)$. Stoichiometric values are used in Fuego. After generating flamelet data with Spitfire, the `fuego_tabular_props` application can be used to assemble it into a sequence of multi-dimensional tables of dependent variable $\phi$ as a function of the mixture fraction vector and reference scalar dissipation rate, $\phi(Z_m, \chi_o)$. These tables may be then be used to evaluate properties in Fuego. Please see section 2.10.6 for details regarding the generation of flamelet libraries with Spitfire.

#### 2.10.1.3. Turbulent Averaging

In turbulent simulations, a filtered form of the governing equations are solved to reduce the resolution requirements to an affordable level. Temporal filtering is used in Reynolds Averaged Navier-Stokes (RANS) models and spatial filtering is used in Large Eddy Simulation (LES) models. Both types of filtering are represented with the notation $\bar{\phi}$, and are handled similarly in the present work. Density-weighted, or Favre filtering greatly simplifies the treatment of variable-density flow. A Favre-filtered quantity is represented by $\tilde{\phi} \equiv \overline{\rho\phi}/\bar{\rho}$. Please see the `fuego` theory manual for further details.

For use in turbulent simulations, a Favre-filtered version of the variables in the property table must be calculated. This is performed by convoluting the property variable with the joint PDF of the

independent variable sub-filter fluctuations, and is mathematically expressed as

$$\tilde{\phi}(\tilde{Z}_m, \widetilde{Z''^2}, \tilde{\chi}) = \int_0^\infty \int_0^1 \phi(Z_m, \chi_o)\; p_{Z\chi}(Z_i, \chi; \tilde{Z}_m, \widetilde{Z''^2}, \tilde{\chi})\, \mathrm{d}Z_i\, \mathrm{d}\chi, \tag{2.316}$$

where $p_{Z\chi}(Z_i, \chi; \tilde{Z}_m, \widetilde{Z''^2}, \tilde{\chi})$ is the joint PDF of sub-filter fluctuations of the dependent variable $\phi$ in $Z_i$-$\chi$ space, parameterized by the filtered mixture fractions $\tilde{Z}_m$ and the variance $\widetilde{Z''^2}$ of mixture fraction component $Z_i$, and the filtered scalar dissipation rate $\tilde{\chi}$. The reference scalar dissipation rate has the functionality $\chi_o(\tilde{Z}_i, \widetilde{Z''^2}, \tilde{\chi})$, which will be discussed in the following section. Variance of only a single component of mixture fraction, $Z_i$, is considered at present for simplicity, although extensions to include additional components are possible. Statistical independence will be assumed between $Z_i$ and $\chi$ fluctuations, so that

$$\tilde{\phi}(\tilde{Z}_m, \widetilde{Z''^2}, \tilde{\chi}) = \int_0^\infty \int_0^1 \phi(Z_m, \chi_o)\; p_Z(Z_i; \tilde{Z}_m, \widetilde{Z''^2})\, p_\chi(\chi; \tilde{\chi})\, \mathrm{d}Z_i\, \mathrm{d}\chi. \tag{2.317}$$

For the present work, $p_Z(Z_i; \tilde{Z}_m, \widetilde{Z''^2})$ will be modeled as either a beta PDF or a clipped Gaussian PDF and $p_\chi(\chi; \tilde{\chi})$ will be modeled as the delta function $\delta(\chi - \tilde{\chi})$.

#### 2.10.1.4. Property Table Implementation

The convolution integral in Equation 2.317 would be prohibitively expensive to evaluate each time a value for $\tilde{\phi}$ is needed by a turbulent reacting simulation. Therefore, this integral will be pre-calculated so that each property table query will only involve an interpolation from a table of values.

Storing the final $\tilde{\phi}(\tilde{Z}_m, \widetilde{Z''^2}, \tilde{\chi})$ values directly is undesirable since the range of possible $\tilde{\chi}$ values for each flamelet is different, resulting in a non-orthogonal table. Instead, the values $\tilde{\phi}_T(\tilde{Z}_m, \widetilde{Z''^2}, \chi_o)$ are stored in an orthogonal table that is indexed by $\tilde{Z}_m$, $\widetilde{Z''^2}$, and $\chi_o(\tilde{Z}_i, \widetilde{Z''^2}, \tilde{\chi})$. These tabulated values are calculated by

$$\tilde{\phi}_T(\tilde{Z}_m, \widetilde{Z''^2}, \chi_o) = \int_0^1 \phi(Z_m, \chi_o)\; p_Z(Z_i; \tilde{Z}_m, \widetilde{Z''^2})\, \mathrm{d}Z_i. \tag{2.318}$$

The reference scalar dissipation rate $\chi_o$ needed for lookup in the table for $\tilde{\phi}_T(\tilde{Z}_m, \widetilde{Z''^2}, \chi_o)$ can be evaluated from the local filtered scalar dissipation rate $\tilde{\chi}$ through laminar flamelet theory. The instantaneous scalar dissipation rate $\chi$ can be approximated by

$$\begin{aligned} \chi &= \chi_{\max} \exp\left(-2[\mathrm{erfc}^{-1}(2Z)]^2\right) \\ &= \chi_{\max} F_\chi(Z), \end{aligned} \tag{2.319}$$

where $\chi_{\max}$ is the maximum scalar dissipation rate found in the counterflow diffusion flame, which occurs at the stagnation point where $Z = 0.5$. (Note that this expression has not yet been extended to multiple mixture fractions, so that this treatment is only applicable for two-stream problems.) The

value of $\chi$ at any reference location in the flamelet can be similarly approximated, so that $\chi_o = \chi_{\max} F_\chi(Z_o)$. Combining these models by equating the unknown $\chi_{\max}$ yields a closed-form expression linking the scalar dissipation rate at any location to the reference value on the flamelet with the same characteristic $\chi_{\max}$,

$$\chi = \chi_o \frac{F_\chi(Z)}{F_\chi(Z_o)}. \tag{2.320}$$

Applying the filtering operation in Equation 2.317 to both sides of Equation 2.320 for a single-mixture fraction configuration yields

$$\begin{aligned}
\tilde{\chi} &= \int_0^\infty \int_0^1 \chi_o \frac{F_\chi(Z)}{F_\chi(Z_o)}\, p_Z(Z;\tilde{Z},\widetilde{Z''^2})\, p_\chi(\chi;\tilde{\chi})\, \mathrm{d}Z\, \mathrm{d}\chi && (2.321)\\
&= \frac{\chi_o}{F_\chi(Z_o)} \int_0^\infty p_\chi(\chi;\tilde{\chi})\, \mathrm{d}\chi \int_0^1 F_\chi(Z)\, p_Z(Z;\tilde{Z},\widetilde{Z''^2})\, \mathrm{d}Z && (2.322)\\
&= \frac{\chi_o}{F_\chi(Z_o)} \int_0^1 F_\chi(Z)\, p_Z(Z;\tilde{Z},\widetilde{Z''^2})\, \mathrm{d}Z, && (2.323)
\end{aligned}$$

so that the filtered reference scalar dissipation rate can be calculated from the filtered quantities provided by the turbulent flame simulation as

$$\chi_o(\tilde{Z},\widetilde{Z''^2},\tilde{\chi}) = \frac{\tilde{\chi}\, F_\chi(Z_o)}{\int_0^1 F_\chi(Z)\, p_Z(Z;\tilde{Z},\widetilde{Z''^2})\, \mathrm{d}Z,}. \tag{2.324}$$

To decrease computational cost, the integral in the denominator can be interpolated from pre-calculated values in a two-dimensional table as a function of $\tilde{Z}$ and $\widetilde{Z''^2}$.

To summarize, the turbulent reacting simulation will query the property table for the variable $\tilde{\phi}(\tilde{Z}_m,\widetilde{Z''^2},\tilde{\chi})$. Internally, Equation 2.324 will be used to calculate $\chi_o$ as a function of the provided filtered independent variables. This value will then be used along with the provided independent variables to interpolate a value for $\tilde{\phi}_T(\tilde{Z}_m,\widetilde{Z''^2},\chi_o)$ from the stored table that was pre-calculated with Equation 2.318. This interpolated value will then be returned to the main simulation as the requested value for $\tilde{\phi}(\tilde{Z}_m,\widetilde{Z''^2},\tilde{\chi})$.

If turbulence/chemistry interactions are to be neglected in the simulation, the delta function $\delta(Z-\tilde{Z})$ may be used for $p_Z(Z;\tilde{Z})$ in Equation 2.324 so that the reference scalar dissipation rate can be computed simply as

$$\chi_o(\tilde{Z},\tilde{\chi}) = \frac{\tilde{\chi}\, F(Z_o)}{F(\tilde{Z})}. \tag{2.325}$$

Once the multidimensional property table has been generated, it can be imported into `fuego` and queried for the dependent variables as a function of the independent variables $\tilde{Z}_m$, $\widetilde{Z''^2}$, and $\tilde{\chi}$. Models are required for each of these independent variables used by the flamelet property table. Sections 2.10.3–2.10.5 present models for each of these quantities for each of the supported turbulence closure models.

## 2.10.2. Nonadiabatic Property Table Generation

When including the effects of radiative or convective heat losses in a flamelet simulation, additional parameterizations beyond those in the previous section are required. These are the "conserved enthalpy", $h^*$ and heat loss parameter $\gamma$, where the heat loss parameter is defined as $\gamma = h - h^*$. The conserved enthalpy is identical to the traditional enthalpy except that its transport equation omits all source terms (typically due to radiative losses).

This formulation is used as a way to parameterize losses in a manner that is consistent with the opposed diffusion flame burner simulations used to generate the flamelet libraries. In these burner simulations, the inflowing pure stream states are fixed and cannot experience any heat losses; Losses only occur in the interior of the burner, and are represented by $\gamma$ variation. A range of inflowing pure stream states may also be computed, and are parameterized through $h^*$ variation. In this way, the full range of possible states may be tabulated and retrieved in a fire simulation through values of $h$ and $h^*$, which are both straightforward to compute.

For turbulent simulations, the Favre-filtered property variable $\tilde{\phi}$ is evaluated as

$$\begin{aligned}\tilde{\phi}(\tilde{Z}_m, \widetilde{Z''^2}, \tilde{\chi}, \tilde{\gamma}, \tilde{h}^*) \;=\; & \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{0}^{\infty}\int_{0}^{1} \phi(Z_m, \chi_o, \gamma_o, h_o^*)\; p_{Z\chi\gamma h^*}(Z_m, \chi, \gamma, h^*; \\ & \tilde{Z}_m, \widetilde{Z''^2}, \tilde{\chi}, \tilde{\gamma}, \tilde{h}^*)\, \mathrm{d}Z_m\, \mathrm{d}\chi\, \mathrm{d}\gamma\, \mathrm{d}h^*, \end{aligned} \tag{2.326}$$

where $\gamma_o$ and $h_o^*$ are reference values of the heat loss parameter and the conserved enthalpy, respectively, to be defined in the following sections. Statistical independence will be assumed between fluctuations of each $Z_m$ component, $\chi$, $\gamma$, and $h^*$, so that

$$\begin{aligned}\tilde{\phi}(\tilde{Z}_m, \widetilde{Z''^2}, \tilde{\chi}, \tilde{\gamma}, \tilde{h}^*) \;=\; & \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{0}^{\infty}\int_{0}^{1}\int_{0}^{1} \phi(Z_m, \chi_o, \gamma_o, h_o^*)\; p_{Z_i}(Z_i; \tilde{Z}_i, \widetilde{Z''^2})\, p_{Z_m}(Z_m; \tilde{Z}_m) \\ & p_\chi(\chi; \tilde{\chi})\, p_\gamma(\gamma; \tilde{\gamma})\, p_{h^*}(h^*; \tilde{h}^*)\, \mathrm{d}Z_i\, \mathrm{d}Z_{m\neq i}\, \mathrm{d}\chi\, \mathrm{d}\gamma\, \mathrm{d}h^*. \end{aligned} \tag{2.327}$$

For the present work, $p_{Z_i}(Z_i; \tilde{Z}_i, \widetilde{Z''^2})$ will be modeled as either a beta PDF or a clipped Gaussian PDF, and $p_{Z_m}(Z_m; \tilde{Z}_m)$, $p_\chi(\chi; \tilde{\chi})$, $p_\gamma(\gamma; \tilde{\gamma})$, and $p_{h^*}(h^*; \tilde{h}^*)$ will be modeled as the delta functions $\delta(Z_m - \tilde{Z}_m)$, $\delta(\gamma - \tilde{\gamma})$, and $\delta(\chi - \tilde{\chi})$, $\delta(\gamma - \tilde{\gamma})$, and $\delta(h^* - \tilde{h}^*)$, respectively.

The convolution integral in Equation 2.327 would be prohibitively expensive to evaluate each time a value for $\tilde{\phi}$ is needed by a turbulent reacting simulation. Therefore, this integral will be pre-calculated so that each property table query will only involve an interpolation from a table of values.

Storing the final $\tilde{\phi}(\tilde{Z}_m, \widetilde{Z''^2}, \tilde{\chi}, \tilde{\gamma}, \tilde{h^*})$ values directly is undesirable since the range of possible $\tilde{\chi}$, $\tilde{\gamma}$, and $\tilde{h}^*$ values for each flamelet is different, resulting in a non-orthogonal table. Instead, the values $\tilde{\phi}_T(\tilde{Z}_m, \widetilde{Z''^2}, \chi_o, \gamma_o, h_o^*)$ are stored in an orthogonal table that is indexed by $\tilde{Z}_m$, $\widetilde{Z''^2}$, $\chi_o(\tilde{Z}_i, \widetilde{Z''^2}, \tilde{\chi})$,

$\gamma_o(\tilde{Z}_i, \widetilde{Z''^2}, \tilde{\gamma})$, and $h_o^*(\tilde{Z}_i, \tilde{h}^*)$. These tabulated values are calculated by

$$\tilde{\phi}_T(\tilde{Z}_m, \widetilde{Z''^2}, \chi_o, \gamma_o, h_o^*) = \int_0^1 \int_0^1 \phi(Z_m, \chi_o, \gamma_o, h_o^*)\; p_{Z_i}(Z_i; \tilde{Z}_i, \widetilde{Z''^2})\, p_{Z_m}(Z_m; \tilde{Z}_m)\, \mathrm{d}Z_i\, \mathrm{d}Z_{m\neq i}. \tag{2.328}$$

The required reference values of $\gamma_o$ and $h_o^*$ are described in the following sections.

#### 2.10.2.1. Boundary heat loss

The addition of a temperature boundary condition on the wall requires a modification of the flamelet formulation of Eq. 2.314. The equation for a normalized temperature variable, $\theta = T/T_{\text{ox.}} - 1$, is

$$\partial_t(\rho\theta) + \nabla\cdot(\rho\mathbf{u}\theta) - \nabla\cdot(\lambda\nabla\theta) = L_D\theta = \dot{\omega}_T(\theta, \vec{Y}) \text{ in } \Omega \tag{2.329}$$

$$\theta(x \in \partial\Omega_{\text{fuel}}) = \theta_{\text{fuel}},\ \theta(x \in \partial\Omega_{\text{ox}}) = 0,\ \text{and } \theta(x \in \partial\Omega_{\text{wall}}) = \theta_{\text{wall}}, \tag{2.330}$$

which now has an extra boundary term $\theta(x \in \partial\Omega_{\text{wall}}) = \theta_{\text{wall}}$. The extra boundary condition remains when we apply the flamelet approximation, leaving $\theta(Z = Z_{\text{wall}}) = \theta_{\text{wall}}$. The value of the mixture fraction at the wall, however, is undecided: we only know that $\nabla Z \cdot n = 0$. During the simulation, the value of mixture fraction directly evaluated at the wall can be determined dynamically and the value of temperature can be computed. Away from the wall, however, one in principle would need to follow the $\zeta$ coordinate from the flamelet transformation until it intersects the wall. However, given that $\nabla Z \cdot n = 0$, the gradient trajectory in principle is tangential to the wall. Although the flamelet equation itself is well-posed, the asymptotic derivation of the flamelet model in the very near region to a nonadiabatic wall and the equations need to modified in some fashion to account.

Flamelets can readily be described when they are adiabatic; in the limit of unity Lewis numbers and adiabatic systems the enthalpy is a linear function of the mixture fraction. The existence of radiative transport and wall heat transfer introduces deviations from this linear relationship between $h$ and $Z$. Heat losses at the predominant boundary temperature are a common scenario. Defining a reference 'boundary temperature' at $T_{\text{fr}}(Z) = (1-Z)T_{\text{ox}} + ZT_{\text{fuel}}$, then this case a simplified flamelet temperature equation with heat losses could be written as

$$\rho\partial_t T - \frac{1}{2}\rho\chi\partial_Z^2 T = \dot{\omega}_T(\vec{\Phi}) - H_T(T - T_{\text{fr}}) \tag{2.331}$$

where $H_T$ represents a heat transfer coefficient that will be further discussed below. This gives a heat loss term that is linear in $T$. Alternately, the heat loss can be written specifically for radiative-style losses, $\dot{q}_{\text{losses}} = \sigma(\vec{\Phi})(T^4 - T_{\text{fr}}^4)$. Regardless, with heat loss expressed in terms of $T_{\text{fr}}$ the flamelet enthalpy is no longer linear in $Z$ but instead takes on a roughly inverted triangular form with an extrema at the peak temperature, roughly $Z_{st}$. This has led us to express the difference between the adiabatic enthalpy, defined as $h_c = h_{\text{ox}}(1-Z) + h_{\text{fuel}}Z$, and the actual flamelet computed enthalpy, $h$, as $\gamma = h - h_c$. The introduction of $\gamma$ is done strictly as an expedient for generation of flamelet libraries. By assuming a triangular form (or any particular assumed form) we can stretch the table entries into a square format by tabulating as a function of the stoichiometric value of $\gamma_{st}$. This does require the use of the assumed form for $\gamma$ for converting from the local $Z$ value of $\gamma = h(Z) - h_c(Z)$ to $\gamma_{\text{st}}$. Comparison with DNS

and unsteady flamelets for laminar flames shows good agreement with this type of enthalpy defect model for radiation in unity Lewis number flames (which is an appropriate assumption for turbulent, hydrocarbon fires) [52]. We make an assumption of path independence for the solution of at particular integrated heat loss, but in reality the solution will depend somewhat on the value of $H_T$ and the form of the added heat loss term.

The compensation for boundary heat loss can be extended to a full range of temperature (below $T_{\text{fr.}}$) in the flamelet libraries by not only including an integrated heat loss rate from the flamelets, but also a translation of the flamelet in enthalpy space. This translation is simply denoted $h^\star$, where the conserved enthalpy line is shifted as $h_c = h_{\text{ox}}(1-Z) + h_{\text{fuel}}Z + h^\star$. This allows a ful description of wall boundary heat loss. Having two heat loss parameterizations, however, makes the lookup procedure non-unique, requiring a method for deciding which point on $(\gamma, h^\star)$ to use for the flamelet lookup. We prefer $\gamma$ and use $\gamma$ as much as possible. When $\gamma$ is insufficient, which would is the case for overly cold or hot walls (wall temperatures outside of the range of temperatures spanned by the solutions of Eq. 2.331). At the wall boundaries, the conserved enthaply is defined to not be affected by heat loss while the true enthalpy is, providing $\gamma$ at the wall.

#### 2.10.2.2. Property Table Heat Loss Parameterization

For nonadiabatic flamelet library generation and tabulation, a functional form for the heat loss parameter $\gamma$ in terms of reference quantities is required, similar in concept to the form of $\chi$ in Equation 2.319. The value of $\gamma$ must be zero in each of the pure streams, and should have a maximum value near the stoichiometric flame sheet since this quantity typically represents radiative losses to the environment. A piecewise linear functional form is selected for simplicity. For a single mixture fraction, this form is simply

$$\gamma = \gamma_o \, F_\gamma(Z, Z_o), \tag{2.332}$$

where $\gamma_o$ is a reference heat loss at reference state $Z_o$ (selected to be the stoichiometric condition $Z_{\text{st}}$) and the nondimensional function $F_\gamma(Z, Z_o)$ is defined as

$$F_\gamma(Z, Z_o) = \begin{cases} \frac{Z}{Z_o} & : \quad Z \leq Z_o \\ \frac{1-Z}{1-Z_o} & : \quad Z > Z_o \end{cases} . \tag{2.333}$$

For multiple mixture fractions, $\gamma$ is calculated by

$$\gamma = \gamma_o \, F_\gamma(Z_m, Z_{o,km}, \gamma_{o,k}^{\max}), \tag{2.334}$$

where $\gamma_o$ is the maximum-magnitude reference heat loss in the vector $\gamma_{o,k}^{\max}$, which contains the reference heat loss parameters corresponding to maximum thermal losses for the $K$ stoichiometric mixture fractions that can be defined between stream pairs, $Z_{o,km}$. The multiple stoichiometric mixture fractions are necessary because a single unique stoichiometric mixture fraction does not exist when using multiple mixture fractions.

The functional form for $F_\gamma$ is quite complex for multiple mixture fractions, and will only be described briefly here. In general, for a three-stream problem, there are two independent mixture fractions and the realizable mixture fraction space is the triangle where the two mixture fractions sum to a value less than

or equal to unity. The value of $F_\gamma$ must be zero at the "corners" of this space, where the coordinates are $(0,0)$, $(0,1)$, and $(1,0)$. The multiple stoichiometric mixture fractions between stream pairs will define points along the boundaries of this realizable mixture fraction space that represent local maxima in the heat loss distribution along that boundary. Straight lines may be used to connect these points in mixture fraction space, forming a "ridge" in the multidimensional $F_\gamma$ distribution. When definable, a linear fit is used between this ridge and a corner where $\gamma$ is zero. When not uniquely definable, linear fits are used between the ridge and the adjacent boundary value along rays extended from the opposite corner of the state space. Note that the values $\gamma_{o,k}$ are required for the calculation of $F_\gamma$ so that the final function may be normalized to a unity maximum value with appropriate relative scaling between the boundary heat loss values. Note that no more than a three-stream configuration is currently supported by `fuego_tabular_props`.

Applying the filtering operation in Equation 2.327 to both sides of Equation 2.334 yields

$$\begin{aligned}
\tilde{\gamma} &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{0}^{\infty}\int_{0}^{1}\int_{0}^{1} \gamma_o\, F_\gamma(Z_m, Z_{o,km}, \gamma_{o,k}^{\max})\, p_{Z_i}(Z_i; \tilde{Z}_i, \widetilde{Z''^2})\, p_{Z_m}(Z_m; \tilde{Z}_m) \\
&\quad p_\chi(\chi;\tilde{\chi})\, p_\gamma(\gamma;\tilde{\gamma})\, p_{h^*}(h^*;\tilde{h}^*)\, \mathrm{d}Z_i\, \mathrm{d}Z_{m\neq i}\, \mathrm{d}\chi\, \mathrm{d}\gamma\, \mathrm{d}h^* \qquad (2.335) \\
&= \gamma_o \int_{-\infty}^{\infty} p_{h^*}(h^*;\tilde{h}^*)\, \mathrm{d}h^* \int_{-\infty}^{\infty} p_\gamma(\gamma;\tilde{\gamma})\, \mathrm{d}\gamma \int_{0}^{\infty} p_\chi(\chi;\tilde{\chi})\, \mathrm{d}\chi \\
&\quad \int_{0}^{1}\int_{0}^{1} F_\gamma(Z_m, Z_{o,km}, \gamma_{o,k}^{\max})\, p_{Z_i}(Z_i; \tilde{Z}_i, \widetilde{Z''^2})\, p_{Z_m}(Z_m; \tilde{Z}_m)\, \mathrm{d}Z_i\, \mathrm{d}Z_{m\neq i} \qquad (2.336) \\
&= \gamma_o \int_{0}^{1}\int_{0}^{1} F_\gamma(Z_m, Z_{o,km}, \gamma_{o,k}^{\max})\, p_{Z_i}(Z_i; \tilde{Z}_m, \widetilde{Z''^2})\, p_{Z_m}(Z_m; \tilde{Z}_m)\, \mathrm{d}Z_i\, \mathrm{d}Z_{m\neq i}, \qquad (2.337)
\end{aligned}$$

so that the filtered heat loss parameter can be calculated from the filtered quantities provided by the turbulent flame simulation as

$$\gamma_o(\tilde{Z}_m, \widetilde{Z''^2}, \tilde{\gamma}) = \frac{\tilde{\gamma}}{\int_0^1\int_0^1 F_\gamma(Z_m, Z_{o,km}, \gamma_{o,k}^{\max})\, p_{Z_i}(Z_i; \tilde{Z}_i, \widetilde{Z''^2})\, p_{Z_m}(Z_m; \tilde{Z}_m)\, \mathrm{d}Z_i\, \mathrm{d}Z_{m\neq i}}. \qquad (2.338)$$

To decrease computational cost, the integral in the denominator can be interpolated from pre-calculated values in a multi-dimensional table as a function of $\tilde{Z}_m$ and $\widetilde{Z''^2}$. Equation 2.338 can be used during a simulation to convert filtered independent variables to the reference heat loss parameter required to perform table lookups to retrieve $\tilde{\phi}_T$.

If turbulence/chemistry interactions are to be neglected in the simulation, the delta function $\delta(Z_i - \tilde{Z}_i)$ may be used for $p_{Z_i}(Z_i; \tilde{Z}_i, \widetilde{Z''^2})$ in Equation 2.338 so that the reference heat loss parameter can be computed simply as

$$\gamma_o(\tilde{Z}_m, \tilde{\gamma}) = \frac{\tilde{\gamma}}{F_\gamma(\tilde{Z}_m, Z_{o,km}, \gamma_{o,k}^{\max})}. \qquad (2.339)$$

#### 2.10.2.3. Property Table Conserved Enthalpy Parameterization

For nonadiabatic flamelet library generation and tabulation, a functional form for the conserved enthalpy $h^*$ in terms of reference quantities is required. The value of $h^*$ should vary linearly within the range provided for each of the pure streams as a function of a reference heat loss parameter $h^*_o$, with an appropriate stream-weighted blending for all other compositions.

The stream-weighted mixture properties are computed with an augmented mixture fraction vector $Z'_n$ in terms of $Z_m$,

$$Z'_n = \left[ Z_1, Z_2, \ldots, Z_M, 1 - \sum_{m=1}^{M} Z_m \right], \tag{2.340}$$

where the last component is simply the last implied mixture fraction to recover a unity sum. A reference augmented mixture fraction is defined as the centroid of the realizable mixture fraction space with each component being identical and equal to

$$Z'_{o,n} = \frac{1}{N}. \tag{2.341}$$

From these definitions, minimum and maximum reference conserved enthalpy values may be computed as

$$h^*_{o,\min} = \sum_{n=1}^{N} h^*_{\text{stream,min},n} \, Z'_{o,n} \tag{2.342}$$

$$h^*_{o,\max} = \sum_{n=1}^{N} h^*_{\text{stream,max},n} \, Z'_{o,n}, \tag{2.343}$$

where $h^*_{\text{stream,min},n}$ and $h^*_{\text{stream,max},n}$ are vectors of the minimum and maximum conserved enthalpy in pure stream $n$, respectively. The conserved enthalpy can then be modeled as

$$h^* = h^*_{\min,Z} + \left( h^*_o - h^*_{o,\min} \right) a_Z, \tag{2.344}$$

where the mixture-weighted minimum conserved enthalpy is

$$h^*_{\min,Z} = \sum_{n=1}^{N} h^*_{\text{stream,min},n} \, Z'_n \tag{2.345}$$

and the mixture-weighted stream variation proportionality constant is

$$a_Z = \sum_{n=1}^{N} \left( \frac{h^*_{\text{stream,max},n} - h^*_{\text{stream,min},n}}{h^*_{o,\max} - h^*_{o,\min}} \right) Z'_n. \tag{2.346}$$

Applying the filtering operation in Equation 2.327 to both sides of Equation 2.344 yields

$$\tilde{h}^* = \tilde{h}^*_{\min,Z} + \left( h^*_o - h^*_{o,\min} \right) \tilde{a}_Z, \tag{2.347}$$

where the two mixture-weighted quantities are now expressed in terms of the augmented filtered mixture fraction as

$$\tilde{h}^*_{\text{min},Z} = \sum_{n=1}^{N} h^*_{\text{stream,min},n}\, \tilde{Z}'_n \tag{2.348}$$

and

$$\tilde{a}_Z = \sum_{n=1}^{N} \frac{h^*_{\text{stream,max},n} - h^*_{\text{stream,min},n}}{h^*_{o,\max} - h^*_{o,\min}}\, \tilde{Z}'_n. \tag{2.349}$$

This allows the reference conserved enthalpy to be expressed in terms of the filtered quantities provided by the turbulent flame simulation as

$$h^*_o(\tilde{Z}_m, \tilde{h}^*) = h^*_{o,\min} + \frac{\tilde{h}^* - \tilde{h}^*_{\text{min},Z}}{\tilde{a}_Z}. \tag{2.350}$$

## 2.10.3. Filtered Scalar Dissipation Rate

### *2.10.3.1. RANS Model*

For RANS turbulence closure models the instantaneous laminar scalar dissipation rate given in Equation 2.315 can be Favre-filtered and expanded to the form

$$\bar{\rho}\tilde{\chi} = \overline{2\rho D \frac{\partial Z}{\partial x_i}\frac{\partial Z}{\partial x_i}} \tag{2.351}$$

$$= 2\overline{\rho D}\frac{\partial \tilde{Z}}{\partial x_i}\frac{\partial \tilde{Z}}{\partial x_i} + \overline{4\rho D \frac{\partial Z''}{\partial x_i}\frac{\partial \tilde{Z}}{\partial x_i}} + \overline{2\rho D \frac{\partial Z''}{\partial x_i}\frac{\partial Z''}{\partial x_i}}. \tag{2.352}$$

The middle term on the RHS is neglected for constant density flow [53]. The first term is referred to as the mean scalar dissipation rate

$$\bar{\rho}\tilde{\chi}_m = 2\overline{\rho D}\frac{\partial \tilde{Z}}{\partial x_i}\frac{\partial \tilde{Z}}{\partial x_i} \tag{2.353}$$

and the third term is the perturbation scalar dissipation rate $\bar{\rho}\tilde{\chi}_p$. This term can be modeled as

$$\bar{\rho}\tilde{\chi}_p = \overline{2\rho D \frac{\partial Z''}{\partial x_i}\frac{\partial Z''}{\partial x_i}} \tag{2.354}$$

$$\approx C_\chi \bar{\rho}\frac{\epsilon}{k}\widetilde{Z''^2} \tag{2.355}$$

for RANS-based turbulence closures where $\frac{\epsilon}{k}$ provides an inverse turbulence time scale, $\widetilde{Z''^2}$ is the scalar variance that will be modeled in Section 2.10.5, and $C_\chi$ is a model constant that typically has a value of 2.0. [51]

Expressing the molecular mass diffusivity as $\bar{\rho}D = \mu/\text{Sc}$, where $\mu$ is the molecular viscosity and Sc is the Schmidt number, the modeled total filtered scalar dissipation rate for RANS closures is

$$\tilde{\chi} = \tilde{\chi}_m + \tilde{\chi}_p \tag{2.356}$$

$$\approx \frac{2}{\bar{\rho}}\frac{\mu}{\text{Sc}}\frac{\partial \tilde{Z}}{\partial x_i}\frac{\partial \tilde{Z}}{\partial x_i} + C_\chi \frac{\epsilon}{k}\widetilde{Z''^2}. \tag{2.357}$$

#### 2.10.3.2. LES Model

For LES closures Equation 2.356 also applies, so that the total filtered scalar dissipation rate is the sum of the mean and the perturbation scalar dissipation rates. The mean scalar dissipation rate is expressed identically to RANS closures as

$$\tilde{\chi}_m = \frac{2}{\bar{\rho}} \frac{\mu}{\mathrm{Sc}} \frac{\partial \tilde{Z}}{\partial x_i} \frac{\partial \tilde{Z}}{\partial x_i}. \tag{2.358}$$

The perturbation scalar dissipation rate $\tilde{\chi}_p$ represents the sub-filter dissipation of scalar variance, and can be modeled by assuming that sub-filter dissipation is in local equilibrium with sub-filter production, and that the sub-filter production can be modeled with a gradient transport assumption as [54]

$$\begin{aligned} \bar{\rho}\tilde{\chi}_p &= 2\overline{\rho D \frac{\partial Z''}{\partial x_i} \frac{\partial Z''}{\partial x_i}} = -2\overline{\rho u_i'' Z''} \frac{\partial \tilde{Z}}{\partial x_i} && (2.359) \\ &\approx 2 \frac{\mu_t}{\mathrm{Sc}_t} \frac{\partial \tilde{Z}}{\partial x_i} \frac{\partial \tilde{Z}}{\partial x_i}, && (2.360) \end{aligned}$$

where $\mu_t$ is the modeled turbulent eddy viscosity and $\mathrm{Sc}_t$ is the turbulent Schmidt number.

This results in the final modeled form for the filtered total scalar dissipation rate for LES closures,

$$\begin{aligned} \tilde{\chi} &= \tilde{\chi}_m + \tilde{\chi}_p && (2.361) \\ &\approx \frac{2}{\bar{\rho}} \left( \frac{\mu}{\mathrm{Sc}} + \frac{\mu_t}{\mathrm{Sc}_t} \right) \frac{\partial \tilde{Z}}{\partial x_i} \frac{\partial \tilde{Z}}{\partial x_i}. && (2.362) \end{aligned}$$

### 2.10.4. Filtered Mixture Fraction

The primary quantity used to identify the chemical state in Flamelet closure models is the mixture fraction, $Z$. While there are many different definitions of the mixture fraction that have subtle variations that attempt to capture effects like differential diffusion, they can all be interpreted as a local mass fraction of the chemical elements that originated in the fuel stream. [55] The mixture fraction is a conserved scalar that varies between 0 in the oxidizer stream and 1 in the fuel stream, and is transported in laminar flow by the equation

$$\frac{\partial \rho Z}{\partial t} + \frac{\partial \rho u_i Z}{\partial x_i} = \frac{\partial}{\partial x_i} \left( \rho D \frac{\partial Z}{\partial x_i} \right), \tag{2.363}$$

where $D$ is an effective molecular mass diffusivity.

Applying either temporal Favre filtering for RANS-based treatments or spatial Favre filtering for LES-based treatments yields

$$\frac{\partial \bar{\rho}\tilde{Z}}{\partial t} + \frac{\partial \bar{\rho}\tilde{u}_i \tilde{Z}}{\partial x_i} = -\tau_{Zu_j} + \frac{\partial}{\partial x_i} \left( \bar{\rho} D \frac{\partial \tilde{Z}}{\partial x_i} \right), \tag{2.364}$$

where sub-filter correlations have been neglected in the molecular diffusive flux vector [56] and the turbulent diffusive flux vector is defined as

$$\tau_{Zu_j} \equiv \bar{\rho}\left(\widetilde{Zu_i} - \tilde{Z}\tilde{u}_i\right). \tag{2.365}$$

Similar to species transport, this sub-filter correlation is modeled in both RANS and LES closures with the gradient transport approximation

$$\tau_{Zu_j} \approx -\bar{\rho}D_t\frac{\partial Z}{\partial x_i}, \tag{2.366}$$

where $D_t$ is the turbulent mass diffusivity, modeled as $\bar{\rho}D_t = \mu_t/\mathrm{Sc}_t$ where $\mu_t$ is the modeled turbulent viscosity from momentum transport and $\mathrm{Sc}_t$ is the turbulent Schmidt number. Please see the Fuego theory manual for further details. The molecular mass diffusivity is then expressed similarly as $\bar{\rho}D = \mu/\mathrm{Sc}$ so that the final modeled form of the filtered mixture fraction transport equation is

$$\frac{\partial\bar{\rho}\tilde{Z}}{\partial t} + \frac{\partial\bar{\rho}\tilde{u}_i\tilde{Z}}{\partial x_i} = \frac{\partial}{\partial x_i}\left[\left(\frac{\mu}{\mathrm{Sc}} + \frac{\mu_t}{\mathrm{Sc}_t}\right)\frac{\partial\tilde{Z}}{\partial x_i}\right]. \tag{2.367}$$

In integral form as used in Fuego, the mixture fraction transport equation is

$$\int\frac{\partial\bar{\rho}\tilde{Z}}{\partial t}\,dV + \int\bar{\rho}\tilde{u}_i\tilde{Z}n_i\,dS = \int\left(\frac{\mu}{\mathrm{Sc}} + \frac{\mu_t}{\mathrm{Sc}_t}\right)\frac{\partial\tilde{Z}}{\partial x_i}n_i\,dS. \tag{2.368}$$

## 2.10.5. Filtered Scalar Variance

### 2.10.5.1. RANS Model

For RANS-based turbulence closures, a transport equation is solved for the filtered scalar variance, $\widetilde{Z''^2}$. This equation can be derived by subtracting Equation 2.364 multiplied by $\tilde{Z}$ from the filter of the multiple of Equation 2.363 and $Z$, yielding

$$\begin{aligned}\frac{\partial\bar{\rho}\widetilde{Z''^2}}{\partial t} + \frac{\partial}{\partial x_i}\left(\bar{\rho}\tilde{u}_i\widetilde{Z''^2}\right) &= -\frac{\partial}{\partial x_i}\left(\overline{\rho u_i''Z''^2}\right) + \frac{\partial}{\partial x_i}\left(\overline{\rho D\frac{\partial Z''^2}{\partial x_i}}\right) + \overline{2Z''^2\frac{\partial}{\partial x_i}\left(\rho D\frac{\partial\tilde{Z}}{\partial x_i}\right)} \\ &\quad -2\overline{\rho u_i''Z''}\frac{\partial\tilde{Z}}{\partial x_i} - \overline{2\rho D\frac{\partial Z''}{\partial x_i}\frac{\partial Z''}{\partial x_i}},\end{aligned} \tag{2.369}$$

where the filtered mixture fraction variance is defined as $\widetilde{Z''^2} \equiv \widetilde{Z^2} - \tilde{Z}^2$.

All five terms on the RHS of Equation 2.369 require closure models. The first term represents turbulent transport of mixture fraction variance, and is modeled by a gradient-transport assumption as

$$-\overline{\rho u_i''Z''^2} \approx \frac{\mu_t}{\mathrm{Sc}_t}\frac{\partial\widetilde{Z''^2}}{\partial x_i}. \tag{2.370}$$

The second and third terms on the RHS of Equation 2.369 taken together represent molecular diffusion of mixture fraction variance, and is typically neglected with respect to turbulent transport for sufficiently high Reynolds numbers. Its effects are included here with another gradient-transport assumption of the form

$$\frac{\partial}{\partial x_i}\left(\overline{\rho D\frac{\partial Z''^2}{\partial x_i}}\right)+\overline{2Z''^2\frac{\partial}{\partial x_i}\left(\rho D\frac{\partial \tilde{Z}}{\partial x_i}\right)}\approx\frac{\partial}{\partial x_i}\left(\frac{\mu}{\mathrm{Sc}}\frac{\partial\widetilde{Z''^2}}{\partial x_i}\right). \tag{2.371}$$

The fourth and fifth terms on the RHS of Equation 2.369 represent production and dissipation of mixture fraction variance, respectively. The production term is similarly modeled with a gradient transport assumption as

$$-2\overline{\rho u_i'' Z''}\frac{\partial\tilde{Z}}{\partial x_i}\approx 2\frac{\mu_t}{\mathrm{Sc}_t}\frac{\partial\tilde{Z}}{\partial x_i}\frac{\partial\tilde{Z}}{\partial x_i}. \tag{2.372}$$

The mixture fraction variance dissipation rate term is equal to the perturbation scalar dissipation rate,

$$\overline{2\rho D\frac{\partial Z''}{\partial x_i}\frac{\partial Z''}{\partial x_i}}=\bar{\rho}\tilde{\chi}_p, \tag{2.373}$$

previously defined in Equation 2.354 and modeled in Equation 2.355. An identical treatment of this term is used here.

The final modeled form of the filtered scalar variance transport equation for RANS turbulence closure models is

$$\frac{\partial\bar{\rho}\widetilde{Z''^2}}{\partial t}+\frac{\partial}{\partial x_i}\left(\bar{\rho}\tilde{u}_i\widetilde{Z''^2}\right)=\frac{\partial}{\partial x_i}\left[\left(\frac{\mu}{\mathrm{Sc}}+\frac{\mu_t}{\mathrm{Sc}_t}\right)\frac{\partial\widetilde{Z''^2}}{\partial x_i}\right]+2\frac{\mu_t}{\mathrm{Sc}_t}\frac{\partial\tilde{Z}}{\partial x_i}\frac{\partial\tilde{Z}}{\partial x_i}-\bar{\rho}\tilde{\chi}_p. \tag{2.374}$$

#### 2.10.5.2. LES Model

For LES turbulence closures, the filtered scalar variance $\widetilde{Z''^2}$ can be modeled with the scaling law [54]

$$\bar{\rho}\widetilde{Z''^2}\approx C_V\bar{\rho}\Delta^2\frac{\partial\tilde{Z}}{\partial x_i}\frac{\partial\tilde{Z}}{\partial x_i}, \tag{2.375}$$

where $\Delta$ is a length scale corresponding to the grid filter size and $C_V$ is a model coefficient. For the $k^{\mathrm{sgs}}$ closure and the non-dynamic Smagorinsky closure, $C_V$ has a fixed value of $0.5$. For the dynamic Smagorinsky LES closure, $C_V$ can be dynamically calculated based on the local instantaneous flowfield.

To dynamically evaluate the filtered scalar variance model coefficient, begin by defining the grid filter-scale correlation

$$\begin{aligned}\tau_{Z''^2} &\equiv \bar{\rho}\widetilde{Z''^2} && (2.376)\\ &= \bar{\rho}\widetilde{Z^2}-\bar{\rho}\tilde{Z}^2 && (2.377)\\ &= \overline{\rho Z^2}-\frac{\left(\overline{\rho Z}\right)^2}{\bar{\rho}}. && (2.378)\end{aligned}$$

Similarly, define an equivalent correlation at a larger test-filter scale

$$T_{Z''^2} \equiv \widehat{\overline{\rho Z^2}} - \frac{\left(\widehat{\overline{\rho Z}}\right)^2}{\hat{\bar{\rho}}}. \tag{2.379}$$

Now, define the quantity $L_{Z''^2}$ as a combination of these two correlations which reduces to an expression that can be evaluated in closed form,

$$L_{Z''^2} \equiv T_{Z''^2} - \widehat{\tau_{Z''^2}} \tag{2.380}$$

$$= \widehat{\bar{\rho}\tilde{Z}^2} - \frac{\left(\widehat{\bar{\rho}\tilde{Z}}\right)^2}{\hat{\bar{\rho}}}. \tag{2.381}$$

By modeling the two correlations in Equation 2.380 and equating them to Equation 2.381, the model coefficient $C_V$ can be dynamically evaluated. The correlations at the two filter scales are modeled analogously as

$$\tau_{Z''^2} \approx C_V \bar{\rho}\Delta^2 \left(\frac{\partial \tilde{Z}}{\partial x_i}\right)^2 \tag{2.382}$$

$$T_{Z''^2} \approx C_V \hat{\bar{\rho}}\hat{\Delta}^2 \left[\frac{\partial}{\partial x_i}\left(\frac{\widehat{\bar{\rho}\tilde{Z}}}{\hat{\bar{\rho}}}\right)\right]^2, \tag{2.383}$$

where $\hat{\Delta}$ is the characteristic test filter length scale and $C_V$ is assumed to be the same at both scales.

Notice that when the modeled forms of $\tau_{Z''^2}$ and $T_{Z''^2}$ are inserted into Equation 2.380, $C_V$ appears inside a test filtering operation. Formally solving this system of equations for $C_V$ requires the expensive solution of an additional set of coupled integro-differential equations [57]. Alternatively, it is common practice to remove $C_V$ from the test filter with the assumption that it is varying slowly over distances on the order of the test filter size. This greatly simplifies calculations, although it can result in non-physical oscillations in the modeled value for $C_V$. The square of the error involved in this approximation is $Q = (L_{Z''^2} - C_V M_{Z''^2})^2$, where

$$L_{Z''^2} = \widehat{\bar{\rho}\tilde{Z}^2} - \frac{\left(\widehat{\bar{\rho}\tilde{Z}}\right)^2}{\hat{\bar{\rho}}} \tag{2.384}$$

$$M_{Z''^2} = \hat{\bar{\rho}}\hat{\Delta}^2 \left[\frac{\partial}{\partial x_i}\left(\frac{\widehat{\bar{\rho}\tilde{Z}}}{\hat{\bar{\rho}}}\right)\right]^2 - \widehat{\bar{\rho}\Delta^2 \left(\frac{\partial \tilde{Z}}{\partial x_i}\right)^2}. \tag{2.385}$$

Minimizing this error in a least-squares fashion with respect to $C_V$ yields an expression for the modeled coefficient,

$$C_V = \frac{L_{Z''^2} M_{Z''^2}}{M_{Z''^2} M_{Z''^2}}, \tag{2.386}$$

that can be used directly in Equation 2.375 for the filtered scalar variance.

Due to the above simplifications, the model coefficient $C_V$ can sometimes fluctuate wildly, possibly leading to numerical instabilities. A common solution to control these oscillations, and the one that is taken here, is to pass the numerator and denominator of Equation 2.386 through a test filter, yielding

$$C_V = \frac{\widehat{L_{Z''^2} M_{Z''^2}}}{\widehat{M_{Z''^2} M_{Z''^2}}}. \tag{2.387}$$

This can be crudely justified by recognizing that $C_V$ was already assumed to vary slowly over distances equal to the test filter size, so that this filtering operation is simply enforcing that assumption.

## 2.10.6. Generating Flamelet Libraries with Spitfire

### *2.10.6.1. Introduction*

Spitfire is a Python-C++ code used to solve low-dimensional combustion problems as a preprocessing step in building tabulated chemistry models for turbulent, sooting fires. In tabulated chemistry models we constrain the thermochemical state space to a manifold over a small number of reaction variables which are often conserved scalars. There are two essential steps in building a tabulated chemistry model. First, solve representative problems such as perfectly stirred reactors or nonpremixed flamelet models, often performing continuation over parameters such as residence times, scalar dissipation rates, or heat loss parameters. Second, use the solution data to build a property reconstruction model with either structured interpolation tables or unstructured regression models (*e.g.*, artificial neural networks). Currently Spitfire solves nonpremixed flamelet equations over parameters of interest and forms the data into a structured table, after which the Tabular Props application is used to build multidimensional B-spline representations of properties which may be queried during a simulation by Fuego.

### *2.10.6.2. User Interfaces*

There are two ways for a user to build a flamelet library with Spitfire. The simplest interface, likely sufficient for most users, works through a simple YAML-style input file, which can be passed to generate any of a set of 'standard' tables. Section 2.10.6.3 shows several examples of the YAML interface.

The second interface, recommended for developers and advanced users, involves the use of Python. At the cost of a little bit of Python programming, this interface offers extensibility, generality, and more precise control than the input file interface. For instance, one can define arbitrarily-complex post processors for tabulated properties, which would enable straightforward parameter studies in soot modeling, for instance. Another examplary use of the Python interface is to generate nonadiabatic flamelet libraries with quasisteady heat loss (a core Spitfire regression test, actually) instead of the transient heat loss allowed by the input file interface. The conda package manager may be used to facilitate installation of Spitfire's external Python and C++ dependencies, as described in the `SpitfireCore` project in the `Spitfire` repository group at `https://cee-gitlab.sandia.gov/spitfirecodes`. Documentation and examples of the Python interface may be found there as well.

#### 2.10.6.3. Using the YAML Interface

On the CEE LAN the YAML interface to Spitfire has been sandboxed into a standalone executable. This removes the need for users to maintain a Python interpreter with Spitfire installed or manage runtime dependencies with paths and environment variables. This reduces the required input down to a chemical mechanism data file in Cantera XML format and an input file for Spitfire. A flamelet library may be generated with the following commands. First, load the `sierra` module to make the Tabular Props application available. Next, run the standalone `generate` executable on the `input.yaml` input file. The `-n 8` argument tells Spitfire to use eight cores wherever possible. The `-t` flag tells Spitfire to run Tabular Props on its output, meaning that after Tabular Props completes a library file will be ready for immediate use with Fuego. Next we discuss the required chemical mechanism data file and show some examples, spanning a simple mixing library to a nonadiabatic, turbulent strained laminar flamelet with soot source terms and radiation sources.

```
$ module load sierra
$ /usr/netpub/mahanse/spitfire_table/generate -i input.yaml -n 8 -t
```

Before showing some examples, we note the "parse-only" mode, which only parses the YAML input file and dumps a comprehensive version with all defaults included. This is helpful when learning how to specify new parameters or examining defaults that are implicitly set.

```
$ /usr/netpub/mahanse/spitfire_table/generate -i input.yaml -p
```

#### 2.10.6.4. Chemical Mechanism Data

Spitfire uses the Cantera code to manage the specification of thermodynamic (species heat capacities), transport (viscosity model parameters) and kinetic rate parameters. This data must be provided in Cantera's XML format. A number of chemical mechanism files are provided in the Spitfire code repositories, and the standalone executable provides several useful mechanism data files listed in Table 2.10-1. Many mechanisms of interest for fire scenarios are available online from various sources, and Cantera provides thorough online resources for building mechanism files from scratch if desired.

**Table 2.10-1.. Chemical mechanism data packaged into the standalone Spitfire executable**

| XML | Fuel | Reference |
|---|---|---|
| `heptane_LiuEtAl.xml` | heptane | Liu et al. (2004) |
| `ethylene_USC.xml` | ethylene | Wang et al. (2007) (USC-mech) |
| `ethylene_LuoEtAl.xml` | ethylene | Luo et al. (2012) (reduced USC-mech) |
| `methane_GRI30.xml` | methane | GRI-3.0 |

### 2.10.6.5. Simple Mixing Library

The following input is as simple as it gets, which may be used for a mixture of air and n-heptane that is not reacting (*e.g.*, cold flow). First, the `chemistry-type` of the mixture fraction model is specified. This must be one of `unreacted` (pure mixing), `Burke-Schumann` (idealized combustion), `equilibrium` (infinitely fast chemistry), or `SLFM` (strained laminar flamelet library). Next, the Cantera mechanism data file is specified (using the pre-packaged n-heptane mechanism). Following that, the mass fractions and temperature of the oxidizer and fuel streams are given. The mass fractions may be specified in any acceptable format for Cantera, but a comma-separated list as shown here is often the most convenient option.

```
mixture-fraction-model:
  chemistry-type: unreacted
  cantera:
    xml: heptane_LiuEtAl.xml
  oxidizer:
    temperature: 300
    mass-fractions: 'N2:0.767,O2:0.233'
  fuel:
    temperature: 300
    mass-fractions: 'NXC7H16:1'
```

This input will produce a flamelet library with the following properties: temperature, viscosity, density, and isobaric heat capacity of the mixture. By default, Spitfire will write output to the `spitfire_output` directory (set with `output-directory`), all intermediate data will be saved (set with `save-intermediate-data`), and the table for Fuego will be named `table_for_fuego.h5` (set with `fuego-table-file-name`). If defaults are unclear or you want to know what else could be specified, you can run Spitfire in parse-only mode to see a comprehensive summary of the expanded input file with all defaults and options. Another option is to generate a table and check the output directory for the file named `complete_spitfire_input.yaml`, which contains the same information as the parse-only mode produces.

### 2.10.6.6. Simple Equilibrium Library

There are a number of default parameters left unspecified in this first example. Among the most important are the `pressure` (defaults to 101325 Pa), `heat-loss-type` (defaults to `adiabatic`), and `turbulence-type` (defaults to `laminar`). For instance, to generate the data for an equilibrium chemistry model at two atmospheres, one could change the `chemistry-type` and directly set the pressure. In addition to the properties that are always tabulated, it may be interesting to visualize mass fractions of certain species in the Fuego simulation. They may be added as shown below.

For equilibrium chemistry it may be important to set details of the grid in the mixture fraction dimension. In the adjacent example the number of grid points is set to 96 (overwriting the default of 32) and details of the automatic grid clustering are specified. Spitfire uses a grid clustered around the stoichiometric mixture fraction by default (leave `grid-cluster-point` unspecified or specify `stoichiometric` to accomplish this). The default value of the `grid-cluster-intensity` is 4, and increasing this number leads to a more tightly clustered grid. In cases where a uniform grid is preferred, simply set the `grid-type` to `uniform`.

```
mixture-fraction-model:
  chemistry-type: equilibrium
  pressure: 202650
  cantera:
    xml: heptane_LiuEtAl.xml
  oxidizer:
    temperature: 300
    mass-fractions: 'N2:0.767,O2:0.233'
  fuel:
    temperature: 485
    mass-fractions: 'NXC7H16:1'
  tabulated-quantities:
    mass-fractions:
      - OH
      - C2H2
```

```
mixture-fraction-model:
  chemistry-type: equilibrium
  pressure: 202650
  cantera:
    xml: heptane_LiuEtAl.xml
  oxidizer:
    temperature: 300
    mass-fractions: 'N2:0.767,O2:0.233'
  fuel:
    temperature: 485
    mass-fractions: 'NXC7H16:1'
  tabulated-quantities:
    mass-fractions:
      - OH
      - C2H2
  mixture-fraction:
    grid-points: 96
    grid-type: clustered
    grid-cluster-intensity: 6
    grid-cluster-point: 0.1
```

#### 2.10.6.7. Laminar, Adiabatic SLFM Library

To generate a strained laminar flamelet model, the only required change from the simpler examples is that the chemistry type be set to SLFM. However, we often prefer to specify the number and distribution of grid points in the mixture fraction and dissipation rate dimensions. For more information, run Spitfire in parse-only mode with -p to see an expanded input file.

```
mixture-fraction-model:
    chemistry-type: SLFM
    heat-loss-type: adiabatic
    turbulence-type: laminar
    cantera:
        xml: heptane_LiuEtAl.xml
    oxidizer:
        temperature: 300
        mass-fractions: 'N2:0.767,O2:0.233'
    fuel:
        temperature: 485
        mass-fractions: 'NXC7H16:1'
    mixture-fraction:
        grid-points: 96
    scalar-dissipation-rate:
        grid-points: 36
```

#### 2.10.6.8. Turbulent, Nonadiabatic SLFM Library

In this final demonstration we incorporate heat loss, radiation and soot source terms, and a presumed PDF model for turbulence-chemistry interaction. For more information on parameter settings, run Spitfire in parse-only mode with -p to see an expanded input file.

```
mixture-fraction-model:
    chemistry-type: SLFM
    heat-loss-type: nonadiabatic-defect
    turbulence-type: turbulent
    presumed-PDF:
        PDF-type: ClipGauss
        integrator-type: GaussKronrod
        scaled-scalar-variance:
            grid-points: 32
        mean-mixture-fraction:
            grid-points: 32
        mean-scalar-dissipation-rate:
            grid-points: 32
        mean-stoichiometric-enthalpy-defect:
            grid-points: 32
        minimize-convolution-time: False
    cantera:
        xml: heptane_LiuEtAl.xml
    oxidizer:
        temperature: 300
        mass-fractions: 'N2:0.767,O2:0.233'
    fuel:
        temperature: 485
        mass-fractions: 'NXC7H16:1'
    mixture-fraction:
        grid-points: 96
    scalar-dissipation-rate:
        grid-points: 36
        min: 1.e-3
        max: 1.e2
        reference: stoichiometric
    stoichiometric-enthalpy-defect:
        grid-points: 36
    tabulated-quantities:
        include-radiation: True
        soot-model: Aksit-Moss
```

# 2.11. TURBULENT REACTING MIXING MODELS

In a reacting flow it is sometimes necessary to limit the finite-rate chemistry reaction when it occurs in regions of high turbulence. In this case, turbulent mixing of the reactants limits the finite-rate (laminar) chemical reaction, the time scale of which can be calculated as

$$t_{\mathrm{c}} = N_{\mathrm{R}} \frac{\sum_{i=1}^{N_{\mathrm{S}}} Y_i}{\sum_{i=1}^{N_{\mathrm{S}}} \frac{\mathrm{dY_i}}{\mathrm{dt}}} \tag{2.388}$$

where $N_{\mathrm{R}}$ is the number of reactions, $N_{\mathrm{S}}$ is the number of species in the reaction, and $Y_i$ represents the concentration of species $i$. The denominator in Equation 2.388 is the production rate of species $i$.

## 2.11.1. Modified EDC Model (PARENTE)

One such model, named here as the PARENTE model, is adopted from [58] which derives explicit dependencies between the EDC model coefficients and the turbulent Reynolds ($\mathrm{Re_t}$) and Damköhler (Da) dimensionless numbers. This model is based on EDC and is used to modifty the chemistry reactions prescribed in the gas phase. The PARENTE turbulent reacting mixing model can can be used

in addition to or in lieu of the built-in EDC turbulent combustion model found in Section 2.9. The current model is described as

$$\mathrm{Da} = \max\left(\min\left(\frac{1}{t_\mathrm{c}}\sqrt{\frac{\nu}{\epsilon}}, 10\right), 1\times10^{-10}\right) \tag{2.389}$$

$$\mathrm{Re_t} = \frac{k^2}{\nu\epsilon} \tag{2.390}$$

$$C_\tau = \min\left(\frac{C_1}{\sqrt{\mathrm{Da}(\mathrm{Re_t}+1)}}, 2.1377\right) \tag{2.391}$$

$$C_\gamma = \max\left(\min\left(C_2\sqrt{\mathrm{Da}(\mathrm{Re_t}+1)}, 5.0\right), 0.4082\right) \tag{2.392}$$

$$\gamma_\mathrm{L} = C_\gamma\left(\frac{\nu\epsilon}{k^2}\right)^{\frac{1}{4}} \tag{2.393}$$

$$\tau^* = C_\tau\sqrt{\frac{\nu}{\epsilon}} \tag{2.394}$$

where $k$ represents the turbulent kinetic energy, $\epsilon$ is the turbulent dissipation and $\nu$ is the kinematic viscosity of the gas. The constants $C_1$ and $C_2$ can be specified by the user, but default to $C_1 = 0.05774$ and $C_2 = 0.5$. The modified reaction time scale, $\tau^*$ is used to solve the reaction equation from time $t^n$ to $t^{n+\tau^*}$. Following this calculation, the reaction source terms for species ($S^*_{Yi}$) and enthalpy ($S^*_H$) can be computed. These computed sources are further scaled via the multiplication of a coefficient $\kappa$ which is calculated as

$$\kappa = \begin{cases} 1, & \text{if } \gamma_\mathrm{L} \geq 1 \\ \max\left(\min\left(\frac{\gamma_\mathrm{L}^{e_1}}{1-\gamma_\mathrm{L}^{e_2}}, 1\right), 0\right), & \text{otherwise} \end{cases}$$

where $e_1 = e_2 = 3.0$ in our implementation of the model. Following the evaluation of $\kappa$, the final source terms for species and enthalpy can be calculated as

$$S_{Y_i} = \kappa S^*_{Y_i} \tag{2.395}$$

$$S_H = \kappa S^*_H \tag{2.396}$$

respectively.

## 2.12. SOOT GENERATION MODEL FOR MULTICOMPONENT COMBUSTION

Soot is an important contributor to radiative exchange within a fire and between a fire and its surroundings. Soot production, destruction and transport at flame scales are still active areas of research, with important chemical/physical processes not understood from a fundamental physics point of view. Basically, soot particles are carbon-rich solid particles generated in regions of excess pyrolyzate, such as on the rich side of a diffusion flame. Unagglomerated soot particles have characteristic dimensions in the range 0.01–0.05 $\mu$m (Zukoski [59]).

*The main purpose of the soot model is for the calculation of the absorption coefficient* in the radiant energy transfer equation. For the current implementation we employ the soot model implemented in the KAMELEON code because it has been used for large turbulent fire calculations with participating media radiation. The model is discussed in Magnussen et al. [41] and Magnussen and Hjertager [60] It is a two-step formulation, first described by Tesner et al. [61]. The model for generation and combustion of soot can be summarized by three principal steps: 1) particle nucleation, where the first solid soot particles (often called radical nuclei) are created as a result of fuel oxidation and pyrolysis, 2) particle growth, whereby the soot particle size increases due to the addition of material which is primarily carbon (10–20% mole fraction hydrogen) through a series of reactions and coagulation, 3) particle oxidation, where soot particles are burned. Additional information is provided in the overview by Haynes and Wagner [62].

Since the soot model is primarily directed at closing emission/absorption terms in the radiative transfer equation, engineering approximations are made with respect to its inclusion in the Navier Stokes equations. Specifically, heats of reaction associated with formation and destruction are not accounted for in the heat balance, and the mass concentrations of soot and radical nuclei are not included in the species mass balances; they are treated as tracers. The model has a significant amount of empiricism associated with it, necessitated by the extreme length scale range of soot processes, its complexity, and the degree to which many processes have yet to be quantified from a first principles perspective. The model choice can be considered to be a pragmatic one based on its prior use in fire calculations.

The present model has been constructed to fit into the same framework as the conceptual model for turbulent combustion outlined in the theory section for the EDC model. In the following subsections, the basic mechanisms of soot formation and destruction are presented. These processes occur on a scale smaller than can be resolved numerically, therefore the following subsections present the basic approach to the subgrid modeling of the elementary mechanisms, suitable for use in a numerical model.

### 2.12.1. EDC Soot Model

It is important to note that the processes of turbulent soot formation and combustion occur on a scale smaller than can be resolved in a numerical approximation. Thus, the averaged governing equations to be solved numerically must be supplemented with subgrid models to account for these subgrid processes. The conceptual model for subgrid turbulent soot generation and combustion is consistent with the two-zone, turbulent, gas-phase, combustion model presented in the last section (see also Holen [50]). One zone is the flame zone (flame structure) and the other is the surrounding zone.

Soot reactions tend to be slower than gas phase hydrocarbon chemistry. Therefore, the infinitely fast chemistry limit used for the gas phase chemistry is not employed for soot. The current model assumes that the formation and combustion rates are long compared to turbulent mixing rates at flame scales. A steady-state, steady-flow assumption is used in the formulation between the production/destruction rates and the turbulent mixing rates to obtain the soot mass fraction in the flame zone in an algebraic manner (avoiding solution of stiff ordinary differential rate equations).

#### 2.12.1.1. Criteria for Soot and Radical Nuclei Formation

To start, the first level criteria for formation of soot are

$$Y_{prod} > Y_{lim} \qquad \text{and} \qquad \gamma\chi > 0 \qquad \text{and} \qquad T^{\circ} > T_{lim}, \tag{2.397}$$

where $Y_{prod}$ denotes the mass fraction of products, $Y_{lim}$ and $T_{lim}$ are minimum values of product mass fraction and surrounding temperatures allowing soot generation, and $\gamma\chi$ is the volume fraction of the reaction zone of the current cell. If these conditions are met, then the first step is to determine how much carbon is available over and above what may potentially react with oxygen to produce $CO_2$, via the 2-step reaction postulated in the chemistry model (see Section 2.9). So, first form the elemental mass fraction of excess (over what may potentially form $CO_2$) carbon in each species,

$$f_{c,i} = \max\left[0, \left(\hat{Y}_i^C - \frac{1}{2}\frac{W_C}{W_O}\hat{Y}_i^O\right)\right], \tag{2.398}$$

where $\hat{Y}_i^C$ is the mass fraction of carbon in species $i$, and $\hat{Y}_i^O$ the mass fraction of elemental oxygen in species $i$. For example, for CO (carbon monoxide), $\hat{Y}_{CO}^C = 12/(12+16)$, etc. Also, for $CO_2$, the excess fraction $f_{c,co2} = 0$, while for any species containing oxygen but no carbon, the formula for the excess fraction is constructed to give zero. Hence, the fraction is non-zero only for species containing carbon but excluding carbon dioxide; i.e., the fuel and carbon monoxide species will have non-zero excess carbon fraction. With the 2-step reaction process being considered, the CO can be considered a fuel in the second reaction, in which CO and $H_2$ are oxidized if enough oxygen in available after the first reaction. Thus, *the computed carbon fraction, $f_{c,i}$, is collectively the available carbon in the "fuel species"*, comprised of the actual CHNO fuel and CO, and will be zero for other species (compounds). Note that this fraction excludes the carbon in the species that can potentially form $CO_2$ via *oxidation with the oxygen present in the species itself.*

Now, the mass fraction of carbon potentially available to produce soot can be computed for the surrounding and flame zones from the following,

$$Y_{c\to s}^{\circ} = \sum_i f_{c,i} Y_i^{\circ} \qquad Y_{c\to s}^{*} = \sum_i f_{c,i} Y_i^{*}. \tag{2.399}$$

Again, these mass fractions represent the potentially available carbon in the fuels, separated into flame zone and surroundings, for formation of soot. The average mass fraction of soot-producing-carbon is,

$$Y_{c\to s} = (\gamma\chi)\, Y_{c\to s}^{*} + (1-\gamma\chi)\, Y_{c\to s}^{\circ}. \tag{2.400}$$

Now we must compare the amount of oxidant (not counting oxidant present in the fuel compound) actually available for burning these fuels to produce $CO_2$; any excess carbon is available to produce additional soot and radical nuclei. The amount of oxygen required to react with *all* of the available soot-producing-carbon ( $Y_{c\to s}$, which already excludes the oxygen present in the fuel compound) to produce $CO_2$ is

$$Y_{O_2,max} = 2\frac{W_O}{W_C} Y_{c\to s}. \tag{2.401}$$

Now, if we can compare this to how much oxygen is actually available, we can decide how much excess carbon is available to produce soot and radical nuclei. Thus the fraction (molar ratio) of excess carbon

for producing soot is determined by subtracting off the amount that will go to stoichiometrically react with the available oxygen to ultimately produce $CO_2$ in the two-step reaction,

$$\xi_c = \frac{Y_{c\to s}/W_c - Y_{O_2}/W_{O_2}}{Y_{c\to s}/W_c} = 1 - \frac{Y_{O_2}}{Y_{O_2,max}} \Rightarrow 1 - \min\left(1, \frac{Y_{O_2}}{Y_{O_2,max}}\right), \tag{2.402}$$

where the last expression is the computational implementation, to take care of "lean" conditions where there is excess oxidizer, and which will result in zero mole fraction of carbon to produce soot.

In other words, it is assumed that for a given fraction of existing soot that gets mixed by turbulence into a flame zone, a fraction $\xi_c$, will contribute to the growth of soot in the flame zone, while the balance, $(1 - \xi_c)$ will be consumed in the production of $CO_2$. Implicit in this assumption is that soot entering a flame will be consumed in proportion to the oxygen present. Therefore in fuel lean regions, soot entering flame zones will be preferentially destroyed.

Now we are in a position to determine whether soot and radical nuclei can be formed under present conditions. They will form if

$$Y_{c\to s} > Y_{soot} \qquad X_{c,soot} > 0. \tag{2.403}$$

The first inequality in Equation 2.403 asserts that the available potential-soot-producing carbon in the fuel must exceed the present amount of soot before enabling generation of additional soot. The construction of $Y_{c\to s}$ sums the total potential soot-producing-carbon, without distinguishing whether the carbon exists as soot or fuel. The second requires enough carbon to exceed the requirements for the combustion reaction; i.e., soot will only be formed under fuel rich conditions.

#### 2.12.1.2. Soot Formation and Termination Models

In general, soot may be considered to be generated in both the reaction zone and in the surrounding zone. This was the assumption invoked in KAMELEON-II (Holen, et al. [50]). As we shall see, in the present implementation for multicomponent species problems, formation/destruction is assumed to take place only in the surrounding fluid. The mass fraction of fuel in the reaction zone is assumed to be proportional to the mass fraction $\gamma^*$, and the reacting fraction of the fuel in the reaction zone, $\chi$. The total rate of radical nuclei formation and destruction is given by a volume averaged sum of the formation within the reaction zone and the surrounding zone.

Assuming the conditions in Equation 2.403 are met, the rates of formation can be computed. The following models for soot formation and termination were originally described by Tesner et al. [61] and have been subsequently modified by Magnussen and co-workers. The elementary mechanisms (subgrid models for the fire code application) of formation and destruction of radical nuclei was described by Tesner et al. [61] in the form,

$$\dot{R}_n = n_0 + (f - g)\, n - g_0 N n \qquad \left[\text{particles/s} - \text{m}^3\right], \tag{2.404}$$

where $n_0$ is the spontaneous origination rate of radical nuclei in particles/(s-$\text{m}^3$) (due to fuel oxidation and fuel pyrolysis), $f$ is the linear branching coefficient (whereby radical nuclei react to create additional radical nuclei), $g$ is the linear termination coefficient (where radical nuclei combine with existing radical nuclei), $n$ is the concentration of radical nuclei in particles/$\text{m}^3$, $g_0$ is the linear coefficient of termination

on soot particles (where radical nuclei combine with existing soot particles), and $N$ is the particle concentration of soot particles (assumed to be spherical with uniform diameter $d_p$) in particles/$\mathrm{m}^3$. The spontaneous origination rate of radical nuclei was given by Tesner as

$$n_0 = 1.08 a_0 \rho Y_{fuel} \exp\left(-\frac{E}{RT}\right). \tag{2.405}$$

The rate of soot particle formation and destruction was given by Tesner et al. as,

$$\dot{R}_{N,f} = (a - bN)\, n \qquad \left[\text{particles/s} - \text{m}^3\right]. \tag{2.406}$$

The parameters appearing in the foregoing, as determined[2] by Tesner et al. [61] and Holen, et al. [50], are given in Table 2.12-1 Tesner et al. [63] provide additional data for various hydrocarbons.

**Table 2.12-1.. Soot model parameters (Tesner et al.(1971); Holen, et al.(1994))**

| $a$ | $f-g$ | $g_0$ | $b$ | $E/R$ | $\rho_{soot}$ | $a_0$ | $d_p$ |
|---|---|---|---|---|---|---|---|
| $[1/\text{s}]$ | $[1/\text{s}]$ | $[\text{cm}^3/\text{part}-\text{s}]$ | $[\text{cm}^3/\text{part}-\text{s}]$ | $[\text{K}]$ | $[\text{g/cm}^3]$ | $[\text{part/g}-\text{s}]$ | $[\text{cm}]$ |
| $10^5$ | $10^2$ | $10^{-9}$ | $8\times10^{-8}$ | $9\times10^4$ | 2.0 | $12.5\times10^{33}$ | $17.85\times10^{-7}$ |

The elementary formation/destruction models of Tesner have been modified by Magnussen et al. (Holen, et al. [50]) for application to multicomponent fire simulation problems. First, for implementation into a computer program, transport equations for two field variables, radical nuclei and soot concentrations, are needed. For computational reasons, it is convenient to write all transport equations in a standard form,

$$\int \frac{\partial \rho\phi}{\partial t}\mathrm{d}V + \int \rho u_j \phi n_j \mathrm{d}S = \int \Lambda \frac{\partial \phi}{\partial x_j} n_j \mathrm{d}S + \rho S_\phi, \tag{2.407}$$

written for the arbitrary scalar field, $\phi$, which will have units of intensity per unit mass (or be dimensionless, such as a mass fraction). Thus the computational variables for the soot model are, respectively, the radical nuclei concentration and soot mass fraction,

$$\beta = \frac{n}{\rho} \qquad \text{and} \qquad Y_{soot} = \frac{C_{soot}}{\rho} \tag{2.408}$$

where $C_{soot}$ denotes the mass concentration of soot (kg/$\mathrm{m}^3$). In terms of these variables, the spontaneous origination of radical nuclei, as modified by Magnussen et al., is determined from,

$$\beta_0^\circ = 1.08 a_0 \left(Y_{c\to s}^\circ - Y_{soot}\right) \exp\left(-\frac{E}{RT^\circ}\right), \tag{2.409}$$

$$\beta_0^* = 1.08 a_0 \left(Y_{c\to s}^* - Y_{soot}\right) \exp\left(-\frac{E}{RT^*}\right), \tag{2.410}$$

[2] In practice, the variables $a$ and $b$ are scaled (multiplied) by $10^{16}$ while $a_0$ is scaled (divided) by $10^{16}$ thereby effectively reducing the nuclei concentration by this amount.

in units of part/kg-sec, which, when compared to Tesner's form, is seen to have been written in terms of the excess soot-producing carbon, rather than simply being proportional to the fuel concentration, of which only a fraction is available to produce radical nuclei and soot. Similarly, the linear branching and termination reactions for radical nuclei can be written in the form,

$$\frac{\dot{R}^{\circ}_{n,f-g,mod}}{\rho^{\circ}} = \max\left(0, f_c^{\circ}\right)(f-g)\,\beta^{\circ} \equiv \max\left(0, f_c^{\circ}\right)\frac{\dot{R}^{\circ}_{n,f-g}}{\rho^{\circ}}, \tag{2.411}$$

$$\frac{\dot{R}^{*}_{n,f-g,mod}}{\rho^{*}} = \max\left(0, f_c^{*}\right)(f-g)\,\beta^{*} \equiv \max\left(0, f_c^{*}\right)\frac{\dot{R}^{*}_{n,f-g}}{\rho^{*}}, \tag{2.412}$$

where the scale factors are defined by,

$$f_c^{\circ} = \frac{\left(Y^{\circ}_{c\to s} - Y_{soot}\right)}{Y^{\circ}_{c\to s}} \qquad \text{and} \qquad f_c^{*} = \frac{\left(Y^{*}_{c\to s} - Y_{soot}\right)}{Y^{*}_{c\to s}}, \tag{2.413}$$

and represent the fraction of soot-producing carbon available in the surroundings and flame zone, respectively. The present formulation reduces the rates by the fraction of soot-producing carbon over and above that which is already present as soot, represented by the last terms in each equation. In contrast, the bilinear termination term for generation of soot is indirectly modified through the soot mass fraction, which is similarly modified (as will be shown shortly). Therefore, the termination term can simply be expressed in terms of the computational variables as,

$$\frac{\dot{R}^{*}_{n,g_0}}{\rho^{*}} = g_0\frac{\rho^{*}Y^{*}_{soot}}{m_p}\beta^{*} \qquad \text{and} \qquad \frac{\dot{R}^{\circ}_{n,g_0}}{\rho^{\circ}} = g_0\frac{\rho^{\circ}Y^{\circ}_{soot}}{m_p}\beta^{\circ}, \tag{2.414}$$

in which the soot particle concentration has been expressed in terms of the soot mass fraction and an average mass of a soot particle, $m_p$ (kg),

$$N = \frac{\left(\rho Y_{soot}\right)}{m_p} \qquad \text{and} \qquad m_p = \rho_{soot}\frac{4}{3}\pi\left(\frac{d_p}{2}\right)^3 \tag{2.415}$$

$$m_p^{\circ} = \frac{b}{a}\rho^{\circ}Y^{\circ}_{c\to s}, \tag{2.416}$$

$$m_p^{*} = \frac{b}{a}\rho^{*}Y^{*}_{c\to s}, \tag{2.417}$$

See Table 2.12-1 for data used in these equations. The generation/destruction term for soot are also modified via the scale factors,

$$\frac{\dot{R}^{*}_{soot,form,mod}}{\rho^{*}} = f_c^{*}m_p\left(a - b\frac{\left(\rho^{*}Y^{*}_{soot}\right)}{m_p}\right)\beta^{*} \equiv f_c^{*}m_p\frac{\dot{R}^{*}_{soot,form}}{\rho^{*}}, \tag{2.418}$$

$$\frac{\dot{R}^{\circ}_{soot,form,mod}}{\rho^{\circ}} = f_c^{\circ}m_p\left(a - b\frac{\left(\rho^{\circ}Y^{\circ}_{soot}\right)}{m_p}\right)\beta^{\circ} \equiv f_c^{\circ}m_p\frac{\dot{R}^{\circ}_{soot,form}}{\rho^{\circ}}. \tag{2.419}$$

to be used in the elementary source expression for the flame zone and surroundings.

The production/destruction of soot in the reaction zone should approach zero for $Y^*_{soot} \to Y^*_{c\to s}$, since production should cease when the amount of soot equals the maximum available soot-producing-carbon in the reaction zone. This is easier to see by substituting this form into the production term,

$$\frac{\dot{R}^*_{soot,form,mod}}{\rho^*} = f^*_c b \left(\rho^* Y^*_{c\to s} - \rho^* Y^*_{soot}\right) \beta^*. \tag{2.420}$$

This term vanishes when the soot mass fraction equals the maximum carbon mass fraction, by virtue of its construction. However, this form is clearly not the form suggested by Tesner [61], the scaling factor notwithstanding.

#### 2.12.1.3. Soot Combustion Model

The soot combustion model assumes that soot is destroyed in the flame zone based on two factors 1) the rate at which it is mixed into the flame zone, and 2) that there is sufficient oxygen to consume it. The mixing rate is the same as in Equation 2.285 (in the gas phase combustion model section) where the species $Y_k$ are treated as follows: In the cell, the fraction of soot that will burn up in the flame zone is $(1-\xi_c)Y_{soot}$. In the flame zone, this mass is converted to $CO_2$, so its mass fraction in the flame zone is zero. The radical nuclei concentration is treated similarly. Therefore,

$$\frac{\dot{R}_{n,comb}}{\bar{\rho}} = \left(\frac{-\left(1-\xi_c\right)\frac{n}{\rho}}{\tau_{res}}\right)\left(\frac{\gamma\chi}{1-\gamma\chi}\right)\chi_3, \tag{2.421}$$

$$\frac{\dot{R}_{soot,comb}}{\bar{\rho}} = \left(\frac{-\left(1-\xi_c\right)Y_{soot}}{\tau_{res}}\right)\left(\frac{\gamma\chi}{1-\gamma\chi}\right)\chi_3. \tag{2.422}$$

It is convenient to define a new timescale,

$$\tau_h = \frac{\left(1-\gamma\chi\right)\tau_{res}}{\chi_3}. \tag{2.423}$$

#### 2.12.1.4. Calculating Properties of the Reaction Zone

The foregoing models for soot and radical nuclei contain properties corresponding to the flame zone and surroundings. This section discusses the method employed by Magnussen et al. to compute these properties. The flame zone properties are computed by assuming local equilibrium mass transfer due to turbulent mixing between the reaction zone and surroundings. In other words, the production and combustion rates are sufficiently slow that the mass concentrations in the flame zone come to an equilibrium state with the surroundings via the turbulent mixing rate. This equilibrium rate is assumed to instantaneously adjust to the new cell conditions at every time step.

For this steady-state, steady flow approximation, a balance equation can be written for both nucleate particles and soot mass fraction for the flame zone. In words, the radical nuclei concentration (or soot mass fraction) mixed into the flame zone minus the radical nuclei concentration (or soot mass fraction) mixed out of the flame zone plus the production of radical nuclei (or soot) minus the combustion of

radical nuclei (or soot) equals zero. Note that the combustion rates given above are equal to the mixing rates times the fraction of radical nuclei concentration (or soot mass fraction) able to combustion. So the difference in these terms is equal to the soot production rates or,

$$\frac{(\beta^* - \xi_c \beta)}{\tau_h} = \beta_0^* + \frac{f_c^* \dot{R}_{n,f-g}^*}{\rho^*} - \frac{\dot{R}_{n,g_0}^*}{\rho^*}, \tag{2.424}$$

$$\frac{(Y_{soot}^* - \xi_c Y_{soot})}{\tau_h} = f_c^* m_p^* \frac{\dot{R}_{soot,form}^*}{\rho^*}. \tag{2.425}$$

Solution of these two algebraic equations with two unknowns gives, $\beta^*$ and $Y_{soot}^*$, the radical nuclei and soot concentrations in the flame zone, respectively. Note that the formation/destruction terms are of a bilinear form in the soot and radical nuclei concentrations. Thus, to compute the flame zone values of radical nuclei and soot mass fractions requires the simultaneous solution of this $2 \times 2$ system of equations. In particular, substituting for these terms from the formula given above, Equation 2.425 can be solved for $Y_{soot}^*$ using Equation 2.420. The result is that the mass fraction of soot in the flame zone in terms of the radical nuclei concentration.

$$Y_{soot}^* = \frac{\xi_c Y_{soot} + \tau_h f_c^* b \rho^* Y_{c \to s}^* \beta^*}{1 + \tau_h f_c^* b \rho^* \beta^*}. \tag{2.426}$$

Equation 2.426 can be used in Equation 2.424 to form a quadratic equation for $\beta^*$,

$$tildea_s \left(\beta^*\right)^2 + \tilde{b}_s \beta^* + tildec_s = 0, \tag{2.427}$$

where,

$$\tilde{a}_s = f_c^* \tau_h \rho^* \left(\tilde{\alpha} b + \tau_h a g_0\right), \tag{2.428}$$

$$\tilde{b}_s = \tilde{\alpha} + \tau_h \rho^* \left( \frac{\xi_c Y_{soot} g_0}{\frac{b}{a} \rho^* Y_{c \to s}^*} - f_c^* b A \right), \tag{2.429}$$

$$\tilde{c}_s = -A, \tag{2.430}$$

$$A = \xi_c \beta + \tau_h \beta_0^*, \tag{2.431}$$

$$\tilde{\alpha} = 1 - \tau_h \left(f - g\right) f_c^*. \tag{2.432}$$

The solution is the negative root of the quadratic, here written in a computationally appropriate form,

$$\beta^* = \frac{-2\tilde{c}_s}{\tilde{b}_s + \sqrt{\tilde{b}_s^2 - 4\tilde{a}_s \tilde{c}_s}} \tag{2.433}$$

In the limit where

$$Y_{soot}^* \to Y_{c \to s}^* \tag{2.434}$$

then the soot mass fraction becomes static and the radical nuclei concentration can be solved for directly. The result is

$$\beta^* = \frac{\xi_c \beta + \tau_h \beta_o^*}{1 + \tau_h \left(-f_c^* \left(f - g\right) + \frac{a g_o}{b}\right)}. \tag{2.435}$$

#### 2.12.1.5. Calculating Properties of the Surroundings

Having computed the properties of the reaction zone, the properties for the surroundings are calculated from the definition of the cell (average) values,

$$\beta^{\circ} = \frac{\beta - \gamma\chi\beta^{*}}{1 - \gamma\chi}, \tag{2.436}$$

$$Y^{\circ}_{soot} = \frac{Y_{soot} - \gamma\chi Y^{*}_{soot}}{1 - \gamma\chi}. \tag{2.437}$$

$$\beta^{\circ} = min\left(\beta^{\circ}, \frac{\frac{ag_o}{b} \times 10^{-6}}{\rho^{\circ}}\right) \tag{2.438}$$

Note that there is an upper bound to the number of nucleate particles based on a 50 percent dense mixture given they are monodisperse at the size given in Table 2.12-1 with mass given by

$$m = \rho_{soot}\frac{4}{3}\pi\left(\frac{d_p}{2}\right)^3. \tag{2.439}$$

Now we are in a position to specify the transport equations and source terms for the soot model.

### 2.12.2. Transport Equations and Source Terms

Two transport equations for radical nuclei and soot mass fractions need be solved,

$$\int \frac{\partial\rho\beta}{\partial t}\mathrm{d}V + \int \rho\beta u_j n_j \mathrm{d}S = \int \frac{\mu_{eff}}{\sigma_Y}\frac{\partial\beta}{\partial x_j}n_j\mathrm{d}S + \int \rho S_n \mathrm{d}V, \tag{2.440}$$

$$\int \frac{\partial\rho Y_{soot}}{\partial t}\mathrm{d}V + \int \rho Y_{soot} u_j n_j \mathrm{d}S = \int \frac{\mu_{eff}}{\sigma_Y}\frac{\partial Y_{soot}}{\partial x_j}n_j\mathrm{d}S + \int \rho S_{soot}\mathrm{d}V. \tag{2.441}$$

In general, the source term, in particles/kg-sec, for radical nuclei is given by,

$$S_n = \gamma\chi\left(\frac{\dot{R}^{*}_{n,form,mod}}{\rho^{*}} - \frac{\dot{R}^{*}_{n,comb}}{\rho^{*}}\right) + (1 - \gamma\chi)\frac{\dot{R}^{\circ}_{n,form,mod}}{\rho^{\circ}} \tag{2.442}$$

where the form of the net formation/destruction source terms is,

$$\frac{\dot{R}^{\circ}_{n,form,mod}}{\rho} = \beta^{\circ}_0 + (f - g)\,\beta^{\circ}\max\left(0, f^{\circ}_c\right) - g_0\frac{aY^{\circ}_{soot}}{bY^{\circ}_{c\to s}}\beta^{\circ}. \tag{2.443}$$

For each of the reaction and surrounding zones, the (production Âŋdestruction) of radical nuclei in the flame zone is given by the mixing balance, or

$$\left(\frac{\dot{R}^{*}_{n,form}}{\rho} - \frac{\dot{R}^{*}_{n,comb}}{\rho}\right) = \left(\frac{\beta^{*} - \beta}{\tau_h}\right). \tag{2.444}$$

Substituting gives,

$$S_n = \gamma\chi\left(\frac{\beta^* - \beta}{\tau_h}\right) + (1-\gamma\chi)\left(\beta_0^\circ + (f-g)\,\beta^\circ \max\left(0, f_c^\circ\right) - g_0 \frac{aY_{soot}^\circ}{bY_{c\to s}^\circ}\beta^\circ\right), \tag{2.445}$$

The general source term for soot (1/sec) is given by

$$S_{soot} = m_p\gamma\chi\left(\frac{\dot{R}^*_{soot,form,mod}}{\rho^*} - \frac{\dot{R}^*_{soot,comb}}{\rho^*}\right) + (1-\gamma\chi)\, m_p \frac{\dot{R}^\circ_{soot,form,mod}}{\rho^\circ} \tag{2.446}$$

The (production-destruction) of soot in the flame zone is likewise given by the mixing balance, or

$$m_p\left(\frac{\dot{R}^*_{soot,form}}{\rho} - \frac{\dot{R}^*_{soot,comb}}{\rho}\right) = \left(\frac{Y^*_{soot} - Y_{soot}}{\tau_h}\right). \tag{2.447}$$

Substituting gives,

$$S_{soot} = \gamma\chi\left(\frac{Y^*_{soot} - Y_{soot}}{\tau_h}\right) + (1-\gamma\chi)\, f_c^\circ b\rho^\circ \left(Y^\circ_{c\to s} - Y^\circ_{soot}\right)\beta^\circ, \tag{2.448}$$

which also follows the practice of using the scale factor and effective mass for a soot particle in the surroundings, $m_p = b\rho^\circ Y^\circ_{c\to s}/a$.

The fact that the soot and radical nuclei concentrations are treated as tracers should be reemphasized. This means that their concentrations in the gas mixture are assumed insignificant such that they do not enter into calculations of density, or other properties of the mixture.

## 2.13. ABSORPTIVITY MODEL

The absorption coefficient submodel calculates a spectrally averaged total absorptivity value for a homogeneous ( in thermodynamic state and composition ) mixture of gaseous $CO_2$, $H_2O$, and soot particles. It should be recognized that this model does not account for either the presence of volatilized hydrocarbon molecules nor for the spectral line broadening effects of $N_2$ gas. The following implicit assumptions are made:

1. Thermodynamic equilibrium between soot and gas phase.
2. Homogeneous mixture over length scale of interest ( cf. input 1 )
3. Individual ( non agglomerated ) spherical soot particles with diameter much smaller than the radiation wavelength (Rayleigh scattering).
4. Absorptivity of the soot varies inversely with radiation wavelength.

The following quantities are required:

1. Length scale indicating the optical path length of interest, $L_{cell}$ in centimeters.

2. Mixture temperature, T, in Kelvin.
3. Total mixture pressure, $p_{mix}$, in bar.
4. Partial pressures of the $CO_2$ and $H_2O$ gaseous components, $p_{co2}$, $p_{h2o}$, in bar.
5. Soot volume fraction, $X_{soot}$.

The absorptivity model generates the following output:

- Spectrally averaged absorptivity, $\alpha$, in $\mathrm{cm}^{-1}$.

The absorptivity is based on empirical correlations for the total emittance of a homogeneous, isothermal mixture with a given optical path length. The correlations used in this model are based on empirical data covering a range of optical path lengths, temperatures, soot concentrations and pressures:

- $1\ \mathrm{cm} \leq L_{cell} \leq 10^3\ \mathrm{cm}$
- $600K \leq T \leq 2400K$
- $10^{-8} \leq X_{soot} \leq 10^{-5}$
- $0.1\ \mathrm{bar} \leq p_{co2},\ p_{h2o} \leq 1\ \mathrm{bar}$

The absorptivity values provided by the equations in this model are accurate to within 10% - 30% of their value with greater accuracy at higher temperatures, path lengths, and concentrations.

## 2.13.1. Theory

The total ( e.g. integrated over all wavelengths ) absorptivity of a homogeneous ( in composition and temperature ) thickness $L_{cell}$ layer of $CO_2$ gas, $H_2O$ gas, and soot particles may be expressed in terms of the total emittance of the layer

$$\alpha = -\frac{1}{L_{cell}} \log\left(1 - \kappa\right), \tag{2.449}$$

where $\alpha$ is the total absorptivity and $\kappa$ is the total emittance. The total emittance of the mixture may be expressed in terms of the total emittance of the soot and gas phase (Siegel and Howell [10], Eq. (13-145)),

$$\kappa = \kappa_{soot} + \kappa_{gas} - \kappa_{soot}\kappa_{gas}, \tag{2.450}$$

where $\kappa_{soot}$ and $\kappa_{gas}$ are the total emittance of the soot and gas phase respectively as if the other phase were not present.

To evaluate the absorptivity within a given control volume, the layer length, $L_{cell}$, is taken to be the geometric path length through the cell. This assumption ( cf. assumption 2 ) implies that the mixture composition and temperature are uniform within the given cell. For convenience, the hydraulic diameter may be used for the layer thickness (in three dimensions),

$$L_{cell} = 2\left[\frac{3}{4}\frac{V}{\pi}\right]^{1/3}, \tag{2.451}$$

where $V$ is the cell volume. Alternatively, Tezduyar [64] proposes a more expensive length scale for finite element grids,

$$L_{cell} = 2\hat{s} \cdot \left( \sum_{i=1}^{n_e} \nabla \phi_i \right), \tag{2.452}$$

where, $L_{cell}$ is the path length through the element in direction $\hat{s}$, and $\phi_i$ is the finite element basis.

## 2.13.2. Emittance Model

The KAMELEON fire code ( Holen, et al. [50] ) employs the work of Felske and Tien [65] to provide the emittance of a mixture of $CO_2$, $H_2O$, and soot particles. Assuming the absorptivity of the soot phase varies inversely with wavelength (Rayleigh scattering theory), a closed form expression may be obtained for the total emittance of the soot phase,

$$\kappa_{soot} = 1 - \frac{15}{\pi^4} \Psi^{(3)} \left[ 1 + \frac{c X_{soot} T L_{cell}}{C_2} \right], \tag{2.453}$$

where, $X_{soot}$ is the soot volume fraction, $T$ is the temperature, $C_2 = 0.01438769$ m-K is the second Planck constant, and $c = 7.0$ ( Felske and Charalampopoulos [66] suggest $c = 5.0$ ). The pentagamma function $\Psi^{(3)}(x)$ is given by Abramowitz and Stegun [67],

$$\Psi^{(n)}(z) = \frac{\mathrm{d}^{n+1}}{\mathrm{d}z^{n+1}} \log\left[\Gamma(z)\right] = (-1)^{n+1} \int_0^\infty \frac{t^n e^{-zt}}{1 - e^{-t}} \mathrm{d}t, \qquad n = 1, 2, 3, \ldots \tag{2.454}$$

Equation 2.454 may be evaluated by the series expansion (Abramowitz and Stegun [67]),

$$\Psi^{(3)}(z) = 6 \sum_{k=0}^{\infty} \frac{1}{(z+k)^4}, \tag{2.455}$$

and by the seven-term asymptotic expansion,

$$\Psi^{(3)}(z) = \frac{2}{z^3} + \frac{3}{z^4} + \frac{2}{z^5} - \frac{1}{z^7} + \frac{4}{3z^9} - \frac{3}{z^{11}} + \frac{10}{z^{13}} + \ldots \tag{2.456}$$

Equation 2.456 is accurate to within 1% of the value given by Equation 2.455 for $z > 1.6$ and accurate to within 0.1% of the value given by Equation 2.455 for $z > 2$. A plot of the pentagamma function and the asymptotic expansion are provided in Figure 2.13-1 for reference.

The emittance of the gas phase is given by Leckner [68]. Leckner's model is relatively involved and assumes that the path length, $L_{cell}$, is given in centimeters, the temperature, $T$, is given in Kelvin, and the pressure, $p$, is given in bars. Leckner also defines a reference temperature, $T_\circ = 273$ K, and pressure, $p_\circ = 1$ bar, for reduction purposes. Two additional quantities used by Leckner are the scaled temperature, $\theta = T/1000\mathrm{K}$ and the logarithm of the optical path length, $\lambda_\nu = \log_{10}(p_\nu L_{cell})$ where the subscript $\nu$ represents one of the species $CO_2$ or $H_2O$. These quantities are summarized in Table 2.13-1.

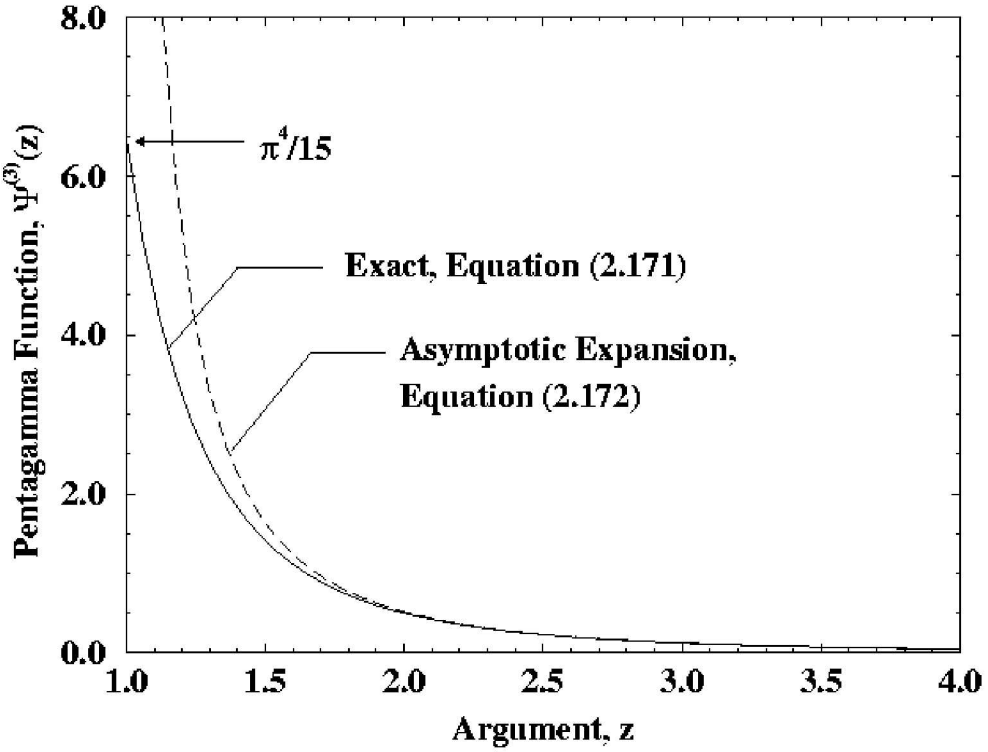


**Figure 2.13-1.. Pentagamma function and asymptotic expansion**

The emittance of the gas phase (cf. Equation 2.450) is the sum of the $CO_2$ and $H_2O$ contributions less a correction factor which accounts for overlap in the $CO_2$ and $H_2O$ absorption bands,

$$\kappa_{gas} = \kappa_{h2o} + \kappa_{co2} - \Delta\kappa, \tag{2.457}$$

where the species emittance at a given partial pressure and temperature is expressed in terms of a scale emittance, $\kappa_{\nu,\circ}$.

$$\frac{\kappa_\nu}{\kappa_{\nu,\circ}} = \exp\left(-\xi\left(\lambda_{max} - \lambda_\nu\right)^2\right)\left(\frac{AP_E + B}{P_E + A + B - 1} - 1\right) + 1 \tag{2.458}$$

Table 2.13-2 summarizes the quantities on the right hand side of Equation 2.458. The scale emittance, $\kappa_{\nu,\circ}$, for both species is given by the expressions

$$\log\left(\kappa_{\nu,\circ}\right) = a_0 + \sum_{i=1}^{M} a_i \lambda_\nu^i, \tag{2.459}$$

**Table 2.13-1.. Parameters used in Leckner's gas phase emittance model.**

| Quantity | Definition |
|---|---|
| Temperature units, $[T]$ | Kelvin |
| Path length units, $[L_{cell}]$ | centimeters |
| Pressure units, $[p]$ | bar |
| Reference temperature, $T_\circ$ | 273 K |
| Reference pressure, $p_\circ$ | 1 bar |
| Scaled path length, $\lambda_\nu$ | $\log_{10}(p_\nu L_{cell})$ |
| Scaled temperature, $\theta$ | T/1000K |

$$a_i = c_{i0} + \sum_{j=1}^{N} c_{ij}\theta^j, \tag{2.460}$$

where the coefficients $a_i$ and $c_{ij}$ are given in Table 2.13-3 and Table 2.13-4 for $CO_2$ and $H_2O$ respectively. (Leckner provides several alternative listings for the coefficients for calculating the total emittance of $CO_2$. The values listed in Table 2.13-3 are the values employed in the KAMELEON-II-FIRE program (1994).)

The effect of the overlap correction factor in Equation 2.457 is relatively small so Leckner [68] employed an approximate expression obtained from emittance data for a total pressure of 1 bar and temperatures between 1000K and 2200K:

$$\Delta\kappa = \left(\frac{\zeta}{10.7 + 101\zeta} - 0.0089\zeta^{10.4}\right)\left(\log_{10}\left[\left(p_{co2} + p_{h2o}\right) L_{cell}\right]\right)^{2.76}, \tag{2.461}$$

where,

$$\zeta = \frac{p_{h2o}}{p_{h2o} + p_{co2}}. \tag{2.462}$$

The following observations are made to clarify the range of applicability of the absorptivity submodel specifically for hydrocarbon combustion applications. The absorptivity model does not account for the presence of volatilized hydrocarbon molecules which may have strong absorption bands in the infrared region. The VULCAN/KAMELEON fire code (Holen, et al. [50]) accounts for the presence hydrocarbon molecules by treating hydrocarbon molecules in the same manner as the $CO_2$ and $H_2O$ product species ( cf. the partial pressure submodel ). This is a convenient although questionable assumption which provides for a zeroth order treatment of absorption by hydrocarbon molecules.

# 2.14. FUEL BOUNDARY CONDITION SUBMODEL

In most cases, fires are the result of burning fuel vapor in air. Exceptions include oxygenated and energetic materials that embody both fuel and oxidizer. The source of fuel vapor may be a gas release, the vapor which forms over a liquid surface due to its vapor pressure, liquid fuel which is heated above its vaporization temperature, or solid materials which are heated to the point where combustible gases are released due to pyrolysis reactions. The purpose of this submodel is to provide the mass flux and

**Table 2.13-2.. Species-specific parameters used in Equation 2.458.**

| Quantity | $CO_2$ | $H_2O$ |
|---|---|---|
| Equivalent pressure, $P_E$ | $P_E = p_{mix}\left(1 + 0.28\frac{p_{co2}}{p_{mix}}\right)$ | $P_E = p_{mix}\left(1 + 1.49\frac{p_{co2}}{p_{mix}}\sqrt{\frac{T_\circ}{T}}\right)$ |
| Maxima location, $\lambda_{max}$ | for $T > 700K$<br>$\lambda_{max} = \log_{10}(0.225\theta^2)$<br><br>for $T < 700K$<br>$\lambda_{max} = \log_{10}(0.054\theta^{-2})$ | $\lambda_{max} = \log_{10}(13.2\theta^2)$ |
| Coefficient, $\xi$ | $\xi = 1.47$ | $\xi = 0.5$ |
| Coefficient, $A$ | $A = 1.0 + 0.1\theta^{-1.45}$ | $A = 1.888 - 2.053\log_{10}\theta$<br>$\theta = 2.145$ if $T < 750K$ |
| Coefficient, $B$ | $B = 0.23$ | $B = 1.1\theta^{-1.4}$ |

**Table 2.13-3.. Coefficients Cij for calculating the scale total emittance of $CO_2$ from Equation 2.459 and Equation 2.460, (valid for T > 400K).**

| i | j (N=4) | | | | |
|---|---|---|---|---|---|
| (M=3) | 0 | 1 | 2 | 3 | 4 |
| 0 | -3.9781 | 2.7353 | -1.9882 | 0.31054 | 0.015719 |
| 1 | 1.9326 | -3.5932 | 3.7247 | -1.4535 | 0.20132 |
| 2 | -0.35366 | 0.61766 | -0.84207 | 0.39859 | -0.063356 |
| 3 | -0.080181 | 0.31466 | -0.19973 | 0.046532 | -0.0033086 |

**Table 2.13-4.. Coefficients Cij for calculating the scale total emittance of $H_2O$ from Equation 2.459 and Equation 2.460, (valid for T > 400K).**

| i | j (N=2) | | |
|---|---|---|---|
| (M=2) | 0 | 1 | 2 |
| 0 | -2.2118 | -1.1987 | 0.035596 |
| 1 | 0.85667 | 0.93048 | -0.14391 |
| 2 | -0.10838 | -0.17156 | 0.045915 |

temperature of fuel vapor which enters the computational domain at the boundaries. This submodel is only required if the source of fuel is a solid or liquid since gas releases can be specified as a flow boundary condition. Since the generation of fuel vapor from these materials involves, as a minimum, representing thermal transport within the material including phase change, a simplified approach is taken here to serve the basic need of present generation fire models. The development of improved, validated models is presently underway. Present generation models are limited to liquid fuels in the form of pools (i.e., a defined amount of fuel constrained in a pool with fixed, known geometry) and spills onto non-absorbing substrates. (See Martinez and Hopkins [69] for a model of fuel spill in a porous medium.) Although the form of the submodel will allow first order estimates of fire growth rates, data acquired to date (Saito et al. [70]) tend to show that relevant flame spread mechanisms include features which occur at lengths scales several orders of magnitude below the resolution of present grids. Additional submodels will be therefore be required to predict flame spread with confidence. The following quantities are required:

1. $T_{fuel,vap}$, the vaporization temperature of the fuel (K).
2. $h_{fg}$, the heat of vaporization of the fuel (KJ/kg),
3. $C_{p_l}$, the specific heat of the liquid fuel (KJ/kg-K),
4. $T_{fuel,init}$, the initial temperature of the liquid fuel (K),
5. $\alpha_{fuel,liq}$, the absorptivity of the liquid fuel,
6. $q''_{rad}$, the radiative heat flux incident on the fuel surface,
7. $q''_{conv}$, the convective heat flux incident on the fuel surface.

The fuel boundary condition submodel generates the following output:

- $\dot{m}''$, the mass flux of fuel (kg/m$^2$-s).

The fuel pool will be modeled as a mass of liquid that is gradually converted to vapor which in turn enters the flow field as a distinct species. The fuel vapor generation rate is based on the incident heat flux to the pool surface. Data for heavy hydrocarbon fuels (Gritzo, et al. [71, 72]) show the following:

- After the initial transient (which includes flame spread) the fuel burning (and hence vaporization) rate is steady.
- Heating of the fuel is limited to the top 1.5 cm (which greatly exceeds the penetration depth for combined thermal transport in semitransparent media).
- Fuel transport occurs within the pool due to the preservation of a fuel free surface and the presence of a non-uniform heat flux to the fuel surface.
- The temperature at the free surface of the fuel is spatially uniform and approximately equal to the mean of the distillation curve for multi-component fuels.

Given these observations, the present submodel includes two options for calculating the fuel vaporization. These options are used for both pool and spill fires.

### 2.14.1. Option 1: Constant, Specified Mass Flux

In this option, the output of the submodel will be specified directly by the user. Fuel will be released at the boundaries defined by a fuel free surface. Since the burning rate is constant, the mass flux can be considered constant. Fuel burn rate data (for example, Blinov and Khudiakov [73]) are available as a function of pool size for a variety of fuels. This option neglects the physical process of fuel heating and is therefore only appropriate for steady burning fires. The spatial variation of fuel vaporization is also neglected.

### 2.14.2. Option 2: Mass Flux as a Function of Incident Heat Flux

Neglecting the transport of liquid fuel within the pool, the local fuel vapor mass flux is given by

$$\dot{m}'' = \frac{\alpha_{fuel,liq} q''_{rad} + q''_{conv}}{h_{fg} + C_{pl} \left( T_{fuel,vap} - T_{fuel,init} \right)}. \tag{2.463}$$

This option includes the physical process of fuel heating and is therefore appropriate as a zeroth order estimate during fire growth. The spatial variation of fuel vaporization is also neglected.

Before to the surface of the fuel reaches its vaporization temperature, the KAMELEON fire code (Holen, et al. [50]) models the heating of the fuel in the same manner as the heating of solid surfaces. The heat transferred into the material is determined using a linearized approximation for the temperature distribution in the media by

$$q_{abs} = \frac{\rho C_p \left( T_s - T_p \right)}{2} h \tag{2.464}$$

where $q_{abs}$ is the heat absorbed by the material, $T_s$ is the temperature at the exposed surface of the control volume, and $T_p$ is the temperature at the control volume center, $h$ is the fuel thickness, and $\rho$ and $C_p$ are the material density and specific heats, respectively.

Due to low diffusivity and high opacity of hydrocarbon fuels, the temperature gradient in the liquid fuel develops quickly, is considerably larger than the linear approximation, and does not extend to the lower surface of the fuel. The transient fuel heating occurs at the same short time and length scales as flame spread. The inclusion of this feature is not suggested until a more rigorous technique for modeling flame spread can be developed.

## 2.15. FUEL SPREADING SUBMODEL

The VULCAN/KAMELEON fire code includes a model which represents the spreading of fuel on a non-absorbing substrate. This feature allows the simulation of fires resulting from fuel spills. Various correlations (Mansfield and Linley [74]) and global, quasi-steady-state, algebraic models (Cline and Koenig [75]; Magnoli [76]) have been developed to determine the size of a circular pool fire resulting from a fuel spill. Since these models are global in nature, and do not include the effects of complex geometries resulting from obstacles, they will not be included as submodel options. The following quantities are required:

1. $\rho_{fuel}$, the density of the liquid fuel,
2. $\mu_{fuel}$, the viscosity of the liquid fuel,
3. $Q_{release}$, the volumetric flow of fuel released by the spill,
4. $A_s$, the surface area of the element,
5. $\gamma$, the surface tension coefficient of the liquid fuel.

The fuel boundary condition submodel generates the following output:

- $h$, the depth of fuel (m).

The following assumptions are invoked as part of the fuel spreading model presently in VULCAN.

1. The fuel is sufficiently thin for inertial forces to be neglected as compared to shear forces.
2. The velocity components in the fuel are always horizontal.
3. The substrate is smooth, horizontal and non-absorbing.
4. The flow is laminar.
5. The interface between the fuel and air at the front of the spreading fuel is parabolic.
6. The shear stress is zero at the top of the film.

Given the preceding assumptions, the spread of fuel is driven by the difference between hydrostatic pressure due to variations in fuel depth. The transport can then be represented by

$$\frac{\partial h}{\partial t} = \frac{\partial}{\partial x_j}\left(\frac{\rho_{fuel} g h^3}{3\mu_{fuel}}\right)\frac{\partial h}{\partial x_j} + S. \tag{2.465}$$

Equation 2.465 is solved explicitly to track the fuel thickness along the flat surface. Boundary conditions and source terms are defined as follows to represent various physical features.

1. Drains - The depth of fuel is set equal to zero for cells occupied by drains. The volume of fuel transported into the drain cell is removed via a negative source term. occupied by drains. The volume of fuel transported into the drain cell is removed via a negative source term.
2. Obstacles - The fuel depth and the gradient of the fuel depth is set equal to zero at the interface between obstacles and surrounding cells.
3. Release Locations - The source term is defined by the volumetric flow of released fuel divided by the surface area of the element (i.e. $Q_{release}/A_s$).

The fuel will spread up until the hydrostatic pressure gradient is balanced by surface tension forces. Subject to the preceding assumptions, the minimum fuel depth is given by

$$h_{min} = \sqrt{\frac{2s}{\rho_{fuel} g}}, \tag{2.466}$$

where $s$ is the coefficient of surface tension for the fuel. The reduction in fuel depth due to the vaporization of fuel is calculated by the same technique used to define the fuel vapor boundary condition for pool fires.

# 2.16. VOLUME OF FLUID MODEL

Beta Capability: The volume-of-fluid capability is a beta feature. The numerics for laminar isothermal flow are well tested, but advanced features such as non-isothermal reacting and evaporating simulations are more experimental.

The volume-of-fluid (VOF) capability allows simulation of two-phase systems. The phases are tracked with a conserved scalar, $\alpha$, which is the volume fraction of fluid in a given control volume. This scalar is 1 in the liquid, 0 in the gas, and between 0 and 1 in the transition region.

There are a number of numerical challenges associated with multi-phase modeling using a diffuse interface. Many of these challenges exist regardless of whether one uses a level set or a volume-of-fluid approach, or some hybrid of the two.

## 2.16.1. Governing Equation

The basic advection equation for the volume fraction ($\alpha$) in VOF equations is

$$\frac{\partial \alpha}{\partial t} + \vec{u} \cdot \nabla \alpha = S_\alpha \tag{2.467}$$

Expanding the conservative form of the convection term with the chain rule as

$$\nabla \cdot (\vec{u}\alpha) = \vec{u} \cdot \nabla \alpha + \alpha \left( \nabla \cdot \vec{u} \right) \tag{2.468}$$

we can express the governing equation in corrected conservative form as

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot (\vec{u}\alpha) = \alpha \left( \nabla \cdot \vec{u} \right) + S_\alpha \tag{2.469}$$

where the right hand side term is 0 for an incompressible flow with no phase change. The standard advection operator is typically too diffusive for practical use, so one of two approaches are typically used:

1. A geometric advection operator [77, 78, 79, 80]
2. An additional interface compression term [81, 82]

Several varieties of geometric advection operators have been proposed in the literature [77, 78, 79, 80], many of which are only applicable for 2D quadrilateral meshes. Those that can be extended to higher dimensions often do so at considerable complexity. For a more detailed discussion of the different options, see [83]. The form implemented here uses the interface compression approach to modify the governing equation to be

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot (\vec{u}\alpha) + \nabla \cdot (\vec{u_c}\alpha(1-\alpha)) = \alpha \left(\nabla \cdot \vec{u}\right) + S_\alpha \tag{2.470}$$

where $\vec{u_c}$ is the compressive velocity. A common form for this is

$$\vec{u_c} = C_\alpha |\vec{u}| \vec{n} \tag{2.471}$$

where $C_\alpha$ is a constant and $\vec{n}$ is the normal vector at the phase interface.

When phase change is present, the source term $S_\alpha$ is non-zero. The volume source of liquid due to evaporation is

$$S_\alpha = -\frac{\dot{m}_{evap}}{\rho_L} \tag{2.472}$$

The evaporation term is calculated using the method described by Hardt and Wondra [84] which converts an evaporative mass flux to a volume source using the interphase area, $A$, and the mass flux predicted by the evaporation model, $m''_{evap}$, as

$$\dot{m}_{evap} = m''_{evap} A \tag{2.473}$$

$$A = W |\nabla \alpha| \frac{\int |\nabla \alpha| dV}{\int W |\nabla \alpha| dV} \tag{2.474}$$

$$W = \alpha \tag{2.475}$$

One variation of this that we explored is $W = (1-\alpha)^2$, which is well suited for a diffuse interface approach. We find that the form by Hardt and Wondra is sometimes destabilizing to the diffuse interface and leads to an artificially high surface regression rate. For a very simple 1D problem or a sharp interface method, the $W = (1-\alpha)^2$ weighting scheme places all the evaporation outside the liquid and does not correctly predict the surface regression rate. However, it can be better for a diffuse interface representation since evaporation acts as a stabilizing liquid sink to counteract interface diffusion. When using a sharp interface method or a 1D problem where there is no interface diffusion, the weighting of Hardt and Wondra ($W = \alpha$) should be used.

Different evaporation models can be used to define $m''_{evap}$. The model used by Hardt and Wondra uses deviation from the saturation temperature to define the rate as

$$m''_{evap} = K(T - T_{sat}) \tag{2.476}$$

$$K = \frac{2\beta}{2-\beta}\frac{h_{fg}}{\sqrt{2\pi R}}\frac{\rho_g}{T_{sat}^{1.5}} \tag{2.477}$$

This form is only physically applicable to either fully saturated gasses, or interface temperatures above the boiling point since it does not take into account the gas composition. As such, this model only allows positive evaporation rates (so no condensation) in order to produce realistic behavior during heat-up and away from the hot regions of the simulation. When using a multi-species gas, the more complete Hertz-Knudsen-Langmuir model can be used, where the rate is driven by the difference between the partial pressure of the vapor and the saturation pressure. This model will be added to the available evaporation models once the multi-species gas treatment is complete.

### 2.16.2. Interface Reconstruction

When the phase interface is represented by a volume-of-fluid variable, numerical diffusion in the normal advection operator causes the interface (cells with volumes of fluid between 0 and 1) to become wider over time. Although the total volume of liquid is conserved, the volume inside the $\alpha = 0.5$ isosurface may change as a result, and the effect of the higher density liquid will be felt well outside the correct liquid interface location. Techniques to counter-act this diffusion in volume-of-fluid include using compressive advection operators in the VOF equation, using a separate anti-diffusive sharpening step, using a geometric reconstruction of the interface to define the advection, and using sharpening and liquid relocation schemes.

With a level set approach, the interface location is defined by the $\phi = 0$ contour. By definition, the interface does not become diffuse, but there are no guarantees of global conservation of liquid either. Techniques to address this typically involve doing a volume-conserving redistancing operation to the level set field. Such operations can preserve the total volume of liquid, but may not preserve the local shape of the liquid-gas interface.

In both VOF and level set approaches, calculation of the interface curvature is required to determine surface tension forces. This is done by taking the divergence of the interface normal vector ($\vec{n}$).

$$\kappa = -\nabla \cdot \vec{n} \tag{2.478}$$

It should be immediately apparent that $\kappa$ will contain not only discretization error from the divergence operator, but also discretization error from the calculation of the normal vector, which is itself a gradient of some other quantity (call it $\alpha_s$ here, it could be the level set field, the volume of fluid field, or a smoothed version of either).

$$\vec{n} = \frac{\nabla \alpha_s}{|\nabla \alpha_s| + \epsilon} \tag{2.479}$$

The simple approach to use $\alpha_s = \alpha$ results in considerable noise in the interface normal vector calculation, particularly in regions of high curvature relative to mesh sizes. In the coupled level set VOF approach (CLSVOF), $\alpha_s$ is the companion level set distance function that is either advected along with the volume fraction or redistanced from the volume fraction contour at $0.5$ (so $\alpha_s = \phi$).

In Fuego we have implemented both a CLSVOF approach and a diffusive approach, where $\alpha_s$ is a diffusively smoothed version of the volume fraction field, $\alpha$, using a user-specified Fourier number ($Fo$) and number of iterations.

$$\alpha_s^0 = \alpha \tag{2.480}$$

$$\alpha_s^{n+1} = \alpha_s^n + Fo\Delta x^2 \nabla^2 \alpha_s^n \tag{2.481}$$

In addition, recall that the analytical curvature for a spherical drop is a function of radius: $\kappa = 2/r$ . This means that if $\kappa$ is calculated in cells that span an interface of finite, non-zero width, there will also be a spatial variation of $\kappa$ across that interface. This means that although in reality the surface tension force is applied sharply at the interface, in a diffuse interface scheme it will be applied over a finite interface thickness of several cells.

This noise and spatial variation in the curvature translates directly into noise in the surface tension force and is the sole cause of the so-called "parasitic currents" that plague diffuse interface methods. Many authors have demonstrated that when using a uniform, analytically prescribed curvature the parasitic currents in the classical static drop test drop to machine precision levels if the forces are otherwise properly balanced.

There are various techniques for improving the curvature calculation. Calculation of the normal vector when the VOF field is sharp is very noisy, so VOF schemes often use a smoothed field or higher order least squares technique. Coupled level set VOF (CLSVOF) carries a level set field along to use for calculating this normal vector on a smooth field (the level set field).

Finally, once the interface has been advected and sharpened, and the curvature has been properly calculated, the surface tension force must be integrated into the momentum and continuity equations in such a manner as to not induce spurious velocities. This is known as the "balanced force" technique, and involves including a surface tension force vector consistently with the pressure gradient.

## 2.16.3. Continuity Equation

In order to handle the high density ratios in multiphase simulations, the form of the continuity equation is modified by dividing both sides of the equation by density. Interpolation of nodal quantities to subcontrol surfaces is an important consideration to note. Quantities interpolated to faces are indicated with an $f$ subscript in the following equations, while nodal quantities are given no subscript.

The regular form of the pressure Poisson equation in discrete form is

$$\sum_f \left(\tau_f \nabla p_f \cdot A_f\right) = \sum_f \left(\left(\rho \vec{u}\right)_f \cdot A_f + \left(\tau \nabla p\right)_f \cdot A_f\right) \tag{2.482}$$

which includes the projected nodal gradient of pressure $\nabla p$. In the prior form, the projected nodal gradient $\nabla p$ was calculated using the divergence theorem as

$$\nabla p = \frac{1}{V} \sum_f p_f A_f \tag{2.483}$$

For the VOF implementation we use the method described by Francois et al. [85] to calculate a projected nodal density-scaled force term ($F$) based on calculation of forces at faces, which includes both pressure gradient and surface tension forces. The balanced force term is

$$\vec{F} = \nabla p - \kappa \sigma \nabla \alpha \tag{2.484}$$

and the projection from subcontrol faces to nodes is done using the method by Francois as

$$F_j = \frac{\sum_f \left(\nabla p_{f,j} - \left(\sigma\kappa\right)_f \nabla \alpha_{f,j}\right) / \rho_f}{\sum_f \left|A_{f,j}\right|} \tag{2.485}$$

This term is updated after the solution to the pressure equation, when updated values for pressure and volume-of-fluid have both been calculated in order to keep them synchronized. This nodal force term is included as a source term in the momentum equation (negative, scaled by volume, and multiplied by nodal density) and can be added to the 4th order pressure projection scheme in discrete form as

$$\sum_f \left(\left(\frac{\tau}{\rho}\right)_f \nabla p_f \cdot A_f\right) = \sum_f \left(\vec{u}_f \cdot A_f + \left(\tau \vec{F}\right)_f \cdot A_f + \left(\frac{\tau}{\rho}\right)_f \left(\sigma\kappa\right)_f \nabla \alpha_f \cdot A_f\right) \tag{2.486}$$

where $\vec{F}$ is the projected nodal density-weighted balanced force, $\kappa$ is the curvature, and $\sigma$ is the surface tension.

The advantage of this projection scheme over a simpler approach where one might define the nodal force term (as done by Lin et al. [86]) as

$$F = \frac{1}{V}\left(\sum_f p_f A_f - \sigma\kappa \sum_f \alpha_f A_f\right) \tag{2.487}$$

is that in such a scheme the evaluation of curvature ($\kappa$) occurs at a different location than the gradient of the volume-of-fluid term ($\nabla\alpha$). In the presence of gradients in $\kappa$–which are guaranteed to occur in real problems–this introduces a force imbalance that can cause significant spurious non-physical currents in the fluid. Spatial inaccuracies in curvature are the primary cause of spurious currents in VOF

simulations âĂŞ and a baseline test for any balanced force implementation is to see how a static drop with a prescribed (not calculated) value for curvature behaves. In a properly balanced force implementation the resulting parasitic currents should be basically zero (on the order of machine precision). However, a weakness of relying solely on a prescribed curvature test is that it fails to show the force imbalance that arises in the projected nodal force calculation due to interpolation of the curvature. The form published by Francois et al. [85] and implemented in Sierra/Fuego not only recovers machine-precision parasitic currents with the prescribed curvature test, but also keeps them acceptably low when using calculated curvatures.

In addition, when there is phase change there is an additional source term on the right hand side of this equation, which is

$$S = \dot{m}_{evap} \left( \frac{1}{\rho_g} - \frac{1}{\rho_L} \right) \tag{2.488}$$

## 2.16.4. Property Evaluation

When using VOF, you are required to provide three material models. There is one model for the liquid, one for the gas, and one top level model. The top level model simply specifies the names for the materials to use for gas and liquid, as well as specifying properties like surface tension. Properties needed for flow calculations are combined from the liquid and gas properties using the following rules

A simple averaging is used for non-specific quantities,

$$\phi = \alpha\phi_L + (1 - \alpha)\phi_g \tag{2.489}$$

where $\phi$ is thermal conductivity ($k$), density ($\rho$), viscosity ($\mu$), absorption ($a$), or density-pressure-derivative ($\frac{\partial \rho}{\partial p}$).

For specific quantities, such as specific heat ($C_p$) and enthalpy ($h$), a density-weighted average is used

$$\phi = \frac{\alpha\rho_L\phi_L + (1 - \alpha)\rho_g\phi_g}{\rho} \tag{2.490}$$

For properties that are only relevant for the gas phase, the property value is simply copied from the gas phase

$$\phi = \phi_g \tag{2.491}$$

Examples include species enthalpy, mass diffusivity, and molecular weight.

# 3. MPMD RADIATION

This chapter outlines the use of Multiple-Program-Multiple-Data (MPMD) coupling to perform radiation solves with two different codes: Scefire and Nalu. MPMD coupling allows the radiation solve to occur on an entirely different set of processors using an executable other than Fuego.

## 3.1. INPUT DECK CONVERSION

The Conversion of a Fuego/Syrinx input deck to a Fuego MPMD input deck is straightforward. The basic steps to convert a Syrinx case to a Fuego-Nalu case are:

1. Delete the Syrinx region, PMR material, PMR solver, and PMR mesh (finite element) blocks from the Fuego input deck.

2. Delete all transfer blocks to or from the PMR region and all calls to them from solution control.

3. Add the `USE MPMD RADIATION` command to the Fuego input deck in the Fuego region block to activate the MPMD coupling.

4. Add the `SKIP STEPS FOR PMR` command in the Fuego procedure or time blocks if you want to run the PMR solve less frequently than every time step.

5. Specify emissivity, transmissivity, radiation environment temperature, and radiation boundary temperature in the Fuego boundary condition blocks. If any of these are omitted defaults will be used. Default conditions are emissivity = 1, transmissivity = 0, and boundary temperature equal to the local or wall temperature depending on the boundary type.

6. Copy the template Nalu input file from Section 3.1.1 and update the name of the mesh file.

7. (Optional) Update the quadrature and numerical scheme in the Nalu input deck per your problem requirements.

8. (Optional) If using a particle region, transfer absorption and radiation sources from the particle region to the Fuego region, and transfer scalar flux from the Fuego region to the particle region (see Section 3.3).

9. Run your case using an MPMD launch command (see Section 3.1.3).

### 3.1.1. Nalu Input Deck

The corresponding Nalu input deck for Fuego/Nalu MPMD calculations appears as follows:

```
Simulations:
  - name: sim1
    time_integrator: ti_1
    optimizer: opt1

MPMD_transfer_run:
linear_solvers:

  - name: solve_scalar
    type: tpetra
    method: gmres
    preconditioner: sgs
    tolerance: 1e-5
    max_iterations: 100
    kspace: 100
    output_level: 0
    write_matrix_files: no

transfers:

  - name: xfer_fluid_pmr
    realm_pair: [MPMDRealm, pmrRealm]
    coupling_physics: mpmd_pmr

  - name: xfer_fluid_pmr2
    realm_pair: [pmrRealm, MPMDRealm]
    coupling_physics: mpmd_pmr

realms:

  - name: MPMDRealm
    mesh: MPMDVirtual
    type: MPMDRealmType

  - name: pmrRealm
    mesh: ./100cmCube.g
    use_edges: yes
    automatic_decomposition_type: rcb

    boundary_conditions:
```

```
    - wall_boundary_condition: walls
      target_name: all_surfaces

    material_properties:
      target_name: all_blocks

    equation_systems:
      name: theEqSys
      max_iterations: 1

      solver_system_specification:
        intensity: solve_scalar

      systems:
        - RadiativeTransport:
            name: myRTE
            max_iterations: 1
            convergence_tolerance: 1.e-8
            quadrature_order: 6
            activate_scattering: no
            activate_upwind: no
            external_coupling: yes
            deactivate_sucv: no

    output:
      output_data_base_name: pmr.e
      output_frequency: 1
      output_node_set: no
      output_variables:
       - absorption_coefficient
       - intensity_bc
       - scalar_flux
       - radiative_heat_flux
       - radiation_source
       - irradiation
       - emissivity
       - transmissivity

Time_Integrators:
  - StandardTimeIntegrator:
      name: ti_1

      realms:
        - pmrRealm
        - MPMDRealm
```

While the input deck is more complicated than the Scefire input deck, the user is required to specify nearly the same set of data, including the mesh, `output_data_base_name` (along with desired output fields), `quadrature_order`, and solver settings. Nalu supports the Thurgood quadrature (identical to $Tn$ from Syrinx), which has $8N^2$ ordinate directions, $N$ being the order of the quadrature as well as level-symmetric quadratures (Lathrop Carlson orders 2,4,6,8,12, and 16 and $PN$-$TN$ for any even order) which have $N(N+2)$ ordinate directions. The default quadrature is Thurgood. Unlike Scefire, Nalu requires the user to specify the name of the sidesets to be used in the `boundary_conditions` specifications. However, for most cases the `all_surfaces` alias can be used. In the case where it cannot be used (e.g. a mesh with internal sidesets), the wall BC sidesets can be supplied as a list (e.g. `[surface_1, surface_2, surface_3]`)

Fuego MPMD simulations will terminate Nalu when Fuego is finished executing. Nalu has a number of discretization schemes that can be used as well including edge-upwind and both edge and element-based SUCV.

### 3.1.2. SceFire Input Deck

Scefire input decks have the following structure:

```
<?xml version="1.0" encoding="utf-8"?>
<SCEPTRE_Input>
  <Mesh_File>100cmCube.g</Mesh_File>
  <Output_Prefix>radResults.e</Output_Prefix>
  <Output_Format>Simple</Output_Format>
  <Output_Options>
    <Verbosity>high</Verbosity>
  </Output_Options>
  <XS_File>thermal-fromShawn.xslib</XS_File>
  <Sn_Options>
    <Sn_Order>6</Sn_Order>
    <Angular_Quadrature_Type>Level_Symmetric</Angular_Quadrature_Type>
    <!--Angular_Quadrature_Type>Lebedev</Angular_Quadrature_Type-->
  </Sn_Options>
  <Enable_User_Defined_Solvers>true</Enable_User_Defined_Solvers>
  <Solvers>
    <Solver name="1stOrder">
      <Solver_Form>First_Order</Solver_Form>
      <Error_Control_Options>
        <Maximum_Number_Iterations>20</Maximum_Number_Iterations>
        <Convergence_Tolerance>1.e-8</Convergence_Tolerance>
      </Error_Control_Options>
    </Solver>
  </Solvers>
  <Solver_Assignment explicit="true">
```

```
    <Solver_By_Group>
      <Group index="1">1stOrder</Group>
    </Solver_By_Group>
  </Solver_Assignment>
</SCEPTRE_Input>
```

The Scefire input deck requires very little user input. The first required component is the `Mesh_File`, here specified as "100cmCube.g". The Scefire mesh does not need to be identical to the Fuego fluid mesh file, but currently Scefire uses a closest node interpolation of data, so that limitation should be taken into consideration. Different decompositions (including numbers of processors) for the Scefire mesh relative to the Fuego Fluid mesh can effectively be used. The name of the output file (radResults.e) is a required entry, though most users will only use output from the Fuego exodus files. Using the "Simple" output format suppresses the Scefire output.

The `Angular_Quadrature_Type` is set here to `Level_Symmetric`, though Lebedev is also available in Scefire. The user also selects the order of the `Angulare_Quadrature` (`Sn_Order`), here set to 6. `Level_Symmetric` quadratures, as in Syrinx, have $N(N+2)$ ordinate directions, where $N$ is the `Sn_Order`. The `First_Order` solver should be left unchanged for all current Fuego/Scefire MPMD simulations, though Maximum Iterations and Convergence tolerances can be changed at the user?s discretion. The rest of the Scefire input deck should remain unchanged. Note that the user does not need to specify any boundary conditions in the Scefire input deck.

### 3.1.3. Running MPMD Jobs

The command to run in MPMD mode is different from what was used to run Fuego-Syrinx cases. To run Fuego-Syrinx cases you simply ran a parallel job of Fuego using something like

```
mpirun -np 100 fuego -i fire.i
```

In this case one executable would use 100 cores, so both Fuego and Syrinx were taking turns using the same CPU resources. With MPMD runs you are launching two separate MPI jobs with two different codes that can communicate. An example MPMD launch command would look like

```
mpirun -np 100 fuego -i fire.i : -np 100 nalu -i pmr.i
```

However, there is no requirement any more that the two codes use the same number of cores, so depending on the mesh and computational costs you may choose a different allocation per code. For example, if your PMR solve is very expensive you may allocate more cores to Nalu than Fuego:

```
mpirun -np 50 fuego -i fire.i : -np 150 nalu -i pmr.i
```

Special care must be taken when submitting MPMD jobs on the HPCs or any queued environment. By default, the two MPMD codes cannot share cores so to launch the case above on an HPC you would need to request an allocation 200 cores. This is unnecessarily wasteful though since Nalu would not be using its 150 cores while Fuego runs, and Fuego would not be using its 50 cores while Nalu runs. To get around this, you must enable oversubscription. Since Fuego and Nalu run sequentially (never executing at the same time) you can allow them to share resources. To get an allocation of 100 cores and use them all for both codes you must add additional mpi flags:

```
mpiexec --oversubscribe --bind-to core:overload-allowed -np 100  \
  fuego -i fire.i : --bind-to core:overload-allowed -np 100 nalu -i pmr.i
```

Keep in mind that the specific command to use can be platform dependent. A more complete example submission script on an HPC may look like

```
#!/bin/bash

#SBATCH --nodes=10
#SBATCH --time=48:00:00
#SBATCH --account=xxxxxxxx
#SBATCH --job-name=fire
#SBATCH --partition=batch

nodes=$SLURM_JOB_NUM_NODES
cores=36

module load sierra
export OMPI_MCA_rmaps_base_oversubscribe=1
mpiexec --oversubscribe                                          \
  --bind-to core:overload-allowed                                \
  --npernode $cores --n $(($cores*$nodes)) fuego -i fire.i :     \
  --bind-to core:overload-allowed                                \
  --npernode $cores --n $(($cores*$nodes)) nalu -i pmr.i
```

Contact `sierra-help@sandia.gov` if you need more help or encounter issues running MPMD jobs.

## 3.2. SPECTRAL RADIATION TRANSPORT

Fuego now supports spectral radiation calculations through MPMD coupling with either Scefire or Nalu (not Syrinx). Spectral radiation transport is instantiated in an MPMD Fuego input deck through a spectral model section similar to the following:

$$
\begin{array}{ll}
T_1 & F_{1,1} \quad A_{1,1} \\
\cdots & \\
T_1 & F_{1,N_1} \quad A_{1,N_1} \\
\cdots & \\
T_M & F_{M,1} \quad A_{M,1} \\
\cdots & \\
T_M & F_{M,N_M} \quad A_{M,N_M}
\end{array}
$$

```
BEGIN RAD TRANSPORT SPECTRAL MODEL SPECIFICATION spectralRadModel
  Spectral Band Model = LINEAR
  Spectral Subband Model = AVERAGE
  Minimum Frequency = 1.0e13
  Maximum Frequency = 2.0e13
  Number Spectral Bands = 2
END RAD TRANSPORT SPECTRAL MODEL SPECIFICATION
```

The spectral model utilized in Fuego is based on the concept of spectral bands, which means that the user must describe frequency bands over which radiation absorptivities are averaged. In this context, a richer set of data is available to the analyst, including radiation transport terms (banded absorptivity, scalar flux, and radiative flux) than is available through the standard gray (nonbanded) radiation transport description that has been available to Fuego analysts for many years.

Spectral transport makes use of spectral data files that the user must provide. Files are named as:

```
Species-data.dat
```

Where "Species" is the name of the chemical species whose spectral transport properties are being described. Each species present (all species for EDC simulations) or those output through the use of output variables in mixture fraction simulations must have all necessary spectral data files available in the working directory when a spectral transport calculation is performed. Spectral data files have the following structure:

For each temperature $T_x$ of the M temperatures specified in the spectral data files, there are $N_X$ frequencies ($F_{x,1}$ through $F_{x,N_X}$) specified with corresponding absorptivity values ($A_{x,1}$ through $A_{x,N_x}$). Temperatures within these files are required to be in non-decreasing order and frequencies must be ordered in increasing value within a specific temperature. Absorptivities are expected to be in units of $cm^2/molecule$, which is a number density weighted absorptivity. Unit conversion within the Fuego input deck is required if the desired units of the simulation are not cgs. The ideal gas law and knowledge of the mass fraction of each species is used to calculate the number density.

For the spectral radiation model specified here, the name of the particular spectral model is given as spectralRadModel, but could be any valid string. For this case, a `LINEAR` "Spectral Band Model" is used, where the set of spectral bands, in this case 2 as specified by "Number of Spectral Bands" are equally divided along the linear spectral frequency range with "Minimum Frequency = 1.0e13" Hz and

"Maximum Frequency = 2.0e13" Hz. One could alternatively choose a `LOGARITHMIC` Spectral Band Model where bands are divided up equally in logarithmic frequency space.

In this case, we have chosen an `AVERAGE` "Spectral Subband Model", which indicates that averaging of absorptivities for frequencies within a spectral band will follow standard (unweighted) averaging over the particular spectral band. A `PLANCK_AVERAGE` Spectral Subband Model is also available where integrations are weighted by their Planck black-body spectrum contribution. `PLANCK_AVERAGE_WITH_REFERENCE_TEMPERATURE` is a final choice that a user can select where the band contributions are weighted by the Planck black body-spectrum at a specified user reference temperature. If this final option is used, a user must also specify a reference temperature, which can be done through: Planck Subband Model Reference Temperature = 2000.0 Here the Reference temperature is set to 2000.0 K, but could be any nonzero temperature.

Please also note that if running a spectral case with Nalu multiple non-linear iterations should be specified. This is currently necessary to converge the system even if the equation system is linear.

## 3.3. PARTICLE RADIATION TERMS

Particle Radiation contributions are also still available in Fuego simulations with Lagrangian particle types that have radiation contributions (heated, CPD, evaporating, general chemistry, wildfire). Previously, the user would specify transfers of necessary thermal/radiation fields between Fuego and Syrinx, which would resemble the following:

```
BEGIN TRANSFER pmr_to_particle
    COPY VOLUME NODES FROM pmr_region TO particle_region
    SEND BLOCK block_1 TO block_1
    SEND FIELD scalar_flux STATE none TO scalar_flux STATE new
END   TRANSFER pmr_to_particle

BEGIN TRANSFER particle_to_pmr
    COPY VOLUME NODES FROM particle_region to pmr_region
    send field particle_absorption_coeff state new to particle_absorption_coeff state no
    SEND FIELD particle_rte_source STATE  new TO particle_rte_source STATE none
END TRANSFER particle_to_pmr
```

For MPMD Radiation transport simulations involving Lagrangian particles, the user now removes the pmr(Syrinx) to particle transfers and instead adds these to the already-present set of fluid to particle transfers as is seen in the following:

```
BEGIN TRANSFER fluid_to_particle
    COPY VOLUME NODES FROM fluid_region TO particle_region
    SEND BLOCK block_1 TO block_1
    ...
```

```
    SEND FIELD scalar_flux STATE none TO scalar_flux STATE new
END   TRANSFER fluid_to_particle

BEGIN TRANSFER particle_to_fluid
    COPY VOLUME NODES FROM particle_region TO fluid_region
    SEND BLOCK block_1 TO block_1
    ...
    send field particle_absorption_coeff state new to particle_absorption_coeff state no
    send field particle_rte_source state new to particle_rte_source state none
END   TRANSFER particle_to_fluid
```

## 3.4. BEAM (DIRECTED FLUX)

Fuego users are able to now specify beam or directed flux boundary conditions for use in MPMD simulations with PMR. A beam boundary condition is specified in the Fuego input deck in the following manner:

```
#CONVERGING/DIVERGING BEAM
BEGIN BEAM RADIATION BOUNDARY SPECIFICATION
     BEAM RADIATION BOUNDARY MODEL = FOCUSED
     FOCAL POINT = FX FY FZ
     FOCAL POWER = FP
     FOCUSED BEAM TYPE = DIVERGING (OR CONVERGING)
     CONSTRAIN TO SURFACES = sideA sideB ...
     BEAM TEMPERATURE = BT
 END BEAM RADIATION BOUNDARY SPECIFICATION
```

In the above case, a focused beam boundary `BEAM RADIATION BOUNDARY MODEL = FOCUSED` is defined where the `FOCAL POINT` of the beam, either sourcing from or arriving at this point are (`FX, FY, FZ`). The user specifies whether the beam diverges from this point or converges to this focal point. For the `DIVERGING` beam, the focal point needs to lie outside the simulation domain, whereas for the `CONVERGING` case, the focal point could lie in or outside the domain. The focal power is specified as `FP` which is the integrated power of the beam assuming the intensity is spherically isotropic. For a converging case, the power should be properly scaled to respect the actual solid angle over which the beam converges to the focal point. For instance if the converging beam actually occupies a solid angle of $\pi$, the `FOCAL POWER` should be scaled up by a factor of $4 = 4\pi/\pi$ since if the beam were incident from all solid angles, the total beam power would increase by this factor. Beams can be constrained to arrive through the domain through only a fixed set of external surfaces if so desired as indicated through `CONSTRAIN TO SURFACES = sideA sideB...`. The user also specifies the `BEAM TEMPERATURE` for cases using spectral radiation transport, since the beam power must then be parceled out among the defined spectral bands.

Another "directed" or beam-type boundary condition is available, where the user specifies not a focused beam but rather a plane wave as is seen below:

```
#PLANE WAVE
BEGIN BEAM RADIATION BOUNDARY SPECIFICATION
     BEAM RADIATION BOUNDARY MODEL = PLANE_WAVE
     PLANE WAVE BEAM DIRECTION = WX WY WZ
     PLANE WAVE INTENSITY = PWI
     CONSTRAIN TO SURFACES = sideA sideB ...
     BEAM TEMPERATURE = 3000.0
END BEAM RADIATION BOUNDARY SPECIFICATION
```

In this case, the `BEAM RADIATION BOUNDARY MODEL = PLANE_WAVE`, indicates a plane-wave type source is to be used. The user also specifies the direction for the plane wave as `PLANE WAVE BEAM DIRECTION = WX WY WZ`. The direction must be non-zero. The intensity of the plane-wave in units of Power per area is defined through `PLANE WAVE INTENSITY = PWI`. The ability to constrain the beam to only certain external surfaces is identical to that for the focused beam above as is the beam temperature.

One should keep in mind that for both beam types (focused and plane wave), beam vectors at external surfaces are only calculated for nodes on surfaces where the inner (dot) product of outwardly directed area vector and the beam flux vector is less than 0, indicating that a user specified beam must enter rather than exit the simulation domain. Of course, given the physics internal to the domain of simulation, a radiative flux vector can exit the simulation, but we disallow the possibility of a user specifying an outwardly directed beam that does not result from internal physics.

Beta Capability:

# 3.5. ONE-DIMENSIONAL COMPOSITE FIRE BOUNDARY CONDITION

## 3.5.1. Conceptual Overview

Fuego includes a boundary condition that is capable of modeling the thermal decomposition and outgassing of a thin sheet of porous material at the boundary surface, initially intended to simulate the combustion of a sheet of carbon fiber composite material. Variation through the material thickness is assumed to be locally one-dimensional. The actual implementation is quite flexible, allowing the simulation of the thermal response of essentially any finite-thickness material that can optionally undergo a user-specified chemical decomposition mechanism.

Figure 3.5-1 illustrates a two-dimensional representation of the virtual mesh used for this 1D composite fire boundary condition. One layer of elements above the boundary is shown, within which Fuego performs its normal fluid solve using the control volume finite element CVFEM method. The CVFEM sub-control volumes are demarcated with dashed lines. An equal-order interpolation methodology is used, so that all solution variables are stored at the element vertices.

For this boundary condition, a series of independent one-dimensional virtual domains exist behind each CVFEM surface node, and each virtual 1D domain has a cross-sectional area that matches the group of CVFEM boundary sub-control surfaces that contain the single "parent" surface node. A classical cell-centered finite volume methodology is used for the 1D virtual domains, where the discretization, storage, and numerical solutions all occur within the boundary condition implementation and only interact with the main CVFEM flow solution through fluxes and solution variables at the exposed surface.

Each 1D domain is assumed to have a fixed geometry that is filled with a simple porous material that is allowed to react chemically to form gaseous species. Since the overall volume of each element is fixed, the porosity of each volume is assumed to increase as species are converted from solid to gas. It is assumed that the gaseous species within the pores of the solid phase are of secondary concern, and as such no discrete transport equation is solved for them. The approximation is instead made that all gases generated within the porous material appear instantaneously at the surface of the material as a flux into the main fluid solution. It would be straightforward to solve additional transport equations for fluid flow within the porous material if that level of fidelity were to become necessary, as in the case of oxidative reactions where oxygen must diffuse through the exposed surface into the porous material before reactions may occur.

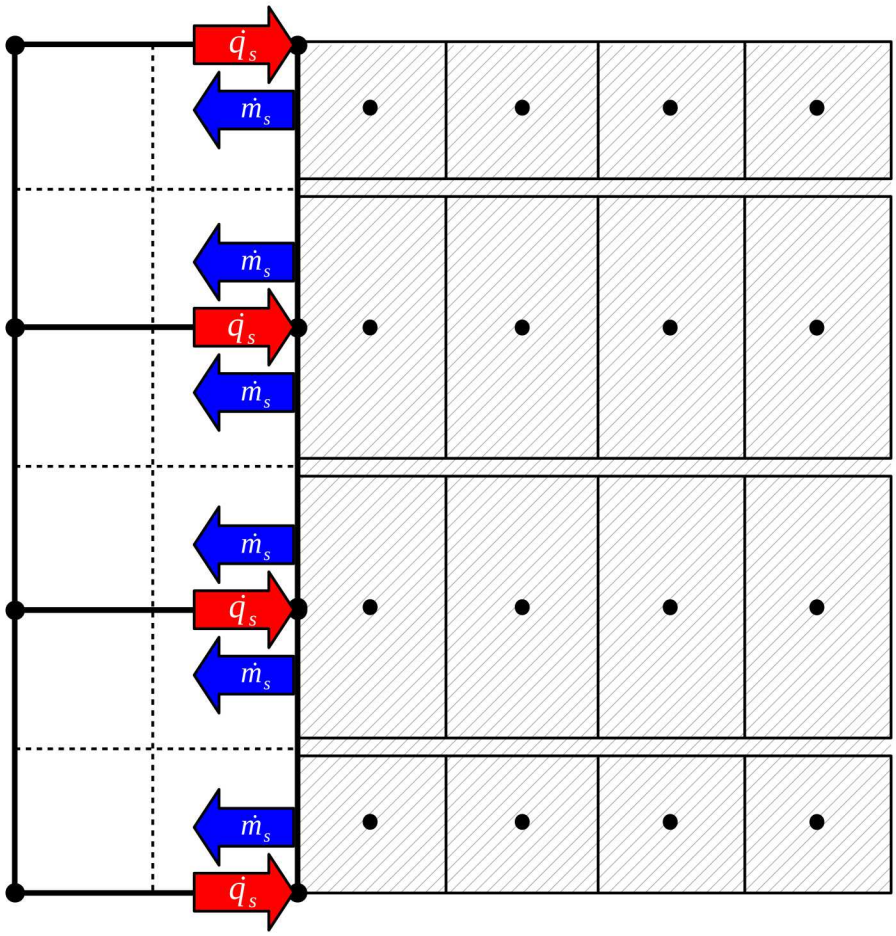


**Figure 3.5-1.. Representative mesh layout for 1-D composite fire boundary condition**

## 3.5.2. Model Formulation

### *3.5.2.1. Transport Equations*

Within the solid phase of the porous material, one-dimensional transport equations for continuity, chemical species, and energy are solved in the form:

$$\frac{\partial \bar{\rho}}{\partial t} = \dot{\omega}_c''' \tag{3.1}$$

$$\frac{\partial \bar{\rho} Y_k}{\partial t} = \dot{\omega}_k''' \tag{3.2}$$

$$\bar{\rho}\bar{c}\frac{\partial T}{\partial t} = \frac{\partial}{\partial x}\left(\bar{k}\frac{\partial T}{\partial x}\right) + \dot{q}''', \tag{3.3}$$

where $\bar{\rho}$, $\bar{c}$, and $\bar{k}$ are the mixture-averaged bulk density, specific heat, and thermal conductivity, respectively, $Y_k$ is the mass fraction of chemical species $k$, $T$ is the temperature of the solid phase, $\dot{q}'''$ is

the volumetric heat generation rate due to chemical reactions, $\dot{\omega}_k'''$ is the volumetric mass generation rate of chemical species $k$, and $\dot{\omega}_c'''$ is the overall mass generation rate computed as $\dot{\omega}_c''' = \sum_k \dot{\omega}_k'''$.

#### 3.5.2.2. Material Models

The composite material used for this boundary condition is assumed to be of a fixed volume, *i.e.* there is no structural deformation allowed. The bulk density of the multi-species solid mixture is assumed to be a function of the density of each component species in their native porous state, as

$$\bar{\rho} = \left( \sum_k \frac{Y_k}{\rho_k} \right)^{-1}, \tag{3.4}$$

where $\rho_k$ is the porous density of species $k$, provided as a material model by the user. This model for the mixture bulk density is only used to compute the initial bulk density field, which is subsequently solved directly from Equation 3.1.

The porosity of the mixture is assumed to follow the model

$$\bar{\psi} = \sum_k X_k \psi_k, \tag{3.5}$$

where $X_k$ is the volume fraction of species $k$,

$$X_k = \bar{\rho} \frac{Y_k}{\rho_k}, \tag{3.6}$$

and $\psi_k$ is the porosity of pure species $k$, modeled as

$$\psi_k = 1 - \frac{\rho_i}{\rho_{s0,k}}, \tag{3.7}$$

where $\rho_{s0,k}$ is the density of the solid (non-porous) species $k$ at a reference temperature. Note that the porosity does not appear explicitly in any of the transport equations or subsequent material models, so that it is never computed as part of the boundary condition solution. It would only appear in transport equations for the gaseous species occupying the pores of the solid skeleton, if this level of detail were ever to be added to this model.

In their most detailed form, the bulk thermal conductivity and specific heat are evaluated as a volume average and mass average of the individual species properties, respectively, as

$$\bar{k} = \sum_k X_k k_k \tag{3.8}$$

$$\bar{c} = \sum_k Y_k c_k, \tag{3.9}$$

although a species-independent model for the overall bulk property may be used if the individual species properties are not known.

The last quantities that require a model are the volumetric species mass production rates, $\dot{\omega}'''$, and the volumetric heat production rate, $\dot{q}'''$. These quantities can be provided by the user in two different ways. The traditional approach is to supply them using standard material property evaluations as a part of the material model definition. These are arbitrary functions that themselves may be dependent on any of the solution variables or other material properties. If a nonreacting material is desired, then these terms may be simply modeled as zero.

The second way of supplying these quantities is by including a chemistry description block in the material model, which allows the user to specify multiple reactions and variable composition gas production.

#### 3.5.2.3. Boundary Conditions

The exposed surface of the composite material interacts thermally with the environment through several mechanisms, including convective heat transfer and both radiation absorption and emission. These external fluxes must balance the conduction inside the composite material at the surface, as

$$\begin{aligned} \dot{q}''' &= \dot{q}'''_{\text{conv}} + \dot{q}'''_{\text{rad}} \\ &= \dot{q}'''_{\text{conv}} + \epsilon \left( \sigma T_1^4 - \dot{q}'''_{\text{irr}} \right) \end{aligned} \tag{3.10}$$

where $\dot{q}'''_{\text{conv}}$ is the convective flux imposed on the surface by the external laminar or turbulent boundary condition treatment, $T_1$ is the temperature solution from the first control volume in the composite material used to model the gray emission, and $\dot{q}'''_{\text{irr}}$ is the external radiative flux incident on the surface.

On the back-side of the virtual composite material, optional convective and radiative heat transfer to a quiescent environment is modeled as

$$\begin{aligned} \dot{q}'''_{\text{b}} &= \dot{q}'''_{\text{b,conv}} + \dot{q}'''_{\text{b,rad}} \\ &= h_c \left( T_N - T_{\text{ref}} \right) + \sigma \epsilon_b \left( T_N^4 - T_{\text{ref}}^4 \right) \end{aligned} \tag{3.11}$$

where $h_c$ is a user-specified convection coefficient, $\epsilon_b$ is a user-specified back-side emissivity, $T_{\text{ref}}$ is the modeled ambient environment temperature, and $T_N$ is the temperature of the solution node closest to the back-side surface, assumed to be equal to the back-side surface temperature itself.

#### 3.5.2.4. Numerical Implementation - Original

A segregated, implicit solution technique is used to numerically integrate Equations 3.1–3.3. The discretized form of the continuity equation, Equation 3.1, is derived by first integrating it over the finite volume $V$ and the time step $\Delta t$ to yield

$$\int_{\Delta t} \left[ \int_V \frac{\partial \bar{\rho}}{\partial t} dV - \int_V \dot{\omega}'''_c dV \right] dt = 0 \tag{3.12}$$

$$\int_{\Delta t} \left[ V_i \frac{\partial \bar{\rho}_i}{\partial t} - V_i \dot{\omega}'''_{c,i} \right] dt = 0. \tag{3.13}$$

Discretizing the temporal derivative using a first-order backward difference approximation and solving for the bulk density at the new time step yields

$$V_i \left( \bar{\rho}_i^{n+1} - \bar{\rho}_i^{n} \right) - V_i \dot{\omega}_{c,i}''' \Delta t = 0 \tag{3.14}$$

$$\bar{\rho}_i^{n+1} = \bar{\rho}_i^{n} + \dot{\omega}_{c,i}''' \Delta t. \tag{3.15}$$

where the mesh indices are defined in Figure 3.5-2. Note that this equation is linearized by evaluating the source term at the most recent estimate of the $n+1$ solution state.

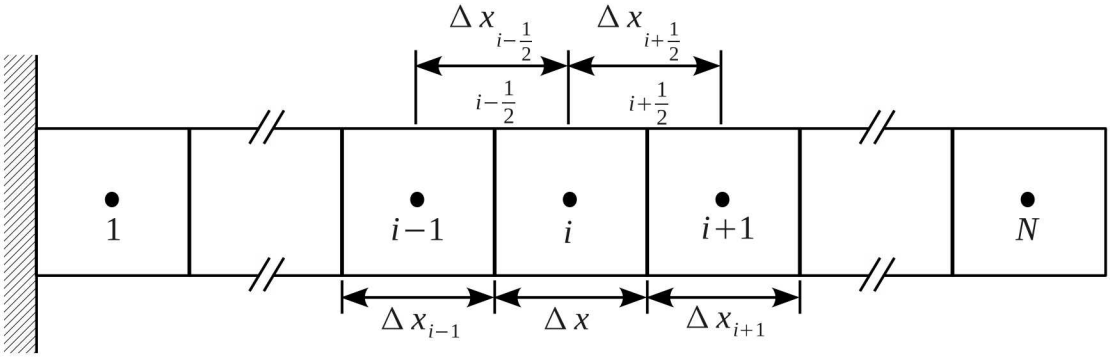


**Figure 3.5-2.. Mesh index definition for 1-D composite fire boundary condition**

The species transport equations, Equation 3.2, undergoes an identical transformation,

$$\int_{\Delta t} \left[ \int_V \frac{\partial \bar{\rho} Y_k}{\partial t} dV - \int_V \dot{\omega}_k''' dV \right] dt = 0 \tag{3.16}$$

$$\int_{\Delta t} \left[ V_i \frac{\partial \bar{\rho}_i Y_k}{\partial t} - V_i \dot{\omega}_{c,i}''' \right] dt = 0. \tag{3.17}$$

$$V_i \left( \bar{\rho}_i^{n+1} Y_{k,i}^{n+1} - \bar{\rho}_i^{n} Y_{k,i}^{n} \right) - V_i \dot{\omega}_{k,i}''' \Delta t = 0 \tag{3.18}$$

$$Y_{k,i}^{n+1} = \frac{\bar{\rho}_i^{n} Y_{k,i}^{n} + \dot{\omega}_{k,i}''' \Delta t}{\bar{\rho}_i^{n+1}}, \tag{3.19}$$

where the bulk density at the new time level is used from Equation 3.15, and the source term is evaluated from the most recent estimate of the $n+1$ solution state.

The energy equation also undergoes a similar transformation, but with added complexity due to the inclusion of spatial derivatives. Equation 3.3 is first integrated in both space and time, and the Gauss divergence theorem is used to remove one level of spatial derivatives in the diffusive flux term,

$$\int_{\Delta t} \left[ \int_V \bar{\rho} \bar{c} \frac{\partial T}{\partial t} dV - \int_V \frac{\partial}{\partial x} \left( \bar{k} \frac{\partial T}{\partial x} \right) dV - \int_V \dot{q}''' dV \right] dt = 0 \tag{3.20}$$

$$\int_{\Delta t} \left[ \int_V \bar{\rho} \bar{c} \frac{\partial T}{\partial t} dV - \int_A n \cdot \left( \bar{k} \frac{\partial T}{\partial x} \right) dA - \int_V \dot{q}''' dV \right] dt = 0. \tag{3.21}$$

Integrating numerically in space yields

$$\int_{\Delta t}\left[\bar{\rho}_i\bar{c}_i V_i\frac{\partial T_i}{\partial t}-\left(\left(-A\bar{k}\frac{\partial T}{\partial x}\right)_{i-\frac{1}{2}}+\left(A\bar{k}\frac{\partial T}{\partial x}\right)_{i+\frac{1}{2}}\right)-\dot{q}_i'''V_i\right]dt=0 \tag{3.22}$$

$$\int_{\Delta t}\left[\bar{\rho}_i\bar{c}_i V_i\frac{\partial T_i}{\partial t}+A_{i-\frac{1}{2}}\bar{k}_{i-\frac{1}{2}}\left(\frac{T_i-T_{i-1}}{\Delta x_{i-\frac{1}{2}}}\right)-A_{i+\frac{1}{2}}\bar{k}_{i+\frac{1}{2}}\left(\frac{T_{i+1}-T_i}{\Delta x_{i+\frac{1}{2}}}\right)-\dot{q}_i'''V_i\right]dt=0, \tag{3.23}$$

and then integrating in time and linearizing the equation by evaluating the coefficients at the most recent estimate of the $n+1$ solution state yields

$$\bar{\rho}_i\bar{c}_i V_i\left(\frac{T_i^{n+1}-T_i^n}{\Delta t}\right)+A_{i-\frac{1}{2}}\bar{k}_{i-\frac{1}{2}}\left(\frac{T_i^{n+1}-T_{i-1}^{n+1}}{\Delta x_{i-\frac{1}{2}}}\right)-A_{i+\frac{1}{2}}\bar{k}_{i+\frac{1}{2}}\left(\frac{T_{i+1}^{n+1}-T_i^{n+1}}{\Delta x_{i+\frac{1}{2}}}\right)-\dot{q}_i'''V_i=0. \tag{3.24}$$

This leads to a tridiagonal system of coupled linear equations for the temperature at time level $(n+1)$, which is solved using a direct method with the DGTSL module of the SLATEC library.

The continuity, species, and energy equations are solved sequentially in the order described, and the solution is repeated until the maximum normalized change in the temperature solution,

$$T_{\text{err}}=\frac{|T^{n+1}-T^*|}{T^{n+1}} \tag{3.25}$$

satisfies the user-specified tolerance, where $T^*$ is the solution from the previous iteration.

Please see the Fuego user's manual for details on the usage of this boundary condition.

#### 3.5.2.5. Numerical Implementation - New

When using the new form of the composite BC, where the chemical mechanism is specified using a chemistry description, the numerical implementation is slightly different. The finite volume discretization used is the same, but the system of equations is solved monolithically using the user-specified ODE solver. The solver handles time stepping during the sub-integration to reduce the overall error below the specified threshhold.

Additionally, when constructing the monolithic system with the new form the DOFs are temperature and $N$ species masses, rather than the prior approach of using temperature, density, and $N-1$ species mass fractions.

# 3.6. NON-CONFORMAL DG BOUNDARY CONDITION

## 3.6.1. Conceptual Overview

The non-conformal boundary condition uses the DG approach described by Domino [87] and is currently implemented for turbulence models, continuity, momentum, and heat conduction. The non-conformal boundary condition is applied where you have two domains, $A$ and $B$, which share a discontinuous interface with individual sidesets, $S_A$ and $S_B$. The algorithm is applied in two passes, first iterating over all integration points in $S_A$ and finding the matching face in $S_B$, then by iterating over the integration points in $S_B$ and finding the matching face in $S_A$. The generic flux of a scalar, $\phi$ at an integration point on $S_A$ is

$$\hat{Q}^A = \left[\frac{(q_j^A n_j^A - q_j^B n_j^B)}{2} + \lambda^A(\phi^A - \phi^B)\right] A_f^A + \dot{m}^A \frac{(\phi^A + \phi^B)}{2} + \eta \frac{|\dot{m}^A|}{2}(\phi^A - \phi^B), \quad (3.26)$$

where $q_j$ is the diffusive flux, $\dot{m}$ is the mass flux, and $\lambda$ is the interior penalty coefficient.

Prior investigations have shown that pressure oscillations can be minimized by using the current integration point normal direction for both diffusive fluxes, so $n_j^B = -n_j^A$.

The penalty term, $\lambda^A$ is given by

$$\lambda^A = \frac{(\Gamma^A/L^A + \Gamma^B/L^B)}{2}, \quad (3.27)$$

where $\Gamma$ is the diffusive flux coefficient and $L$ is an element length scale.

The advection coefficient, $\eta$, defines the degree of upwinding to use. A value of $\eta = 1$ results in a fully upwind scheme, while $\eta = 0$ results in a central difference scheme. When using a hybrid approach, this value is calculated locally based on the Peclet number.

### *3.6.1.1. Continuity*

The mass flow rate at the non-conformal boundary includes the pressure stabilization terms, as

$$\dot{m}^A = \left[\frac{(\rho u_j^A + a_p^A G_j^A p - a_p^A \frac{\partial p^A}{\partial x_j})n_j^A - (\rho u_j^B + a_p^B G_j^B p - a_p^B \frac{\partial p^B}{\partial x_j})n_j^B}{2} + \lambda^A(p^A - p^B)\right] A_f^A. \quad (3.28)$$

### 3.6.2. Performance Considerations

There is a computational cost associated with the use of the non-conformal interface. This is largely due to two tasks: a search to match integration points with opposing faces on both non-conformal boundaries, and the resulting changes to the linear system stencil if the interface moves. Preliminary testing has shown that the cost of reinitializing the linear system with a new stencil is at least an order of magnitude greater than the cost of the search. For this reason, the algorithm implemented in Fuego will do an extra search in order to only reinitialize the linear system when the stencil actually changes. The user can expand the search boxes used in the stencil definition in order to reduce the number of linear system reinitializations by setting the "Search Expansion Factor" in the non-conformal boundary condition specification. This number is the approximate diametrical size increase in the stencil in terms of number of elements.

### 3.6.3. Non-Conformal Moving Walls

The non-conformal BC in Fuego supports having a moving surface as one side of the interface. This is implemented by providing a shell block containing nodal displacements on one side of the interface. Then non-conformal search will search for matching faces in opposing blocks and shells. If a shell is found, an appropriate wall boundary condition is applied at the integration point, while if a non-shell element is found the interior algorithms described previously are used.

# 3.7. POROUS-FLUID COUPLING ALGORITHM

This section provides a brief overview of the current porous/fluid coupling algorithm, as it is intended to be used in simulations of composite fires using Fuego to model the fluid region and coupling to Aria to model the porous region.. This is a loosely-coupled algorithm, relying on framework interpolation transfers of nodal fields between the porous region and the low-Mach fluid region and region-region Picard loops to converge the overall problem within a timestep.

Note that the shorthand is adopted where the porous region is described as region $A$ and the low-Mach free fluid region is described as region $B$, with the interface between them referred to as $\Gamma_{AB}$ and other boundaries not a part of this interface are referred to as $\Gamma \backslash \Gamma_{AB}$.

## 3.7.1. Fluid Flow

### *3.7.1.1. Bulk Equations*

**Porous Continuity Equation** The porous region contains a condensed phase (the solid skeleton of the porous system) and a gas phase occupying the pores of the condensed phase. The condensed phase is not discussed explicitly in this description, although it interacts with the gas phase through things like its permeability and porosity, and its decomposition which can produce gas-phase mass through chemical source terms.

The porous gas-phase continuity equation within a porous region, to be solved for the gas-phase pressure $p_g$, is

$$\frac{\partial(\bar{\psi}\rho_g)}{\partial t} + \frac{\partial(\rho_g u_{j,g})}{\partial x_j} = \dot{\omega}'''_{fg}, \tag{3.29}$$

where $\bar{\psi}$ is the mixture-averaged condensed-phase porosity, $\rho_g$ is the gas-phase density, and $u_{j,g}$ is the gas-phase velocity vector computed from Darcy's approximation as

$$u_{j,g} = -\frac{\bar{K}}{\mu_g}\left(\frac{\partial p_g}{\partial x_j} + \rho_g g_j\right), \tag{3.30}$$

where $\bar{K}$ is the mixture-averaged condensed-phase permeability, $\mu_g$ is the gas-phase viscosity, and $g_j$ is the gravity vector. The term $\dot{\omega}'''_{fg}$ represents the formation rate of gas-phase mass from the condensed phase.

Multiplying Equation 3.29 by an arbitrary test function $w$ and integrating over the domain $\Omega$ while integrating the advection term by parts, yields the variational form of the continuity equation that is solved for $p_g$ using the Galerkin finite element method,

$$\int_\Omega w\left(\frac{\partial(\bar{\psi}\rho_g)}{\partial t} - \dot{\omega}'''_{fg}\right)d\Omega - \int_\Omega \frac{\partial w}{\partial x_j}\rho_g u_{j,g} d\Omega + \int_\Gamma w\rho_g u_{j,g} n_j d\Gamma = 0, \tag{3.31}$$

where $n_j$ is the boundary surface normal. The boundary flux term is then split into contributions on the interface between regions $A$ and $B$ and off the interface so that they may be treated separately. The continuity equation then takes the form

$$\int_{\Omega} w \left( \frac{\partial(\bar{\psi}\rho_g)}{\partial t} - \dot{\omega}'''_{fg} \right) d\Omega - \int_{\Omega} \frac{\partial w}{\partial x_j} \rho_g u_{j,g} d\Omega + \int_{\Gamma \backslash \Gamma_{AB}} w \rho_g u_{j,g} n_j d\Gamma + \int_{\Gamma_{AB}} w F_A d\Gamma = 0, \tag{3.32}$$

where $F_A$ is the imposed flux on the porous side ($A$) of the $\Gamma_{AB}$ interface. A detailed description of the coupling boundary flux is given in Section 3.7.1.2

**Low-Mach Continuity Equation** The continuity equation within the low-Mach fluid region, to be solved for the pressure $p$, is

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_j}{\partial x_j} = S, \tag{3.33}$$

where $\rho$ is the fluid density, $u_j$ is the fluid velocity, and $S$ is a generic mass volumetric source term. Integrating Equation 3.33 over a CVFEM control volume and using the Gauss divergence theorem on the advection and diffusive flux terms, yields the integral form of the continuity equation that is solved,

$$\int_{\Omega} \left( \frac{\partial \rho}{\partial t} - S \right) d\Omega + \int_{\Gamma} \rho u_j n_j d\Gamma = 0. \tag{3.34}$$

Similar to the porous continuity equation, the boundary flux term is split into contributions both on and off the $\Gamma_{AB}$ interface, yielding

$$\int_{\Omega} \left( \frac{\partial \rho}{\partial t} - S \right) d\Omega + \int_{\Gamma \backslash \Gamma_{AB}} \rho u_j n_j d\Gamma + \int_{\Gamma_{AB}} F_B d\Gamma = 0. \tag{3.35}$$

The interface coupling flux is described in Section 3.7.1.2.

**Low-Mach Momentum Equation** The momentum equation within the low-Mach fluid region, to be solved for the velocity $u_i$, is

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_j u_i}{\partial x_j} = \frac{\partial \sigma_{ij}}{\partial x_j} + \rho g_i \tag{3.36}$$

where the Cauchy stress tensor is given by

$$\sigma_{ij} = \tau_{ij} - p\delta_{ij} \tag{3.37}$$

in terms of the viscous stress tensor

$$\tau_{ij} = \mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \mu \frac{\partial u_k}{\partial x_k} \delta_{ij}. \tag{3.38}$$

Integrating Equation 3.36 over a CVFEM control volume and using the Gauss divergence theorem on the advection and diffusive flux terms, yields the integral form of the momentum equation that is solved,

$$\int_{\Omega} \frac{\partial \rho u_i}{\partial t} d\Omega + \int_{\Gamma} \rho u_j u_i n_j d\Gamma - \int_{\Gamma} \sigma_{ij} n_j d\Gamma - \int_{\Omega} \rho g_i d\Omega = 0. \tag{3.39}$$

Multiplying this equation by an arbitrary test function $w$, integrating the advection and stress terms by parts, and splitting the boundary flux terms into on-interface and off-interface contributions yields

$$\begin{aligned} \int_{\Omega} \frac{\partial \rho u_i}{\partial t} d\Omega - \int_{\Omega} \rho g_i d\Omega + \int_{\Gamma \backslash \Gamma_{AB}} \rho u_j u_i n_j d\Gamma - \int_{\Gamma \backslash \Gamma_{AB}} \sigma_{ij} n_j d\Gamma \\ + \int_{\Gamma_{AB}} \rho u_j u_i n_j d\Gamma - \int_{\Gamma_{AB}} \sigma_{ij} n_j d\Gamma = 0. \end{aligned} \tag{3.40}$$

#### 3.7.1.2. Coupling Boundary Conditions

Coupling between the porous and fluid regions is achieved using an interface flux that is imposed as a Robin-style boundary condition. This approach has been used successfully in the past for coupling incompressible Darcy and Stokes flows [88]. Here we generalize the coupling for compressible fluids and Navier-Stokes flow.

The fluxes applied to the porous and fluid continuity equations at the interface $\Gamma_{AB}$ are

$$F_A = \dot{m}_B \cdot \hat{n} + \beta(p_A - p_B) \tag{3.41}$$

$$F_B = \dot{m}_A \cdot \hat{n} + \beta(p_B - p_A), \tag{3.42}$$

where $\dot{m}_A = \rho_g \vec{u_g}$, $\dot{m}_B = \rho \vec{u}$, and the free constant $\beta$ is computed as

$$\beta = c \frac{\bar{K} \rho_g}{\mu_g h} \tag{3.43}$$

with $h$ being a measure of the mesh size adjacent to the interface, and $c$ a user-specified scaling coefficient. The same value of $\beta$ is used on both sides of the interface because that results in excellent mass conservation even on coarse meshes. If a different value of $\beta$ is used on each side the method is still convergent but worse mass conservation is observed when solving on under-resolved meshes. Some attempts have been made to use an averaged penalty coefficient of the form

$$\beta = \frac{\beta_A + \beta_B}{2} \tag{3.44}$$

$$\beta_A = \frac{\bar{K} \rho_g}{\mu_g h} \tag{3.45}$$

$$\beta_B = \frac{\tau^c}{h}, \text{ or } \beta_B = \frac{\mu}{h P_{ref}}, \tag{3.46}$$

however they resulted in an impractically large number of Picard iterations to converge for some test problems.

A distinguishing condition BC for velocity is applied to the low-mach momentum equation in the form

$$u_j - (u_{j,n}^D + u_{j,t}^D) = 0, \tag{3.47}$$

where $u_{j,n}^D$ is the imposed normal component of velocity and $u_{j,t}^D$ is the imposed tangential component of velocity. The normal component is computed directly from the continuity flux at the interface,

$$u_{j,n}^D = \frac{F_B}{\rho} n_j. \tag{3.48}$$

The tangential component is based on a variation of the classical Beavers-Joseph-Saffman condition [89, 90] for the slip velocity which has been extended to non-planar surfaces in multidimensional flow [91], which defines a provisional model velocity

$$u_j^{\mathrm{BJS}} = -\frac{\sqrt{\bar{K}}}{\alpha\mu} \left(n_i \tau_{ij}\right) \tag{3.49}$$

where $\bar{K}$ is the permeability of the porous region at the interface, $\mu$ is the viscosity of the local fluid at the interface, $\tau_{ij}$ is the viscous stress tensor of the fluid at the interface, and $\alpha$ is a dimensionless model parameter that is a function of the microstructure of the porous material, which has been found to have typical values near $0.1$ [90]. The tangential component of this vector quantity is used as the tangential component of the distinguishing condition velocity, and is computed as

$$u_{j,t}^D = u_j^{\mathrm{BJS}} - \left(u_k^{\mathrm{BJS}} n_k\right) n_j. \tag{3.50}$$

## 3.7.2. Enthalpy Transport

### *3.7.2.1. Bulk Equations*

**Porous Gas-Phase Enthalpy Equation** The gas-phase enthalpy equation within a porous region, to be solved for the gas-phase temperature $T_g$, is

$$\begin{aligned}\frac{\partial(\bar{\psi}\rho_g h_g)}{\partial t} + \frac{\partial(\rho_g u_{j,g} h_g)}{\partial x_j} = -\frac{\partial q_j^{h,g}}{\partial x_j} + \left(\frac{\partial(\bar{\psi} p_g)}{\partial t} + u_{j,g}\frac{\partial p_g}{\partial x_j}\right) \\ + h_{cv}\left(\bar{T} - T_g\right) + \sum_k \left(\dot{\omega}_{s,fk}''' - \dot{\omega}_{s,dk}'''\right) h_{k,g}\end{aligned} \tag{3.51}$$

where $h_g$ is the mixture-averaged gas-phase enthalpy, $h_{cv}$ is the volumetric heat transfer coefficient, $\bar{T}$ is the porous condensed-phase temperature, $\left(\dot{\omega}_{s,fk}''' - \dot{\omega}_{s,dk}'''\right)$ is the formation and destruction of gas-phase species due to heterogeneous reactions, and $h_{k,g}$ is the gas-phase enthalpy of chemical species $k$. The gas-phase energy diffusive flux vector $q_j^{h,g}$ is modeled as

$$q_j^{h,g} = -\bar{\psi}\rho_g D_g \frac{\partial h_g}{\partial x_j}, \tag{3.52}$$

where $D_g$ is the mixture-averaged gas-phase mass diffusivity.

Note that, in Equation 3.51, there is some concern that the pressure spatial derivative term, $u_{j,g}\frac{\partial p_g}{\partial x_j}$, is incorrect. A crude re-derivation of this equation indicates that its form should instead be $\frac{u_{j,g}}{\bar{\psi}}\frac{\partial(\bar{\psi}p_g)}{\partial x_j}$. A more formal re-derivation from first principles is required to decide conclusively on the correct form of this term, so it is left in its current form for now. Additionally, the diffusive flux vector is also of concern since the current form was derived under the assumption of constant specific heat, equal species mass diffusivities, and unity Lewis number. These assumptions may not be valid in future simulations, meaning that this term should possibly be returned to the standard Fick's law version that includes a contribution due to enthalpy transport by differential diffusion of chemical species. Again, this term is left in its current form for the present work.

Multiplying Equation 3.51 by an arbitrary test function $w$ and integrating over the domain $\Omega$ while integrating the advection and diffusion terms by parts, yields the variational form of the enthalpy equation that is solved for $h_g$ using the Galerkin finite element method,

$$\int_\Omega w\left(\frac{\partial(\bar{\psi}\rho_g h_g)}{\partial t}-\left(\frac{\partial(\bar{\psi}p_g)}{\partial t}+u_{j,g}\frac{\partial p_g}{\partial x_j}\right)-h_{cv}\left(\bar{T}-T_g\right)-\sum_k\left(\dot{\omega}'''_{s,fk}-\dot{\omega}'''_{s,dk}\right)h_{g,k}\right)d\Omega$$
$$-\int_\Omega\frac{\partial w}{\partial x_j}\left(\rho_g u_{j,g}h_g\right)d\Omega+\int_\Gamma w\left(\rho_g u_{j,g}h_g\right)n_j d\Gamma$$
$$-\int_\Omega\frac{\partial w}{\partial x_j}q_j^{h,g}d\Omega+\int_\Gamma w q_j^{h,g}n_j d\Gamma=0. \quad (3.53)$$

The boundary flux terms are then split into contributions on the interface between regions $A$ and $B$ and off the interface so that they may be treated separately. The enthalpy equation then takes the form

$$\int_\Omega w\left(\frac{\partial(\bar{\psi}\rho_g h_g)}{\partial t}-\left(\frac{\partial(\bar{\psi}p_g)}{\partial t}+u_{j,g}\frac{\partial p_g}{\partial x_j}\right)-h_{cv}\left(\bar{T}-T_g\right)-\sum_k\left(\dot{\omega}'''_{s,fk}-\dot{\omega}'''_{s,dk}\right)h_{g,k}\right)d\Omega$$
$$-\int_\Omega\frac{\partial w}{\partial x_j}\left(\rho_g u_{j,g}h_g\right)d\Omega-\int_\Omega\frac{\partial w}{\partial x_j}q_j^{h,g}d\Omega$$
$$+\int_{\Gamma\backslash\Gamma_{AB}} w\left(\rho_g u_{j,g}h_g\right)n_j d\Gamma+\int_{\Gamma\backslash\Gamma_{AB}} w q_j^{h,g}n_j d\Gamma+\int_{\Gamma_{AB}} w J_A^H d\Gamma=0. \quad (3.54)$$

where $J_A^H$ is the imposed flux on the porous side ($A$) of the $\Gamma_{AB}$ interface. A detailed description of the coupling boundary flux is given in Section 3.7.2.2.

**Low-Mach Enthalpy Equation** The enthalpy equation within the low-Mach fluid region, to be solved for the fluid temperature $T$, is

$$\frac{\partial(\rho h)}{\partial t}+\frac{\partial(\rho u_j h)}{\partial x_j}=-\frac{\partial q_j^h}{\partial x_j}-\frac{\partial q_j^r}{\partial x_j}+\left(\frac{\partial p}{\partial t}+u_j\frac{\partial p}{\partial x_j}\right)+\tau_{ij}\frac{\partial u_i}{\partial x_j} \quad (3.55)$$

where $h$ is the mixture-averaged fluid enthalpy, $q_j^r$ is a source term due to radiation absorption and emission, and $p$ is the fluid pressure. The diffusive flux vector is given by

$$q_j^h = -\lambda \frac{\partial T}{\partial x_j} + \sum_k \rho h_k Y_k \hat{u}_{jk}, \tag{3.56}$$

where $\lambda$ is the mixture thermal conductivity, $h_k$ is the enthalpy of species $k$, $Y_k$ is the mass fraction of species $k$, and $\hat{u}_{jk}$ is the diffusion velocity of species $k$ in the $j$ direction.

Integrating Equation 3.55 over a CVFEM control volume and using the Gauss divergence theorem on the advective and diffusive flux terms, yields the integral form of the enthalpy equation to be solved,

$$\int_\Omega \frac{\partial(\rho h)}{\partial t} d\Omega + \int_\Omega \frac{\partial q_j^r}{\partial x_j} d\Omega - \int_\Omega \left( \frac{\partial p}{\partial t} + u_j \frac{\partial p}{\partial x_j} \right) d\Omega - \int_\Omega \tau_{ij} \frac{\partial u_i}{\partial x_j} d\Omega + \int_\Gamma (\rho u_j h)\, n_j d\Gamma + \int_\Gamma q_j^h n_j d\Gamma = 0. \tag{3.57}$$

The boundary flux terms are then split into contributions on the interface between regions $A$ and $B$ and off the interface so that they may be treated separately. The enthalpy equation then takes the form

$$\int_\Omega \frac{\partial(\rho h)}{\partial t} d\Omega + \int_\Omega \frac{\partial q_j^r}{\partial x_j} d\Omega - \int_\Omega \left( \frac{\partial p}{\partial t} + u_j \frac{\partial p}{\partial x_j} \right) d\Omega - \int_\Omega \tau_{ij} \frac{\partial u_i}{\partial x_j} d\Omega + \int_{\Gamma \setminus \Gamma_{AB}} (\rho u_j h)\, n_j d\Gamma + \int_{\Gamma \setminus \Gamma_{AB}} q_j^h n_j d\Gamma + \int_{\Gamma_{AB}} w J_B^h d\Gamma = 0, \tag{3.58}$$

where $J_B^h$ is the imposed flux on the fluid side ($B$) of the $\Gamma_{AB}$ interface. A detailed description of the coupling boundary condition is given in Section 3.7.2.2.

#### 3.7.2.2. Coupling Boundary Conditions

Coupling enthalpy transport between the porous and fluid regions is complicated by the use of a two temperature model in the porous region.

To resolve this complication the energy flux applied to the fluid region has a diffusive/conductive component from the gas phase in the porous region, an advective component from the gas phase in the porous region, a convective component from the condensed phase in the porous region, and a penalty coefficient to enforce temperature continuity between the porous gas phase and the fluid. This takes the form

$$J_B^h = J_{A,g}^{\text{diff}} + J_{A,g}^{\text{adv}} + J_{A,c}^{\text{conv}} + \overline{\left(\frac{\lambda}{h}\right)}(T_f - T_g), \tag{3.59}$$

where $J_{A,g}^{\text{diff}}$ is the diffusive energy transport from the porous gas phase, $J_{A,g}^{\text{adv}}$ is the advective energy transport from the porous gas phase, $J_{A,c}^{\text{conv}}$ is the convective energy transport from the porous condensed phase, and $\overline{\left(\frac{\lambda}{h}\right)}$ is the averaged thermal conductivity / mesh size between the porous and

fluid regions. As with the flow coupling boundary conditions this same penalty coefficient is used in both regions to get the best energy conservation on coarse meshes.

The advective energy transport component takes the form

$$J_{A,g}^{\text{adv}} = F_B h_{AB}, \tag{3.60}$$

where $h_{AB}$ is the upwinded interface enthalpy (i.e. it is either $h_A$ or $h_B$ depending on the direction of $F_B$). The convective component from the condensed phase has the form

$$J_{A,c}^{\text{conv}} = (1 - \bar{\psi})\frac{h_{cv}}{a_s}(T_f - T_c), \tag{3.61}$$

where $h_{cv}$ is the volumetric heat transfer coefficient of the porous region and $a_s$ is the specific surface area ($m^2/m^3$) of the porous medium. This formulation of the convective component assumes that the convective heat transfer between the condensed phase and the free fluid is consistent with the convective heat transfer in the bulk of the porous medium that results in the volumetric heat transfer term of the bulk equations.

The coupling back to the porous region is derived based on the assumption that

$$J_{A,g}^{h} + J_{A,c}^{h} = J_{B}^{\text{diff}} + J_{B}^{\text{adv}}, \tag{3.62}$$

that is, the fluid region applies advective and diffusive energy transport components to the porous region as a whole. The flux applied to the condensed phase is assumed to be the same as the convective flux component it applies to the free fluid,

$$J_{A,c}^{h} = (1 - \bar{\psi})\frac{h_{cv}}{a_s}(T_c - T_f). \tag{3.63}$$

The flux applied to the porous gas phase is then given by

$$J_{A,g}^{h} = J_{B}^{\text{diff}} + J_{B}^{\text{adv}} - J_{A,c}^{h} + \overline{\left(\frac{\lambda}{h}\right)}(T_g - T_f), \tag{3.64}$$

where the advective component is computed in the same manner as is done for the advective flux applied to the fluid region,

$$J_{B}^{\text{adv}} = F_A h_{AB}. \tag{3.65}$$

## 3.7.3. Species Transport

### *3.7.3.1. Bulk Equations*

**Porous Gas-Phase Species Equation** The gas-phase species equation within a porous region, to be solved for the gas-phase mass fraction $Y_{k,g}$ of species $k$, is

$$\frac{\partial(\bar{\psi}\rho_g Y_{k,g})}{\partial t} + \frac{\partial(\rho_g u_{j,g} Y_{k,g})}{\partial x_j} = -\frac{\partial q_{kj}^{Y,g}}{\partial x_j} + \left(\dot{\omega}_{s,fk}''' - \dot{\omega}_{s,dk}'''\right) + \left(\dot{\omega}_{g,fk}''' - \dot{\omega}_{g,dk}'''\right) \tag{3.66}$$

where $\left(\dot{\omega}'''_{s,fk} - \dot{\omega}'''_{s,dk}\right)$ is the formation and destruction of gas-phase species due to heterogeneous reactions, and $\left(\dot{\omega}'''_{g,fk} - \dot{\omega}'''_{g,dk}\right)$ is the formation and destruction of gas-phase species due to homogeneous reactions. The gas-phase species diffusion flux vector $q^{Y,g}_{kj}$ is modeled as

$$q^{Y,g}_{kj} = -\bar{\psi}\rho_g D_{k,g}\frac{\partial Y_{k,g}}{\partial x_j}, \tag{3.67}$$

where $D_{k,g}$ is the gas-phase mass diffusivity for species $k$. Note that if the mass diffusivities are not equal for all species, then an additional correction is required to maintain mass conservation.

Multiplying Equation 3.66 by an arbitrary test function $w$ and integrating over the domain $\Omega$ while integrating the advection and diffusion terms by parts, yields the variational form of the species equation that is solved for $Y_{k,g}$ using the Galerkin finite element method,

$$\begin{aligned}\int_\Omega w\left(\frac{\partial(\bar{\psi}\rho_g Y_{k,g})}{\partial t} - \left(\dot{\omega}'''_{s,fk} - \dot{\omega}'''_{s,dk}\right) - \left(\dot{\omega}'''_{g,fk} - \dot{\omega}'''_{g,dk}\right)\right) d\Omega \\ - \int_\Omega \frac{\partial w}{\partial x_j}\left(\rho_g u_{j,g} Y_{k,g}\right) d\Omega + \int_\Gamma w\left(\rho_g u_{j,g} Y_{k,g}\right) n_j d\Gamma \\ - \int_\Omega \frac{\partial w}{\partial x_j} q^{Y,g}_{kj} d\Omega + \int_\Gamma w q^{Y,g}_{kj} n_j d\Gamma = 0.\end{aligned} \tag{3.68}$$

The boundary flux terms are then split into contributions on the interface between regions $A$ and $B$ and off the interface so that they may be treated separately. The species equation then takes the form

$$\begin{aligned}\int_\Omega w\left(\frac{\partial(\bar{\psi}\rho_g Y_{k,g})}{\partial t} - \left(\dot{\omega}'''_{s,fk} - \dot{\omega}'''_{s,dk}\right) - \left(\dot{\omega}'''_{g,fk} - \dot{\omega}'''_{g,dk}\right)\right) d\Omega \\ - \int_\Omega \frac{\partial w}{\partial x_j}\left(\rho_g u_{j,g} Y_{k,g}\right) d\Omega - \int_\Omega \frac{\partial w}{\partial x_j} q^{Y,g}_{kj} d\Omega \\ + \int_{\Gamma\setminus\Gamma_{AB}} w\left(\rho_g u_{j,g} Y_{k,g}\right) n_j d\Gamma + \int_{\Gamma\setminus\Gamma_{AB}} w q^{Y,g}_{kj} n_j d\Gamma + \int_{\Gamma_{AB}} w J^{Y_k}_A d\Gamma = 0.\end{aligned} \tag{3.69}$$

where $J^{Y_k}_A$ is the imposed flux on the porous side ($A$) of the $\Gamma_{AB}$ interface. A detailed description of the coupling boundary flux is given in Section 3.7.3.2.

**Low-Mach Species Equation** The species equation within the low-Mach fluid region, to be solved for the mass fraction $Y_k$ for species $k$, is

$$\frac{\partial(\rho Y_k)}{\partial t} + \frac{\partial(\rho u_j Y_k)}{\partial x_j} = -\frac{\partial q^Y_{kj}}{\partial x_j} + \dot{\omega}'''_k \tag{3.70}$$

where $\dot{\omega}'''_k$ is the volumetric mass formation rate if species $Y_k$, and the diffusive flux vector is given by

$$q^Y_{kj} = -\rho\hat{u}_{j,k} Y_k, \tag{3.71}$$

with $\hat{u}_{j,k}$ being the species diffusion velocity. Several forms for this velocity are possible, with the simplest being

$$\hat{u}_{j,k} = -D\frac{1}{Y_k}\frac{\partial Y_k}{\partial x_j} \tag{3.72}$$

for equal mass diffusivities $D$ for all species. A more complex form is needed for unequal mass diffusivities, which is not presented here.

Integrating Equation 3.70 over a CVFEM control volume and using the Gauss divergence theorem on the advective and diffusive flux terms yields the integral form of the species equation that is solved,

$$\int_\Omega \frac{\partial(\rho Y_k)}{\partial t} d\Omega - \int_\Omega \dot{\omega}_k''' d\Omega + \int_\Gamma (\rho u_j Y_k)\, n_j d\Gamma + \int_\Gamma q_{kj}^Y n_j d\Gamma = 0. \tag{3.73}$$

The boundary flux terms are then split into contributions on the interface between regions $A$ and $B$ and off the interface so that they may be treated separately. The species equation then takes the form

$$\int_\Omega \frac{\partial(\rho Y_k)}{\partial t} d\Omega - \int_\Omega \dot{\omega}_k''' d\Omega + \int_{\Gamma\backslash\Gamma_{AB}} (\rho u_j Y_k)\, n_j d\Gamma + \int_{\Gamma\backslash\Gamma_{AB}} q_{kj}^Y n_j d\Gamma + \int_{\Gamma_{AB}} w J_B^{Y_k} d\Gamma = 0. \tag{3.74}$$

where $J_B^{Y_k}$ is the imposed flux on the fluid side ($B$) of the $\Gamma_{AB}$ interface. A detailed description of the coupling boundary condition is given in Section 3.7.3.2.

#### 3.7.3.2. Coupling Boundary Conditions

Coupling species transport across the porous-fluid interface is relatively simple compared to enthalpy transport. As with the flow coupling Robin style boundary conditions are applied on both the porous and fluid regions, but with both diffusive and advective flux components.

For the flux of a species $k$ this takes the form:

$$J_A^{Y_k} = J_B^{\text{diff}} + F_A \rho_g Y_{k,AB} + \overline{\left(\frac{D_k \rho}{h}\right)}(Y_{k,A} - Y_{k,B}) \tag{3.75}$$

$$J_B^{Y_k} = J_A^{\text{diff}} + F_B \rho Y_{k,AB} + \overline{\left(\frac{D_k \rho}{h}\right)}(Y_{k,B} - Y_{k,A}) \tag{3.76}$$

where $Y_{k,AB}$ is the upwinded interface mass fraction, equivalent to $h_{AB}$ from the enthalpy coupling. Once again the same penalty coefficient is used on each side in order to get good mass conservation even on coarse meshes.

# 4. PARTICLES

## 4.1. INTRODUCTION

The transport of particles through a gas-phase flow is of importance to a tremendous range of applications. Applications in the area of combustion and fire science include fuel sprays, suppressant transport and metal particle combustion [92, 93, 94, 95, 96, 97, 98, 99]. These applications typically have a strong coupling between the heat and mass transfer. For example, fuel spray combustion is typically limited by the diffusion of the oxidizer towards the particle. In fire suppressant distribution, the cooling associated with the evaporating suppressant can dramatically slow suppressant evaporation. In metal particle combustion, in order for the metal oxide combustion product to condense out, the enthalpy of condensation must be dissipated; this energy dissipation is a combination of radiative and conductive transport, each of which results in differing heat flux consequences. Also relevant are the transport of contaminants through the atmosphere and the dynamics of clouds [100]. A large number of industrial processes share similar physics including powder manufacturing, painting, coating and ink-jet printing.

This report describes the development of a Lagrangian particle and droplet transport model and its integration with a computational fluid dynamics (CFD) code that solves, on an Eulerian mesh, the continuum phase. Conservation of mass, momentum and energy are considered for the coupled system allowing combustion along with evaporating and condensing particles. Since examples of this type of flow are typically sprays, this model is sometimes referred to as a spray model, but it can handle general classes of particulate flows. This model is developed to be suitable for modeling evaporating, condensing or combusting flows of particles in continuum gas-phase flows. This model is based partly on the initial implementation of a dilute spray model in the Vulcan fire-physics computational modeling code [101, 102] as described in [103].

Two significant limitations are stipulated that lead to the simplified conservation equations employed. First, the spray must be dilute, that is the volume fraction of the particle phase must be small (i.e. less than 10 percent). Second, the physical density of the particle should be orders of magnitude greater than the continuum (gas) phase and the particle Reynolds numbers should not be too large or additional terms will appear in the particle evolution equations [104].

### 4.1.1. The Spray Equation

For given physical properties of the particle (composition, density, etc.), the particle field is characterized by the particle locations, velocities, radii and temperatures. This can be expressed in terms of a particle

distribution function, f, so that:

$$f(\mathbf{x}_p, \mathbf{u}_p, r_p, T_p; t) d\mathbf{u}_p dr_p dT_p \tag{4.1}$$

is the probable number of droplets per unit volume at location $\mathbf{x}_p$ in the velocity range $(\mathbf{u}_p, \mathbf{u}_p + d\mathbf{u}_p)$, the size range $(r_p, r_p + dr_p)$, and the temperature range $(T_p, T_p + dT_p)$. The evolution of this particle distribution can be described by an equation of the Fokker-Planck form [105]:

$$\frac{df}{dt} + \nabla \cdot (\mathbf{u}_p f) + \nabla_u \cdot \left( \frac{d\mathbf{u}_p}{dt} f \right) + \frac{d}{dr_p} \cdot \left( \frac{dr_p}{dt} f \right) + \frac{d}{dT_p} \cdot \left( \frac{dT_p}{dt} f \right) = \left( \frac{df}{dt} \right)_{coll} + \left( \frac{df}{dt} \right)_{brk} . \tag{4.2}$$

Here, expressions for the particle acceleration, evaporation and heating are required in the third through fifth terms on the left-hand side. Similar models for particle collision and breakup appear on the right-hand side. Such models are available in the literature [97, 106] and are described in the earlier report [103]. Unfortunately, Eq. 4.2 is a differential equation in nine dimensions, a fact which makes direct numerical evolution prohibitive in the general case. The standard alternative is to represent $f$ using a fine-grained distribution and Monte Carlo methods. That is, $f$ is represented by a sufficiently large number of discrete distributions, each representing a number of particles, $N_p$, with the same particular characteristics $(\mathbf{x}_p, \mathbf{u}_p, r_p, T_p, t)$. All of the $N_p$ particles in a fine-grained distribution share the same evolution equation, and $f$ is found by summing over the discrete distributions. In this manner, the evolution of $f$ can be described using evolution equations for individual particles. Such evolution equations are provided in 4.2.

Because it is typically prohibitive to track all of the particles in a flow, representative parcels of particles are instead tracked. The particles in a given parcel share a common origin and common material properties. To further simplify the parcel evolution equations, each parcel consists of mono-disperse (single-diameter) particles so that all particles in the parcel are described by the same set of evolution equations. For flows where the particle size is distributed over a range of values, it is still necessary to track a statistically significant number of parcels to reproduce the mean behavior. Typically, a large number of parcels (tens or hundreds of thousands) are tracked to describe the evolution of the particle field.

### 4.1.2. A Combined Eulerian-Lagrangian Approach

The typical approach to CFD is to employ Eulerian descriptions of the flow field. Such an Eulerian formulation is employed to evolve the gas-phase continuum flow in the present case using standard methods [107]. To evolve the fine-grained distribution as indicated above, a Lagrangian approach is necessary [108, 109, 110, 111]. Such an approach has been used in other CFD applications including, for example, the popular internal combustion engineering simulation code, KIVA [97]. The coupling between the Eulerian and Lagrangian fields is key to capturing certain relevant physics, and this coupling is described in detail in 4.2 and in 4.3.

The present paper presumes that turbulent flow fields will be of interest, and that these turbulent flows cannot be fully resolved. Then, in addition to the continuity, momentum, species, and energy equations, there will be representations for the turbulent fluctuations. It is common to employ two-equation models in Reynolds-averaged Navier-Stokes (RANS) approaches while large-eddy

simulation (LES) techniques employ estimates of the subgrid fluctuations based on resolved quantities. For the present purposes, the $k-\epsilon$ turbulence model [112] will be presumed with extensions to other methods being straightforward.

When a particle collides with a solid wall, it is assumed to adhere to the wall if the impact velocity (kinetic energy) is sufficiently high, and bounces otherwise. In general, adherence is the predominant result of collisions for the particles considered here. It is well known that fine powders can be convectively lifted from surfaces and transported elsewhere, but this is beyond the scope of the current study. Models for the particle breakup due to hydrodynamic forces and for particle collisions are also available. For these purposes, models developed elsewhere and available in the literature [97, 106] are employed.

## 4.2. PARTICLE TRANSPORT MODEL

In this section, the equations describing the evolution of parcels of particles are presented. Models are presented for the particle motion and for heat and mass transfer (ie. evaporation and combustion).

### 4.2.1. Particle Acceleration and Trajectories

Particles with densities much greater than that of a the fluid phase (solid or liquid particles in gaseous flows) are primarily affected by drag forces and body forces. In this limit where $\rho_p \gg \rho_g$, the particle acceleration is written [98]:

$$\frac{du_{p,i}}{dt} = \frac{3\rho_g C_D |\mathbf{u}_g - \mathbf{u}_p|}{4\rho_p d_p}(u_{g,i} - u_{p,i}) + \left(\frac{\rho_p - \rho_g}{\rho_p}\right) g_i \tag{4.3}$$

where $u_{p,i}$ and $u_{g,i}$ are the $i^{th}$ component of the particle and gas velocities, respectively, $|\mathbf{u}_g - \mathbf{u}_p|$ is the vector magnitude of the velocity differences, $\rho_p$ and $\rho_g$ are the particle and gas densities, and $g_i$ is the $i^{th}$ component of the acceleration due to body forces. The particle diameter is $d_p$ and this should be considered to be the equivalent particle diameter corresponding to a spherical particle. The effects of non-sphericity on the acceleration can be accounted for through the drag coefficient, $C_D$. The gas-velocity to be employed in 4.3 is taken from the Eulerian solution of the continuum field. For turbulent flows, the sum of the mean (resolved) velocity and a perturbation to that mean, accounting for the turbulent fluctuations, both contribute to the gas velocity. The effects of turbulent fluctuations are described in the following section, 4.2.2.

For a spherical particle, the drag coefficient is modeled using standard drag coefficient relations:

$$C_D = \begin{cases} 24(1 + Re_p)^{2/3}/Re_p & for Re_p < 1000 \\ 0.424 & for Re_p > 1000 \end{cases} \tag{4.4}$$

The particle Reynolds number, $Re_p$, is based on the slip velocity and the particle diameter

$$Re_p = \frac{\rho_g d_p |\mathbf{u}_g - \mathbf{u}_p|}{\mu_g} \tag{4.5}$$

Yuen and Chen [113] recommend evaluating the viscosity in 4.5 based on the weighted average of the properties at the gas-phase side of the particle surface (weighted by two-thirds) and the gas-phase properties far from the particle (weighted by one-third), the so called '1/3 rule.' So for properties at the surface and far field identified with a subscripted $f$ and $\infty$, respectively, the viscosity would be

$$\mu_g = \mu_\infty/3 + 2\mu_f/3. \tag{4.6}$$

A similar relationship is suggested for other transport coefficients (conductivity, diffusivity, etc.). Note that additional forces are relevant for particles with densities nearer to or less than the continuum phase (bubbles) and for particles with high Reynolds numbers. A comprehensive overview of the force on particles is available from Maxey and Riley [104]. Equation 4.3 can be linearized and written

$$\frac{du_{p,i}}{dt} = \frac{(u_{g,i} - u_{p,i})}{\tau_p} + \left(\frac{\rho_p - \rho_g}{\rho_p}\right) g_i \tag{4.7}$$

$$\tau_p = \frac{4\rho_p d_p}{3\rho_g C_D |\mathbf{u}_g - \mathbf{u}_p|} \tag{4.8}$$

where $\tau_p$ is the particle velocity response time. In the small Reynolds number limit where the drag coefficient is inversely proportional to the slip velocity, this linearization is particularly relevant. Given the particle acceleration from Eq. 4.3, the particle trajectories can be determined by integrating the simple ODE

$$\frac{dx_{p,i}}{dt} = u_{p,i} \tag{4.9}$$

Since all the particles within a parcel are the same size, all parcel trajectories are determined by Eqn. 4.9, subject to turbulence effects described below in 4.2.2.

### 4.2.2. Particle Dispersion and Turbulence

In CFD modeling of turbulent flows, the full velocity spectrum is generally not resolved. Instead, certain velocity fluctuations are modeled, being represented through the turbulent kinetic energy, $k$, which is half the sum of the squares of the velocity fluctuations. These velocity fluctuations tend to introduce random fluctuations in the particle velocities that result in real particles being dispersed relative to the mean continuum flow [114, 115]. For Lagrangian particle methods, this phenomenon is modeled in two ways: by perturbing the velocity of parcels of particles and by affecting the spatial extent of the parcel itself.

To account for the effects of the velocity fluctuations on the parcels or particles, the random walk model of Gosman and Ioannides [108], as modified by Shuen et al. [109], is employed. In this approach, the gas velocity employed in equations 4.3, 4.5, 4.7, and 4.8 is the sum of the mean gas velocity, $\langle u_{g,i} \rangle$, and a fluctuating velocity that is sampled from a normal (Gaussian) velocity distribution with a standard deviation given by $\sigma_u = \sqrt{2k/3}$. For LES, $k$ here would be replaced with the subgrid kinetic energy. Sampling from the inverted cumulative distribution function with a random number uniformly distributed between zero and unity, $RN$, gives the appropriate fluctuating velocity. The total and fluctuating gas velocities are then

$$u_{g,i} = \langle u_{g,i} \rangle + u'_{g,i} \tag{4.10}$$

$$u'_{g,i} = \sqrt{2}\sigma_u erf^{-1}(2RN - 1) \tag{4.11}$$

where $erf^{-1}$ is the inverse error function. The time during which a given velocity fluctuation affects a given parcel is determined by the expected time that the particle takes to cross the eddy inducing the given velocity fluctuation. Small particles will tend to stay in an eddy for the duration of the eddy lifetime,

$$\tau_e = \sqrt{3/2}C_\mu^{3/4}k/\epsilon \tag{4.12}$$

where $\epsilon$ is the turbulent energy dissipation rate and $C_\mu = 0.09$. Larger particles will have sufficient slip velocity to cross the eddy. The eddy-crossing time is estimated as

$$\tau_C = -\tau_p \, ln\left[1 - \frac{L_e}{\tau_p|\mathbf{u}_g - \mathbf{u}_p|}\right] \tag{4.13}$$

where the eddy length scale is $L_e = C_\mu^{3/4}k^{3/2}/\epsilon$. The eddy interaction time, that is the time over which a given velocity perturbation affect $u_{g,i}$ in Eqns 4.10 and 4.11, is the minimum of $\tau_e$ and $\tau_C$. The particles comprising a parcel are presumed to be distributed about the center of the parcel, tracked by Eq. 4.9, in a normal (Gaussian) manner with the standard deviation in each direction given by $\sigma_i$, where $i$ is either $x$, $y$, or $z$. This distribution is written

$$f_\sigma(\vec{x};\vec{x_o},t) = \frac{N_p}{(2\pi)^{3/2}\sigma_x\sigma_y\sigma_z} exp\left(-\left[\frac{(x-x_o)^2}{2{\sigma_x}^2} + \frac{(y-y_o)^2}{2{\sigma_y}^2} + \frac{(z-z_o)^2}{2{\sigma_z}^2}\right]\right) \tag{4.14}$$

where $N_p$ is the total number of particles in the parcel and $\vec{x_o}$ is the center of the parcel. The spatial extent of the parcel is thus determined by $\sigma_i$. This term is affected by unresolved turbulent fluctuations that act differently on the particles across the parcel. Following Zhou and Yao [110], the spatial extent of the parcel is the mean square displacement over time

$${\sigma_x}^2 = \sum\left[{u'_{p,i}}^2(\Delta t_i)^2\right] \tag{4.15}$$

where $u'_{p,i}$ satisfies the ODE

$$\frac{du'_{p,i}}{dt} = \frac{(u'_{g,i} - u'_{p,i})}{\tau_p} \tag{4.16}$$

and $\Delta t_i$ is the time over which $u'_{p,i}$ acts. Equation 4.16 is obtained by subtracting the instantaneous particle equations from the mean particle equations. Note that large particles with large $\tau_p$ are not dispersed appreciably.

### 4.2.3. Mass and Energy Exchange between Particles and the Gas Phase

Particles may exchange mass and energy with the gas phase according to conservation principles across the interface. The physics of mass and energy transfer are described in detail here because the anticipated applications include a wide range of physical phenomena that have not been described together. Included in the phenomena of interest are metal and hydrocarbon particle combustion as well as particle

condensation and evaporation. Each of these is anticipated to be strongly energetic in the sense that the product of the evaporation rate with the ethalpy of evaporation and combustion is expected to be substantial. Further, for evaporation and condensation applications, it is expected that vapor pressures will range over a sufficiently wide range that the transition from evaporation to condensation should be correctly described as this often limits evaporation and condensation rates. For metal combustion, the associated temperatures are sufficiently high so that radiative heat transfer must be considered.

The theory for droplet vaporization and combustion has generally been developed based on a large number of simplifications [92, 93] including spherical-symmetry, unity Lewis numbers, droplet surfaces at the boiling temperature, and infinitely fast conduction through the droplet. Recent work provides guidance as to how these assumptions can be relaxed [116].

#### 4.2.3.1. Theory for spherically symmetric flow

In general, the theory of heat and mass transfer is developed for spherically symmetric systems and then corrected to account for increased transfer associated with advection and asymmetry. In this section, relations are developed based on the assumption of spherical symmetry and corrections is provided in 4.2.3.2.

$$\frac{\partial \rho}{\partial t} + \frac{1}{r^2}\frac{\partial}{\partial r}(\rho r^2 \nu) = 0 \tag{4.17}$$

can be integrated to give

$$\dot{m}_o = 4\pi \rho r^2 \nu \tag{4.18}$$

where $\dot{m}_o$ is the rate of mass evaporation from the particle, $\nu$ is the Stefan velocity, directed normally away from the particle, and $\rho$ is the local vapor density considering both the particle vapor and the continuum gas concentrations. A coordinate transformation to the variable

$$\xi_T = \int_{r_o}^{\infty} \frac{1}{4\pi r^2 (\lambda / c_p)} dr \tag{4.19}$$

or

$$\xi_F = \dot{m}_o \int_{r_o}^{\infty} \frac{1}{4\pi r^2 (\rho D_F)} dr \tag{4.20}$$

which represents the ratio of the Stefan velocity to the thermal diffusion velocity, greatly simplifying the species and energy conservation equations. In Eqns. 4.19 and 4.20, $\lambda$ is the thermal conductivity of the vapor, $c_p$ is its specific heat at constant pressure, and $D_F$ is the evaporating species diffusion coefficient. The subscript o on the evaporation rate indicates that this evaporation corresponds to that for the spherically-symmetric case; corrections relating the evaporation rate for the spherically-symmetric case to that with finite-slip velocities are provided in 4.2.3.2. In the spherically-symmetric case, conservation equations for conserved scalars, $\beta_k$, can be written

$$\frac{\partial \beta_k}{\partial \xi_k} + \frac{\partial^2 \beta_k}{\partial {\xi_k}^2} = 0 \tag{4.21}$$

for which an analytic solution

$$\beta_k = C_1 + C_2 e^{-\xi_k} \tag{4.22}$$

is readily obtained. In non-reacting, non-radiating flows, any mass fraction or temperature can be a conserved scalar. Other conserved scalars are provided below. The appropriate from of 4.19 and 4.20 to be used depends on the variable represented by 4.21. Since the diffusion coefficients appearing in 4.20 are most important near the particle surface, the diffusion coefficient to be employed is that most relevant at the surface. The choice will be clearly identified below. The application of the Dirichlet boundary conditions at both the surface and far from the droplet and the application of a Neumann boundary condition at the surface relate the boundary conditions for the conserved scalar to $\dot{m}_o$ through $\xi_k$

$$\xi_{k,f} = \dot{m}_o \int_{r_o}^{\infty} \frac{1}{4\pi r^2(\rho D)} dr = \ln\left(1 + \frac{\beta_\infty - \beta_f}{-\frac{d\beta}{d\xi_k}|_f}\right) \tag{4.23}$$

where the subscript $f$ indicates quantities evaluated at the droplet surface, the so-called film state. For reacting flows [1] we will consider two such conserved scalars in the present work. Allowing variable properties, the temperature oxidizer coupling function is

$$\beta_{T-O} = \int_{T_o}^{T} c_P dT + \frac{Y_O W_F q_{comb}}{\nu_O W_O} \tag{4.24}$$

where $W_i$ is the molecular weight and $\nu_i$ is the stoichiometric coefficient of species $i$. The standard enthalpy of combustion for fuel and oxidizer, per unit mass of fuel evaluated at the film temperature, $T_f$, is

$$q_{comb} = h_{F,f} + \frac{h_{O,f} W_O \nu_O}{W_F \nu_F} - \sum \left[\frac{h_{p,f} W_p \nu_p}{W_F \nu_F}\right] \tag{4.25}$$

where the last summation is taken over the produces of reaction, $p$. Note that when there is no combustion, the second term in Eqn. 4.25 is ignored. For $\beta_{T-O}$, the thermal diffusivity is the relevant diffusivity so that Eqn. 4.19 is employed in conjunction with $\beta_{T-O}$.

The constants $C_1$ and $C_2$ in 4.22 are evaluated using the boundary conditions for the temperature and oxidizer at the droplet surface and in the far field. The boundary conditions employed for temperature are that the heat flux into the particle is balanced by the sum of the enthalpy of vaporization, the heating of the particle and any radiative losses

$$4\pi r^2 \lambda \left(\frac{dT}{dr}\bigg|_f\right) = -\dot{m}_o c_{p,f} \left(\frac{dT}{\xi_T}\bigg|_f\right) = \dot{m} h_{vap} + Q_{rad} + m_p c_{v,p} \frac{dT_p}{dt} \tag{4.26}$$

Here, the enthalpy of vaporization is $h_{vap}$, the particle specific heat is $c_{v,p}$, and the radiative heat loss over the droplet surface is

$$Q_{rad} = 4\pi {r_p}^2 \alpha(\sigma {T_p}^4 - G_{in}/4) \tag{4.27}$$

where $\alpha$ is the particle absorptivity, $\sigma$ is the Stefan-Boltzmann constant, and $G_{in}$ is the incident radiation. The incident radiation is the radiation intensity integrated over all directions, that is, the

[1] The general configuration considered is a fuel droplet reacting in an oxidizing atmosphere; in the event that, for example, an oxidizer droplet is reacting in a fuel atmosphere, then the "oxidizer" and "fuel" described would be switched. Also, if there is no reaction, the results generalize to droplet evaporation where the "fuel" is the evaporating component.

entire $4\pi$ steradian solid angle. If the radiation intensity is $I$, then $G_{in} = \int_{4\pi} I d\Omega$ where $d\Omega$ is the differential solid angle. If the particle is not opaque, the particle absorptivity will be a function of the particle size [117]. Note that, in Eqn 4.26, the evaporation rate appears without the subscript o, indicating that this is the evaporation rate corrected for finite-slip velocities as prescribed in 4.42 below. This is appropriate because the heat and mass flux to the surface are both increased by the relative droplet motion, while the other terms in 4.26 are not affected by that. The oxidizer is presumed to not be absorbed by the surface so that a no-flux boundary condition is employed

$$\frac{dY_O}{d\xi_T} = -Le_{O,f} Y_{O,f}. \tag{4.28}$$

Because the thermal diffusivity is used in definition $\xi_T$ for $\beta_{T-O}$, the ratio of the thermal to mass diffusivity in the form of th oxidizer Lewis number appears in 4.28. Taking the derivative of Eqn. 4.24 evaluated at the droplet surface and using 4.26 and 4.28, we obtain

$$\left( \left. \frac{d\beta_{T-O}}{d\xi_T} \right|_f \right) = -h_{vap} - \frac{Q_{rad}}{\dot{m}} - \frac{m_p c_{v,p}}{\dot{m}} \frac{dT_p}{dt} + \int_{T_o}^{T} \frac{dc_{p,f}}{d\xi_T} dT - \frac{Y_{O,f} W_F q_{comb}}{Le_{O,f} \nu_O W_O} \tag{4.29}$$

that will appear in the denominator of Eqn. 4.23. Also the temperature and oxidizer mass fraction must approach their far field values at large radii. Additional assumptions are required to identify the temperature and the oxidizer mass fraction at the surface; these will be discussed later. Applying these boundary conditions to 4.22 for $\beta_{T-O}$ provides an expression for the rate of evaporation in terms of $\xi_T$ evaluated at the surface of the particle

$$\xi_{T,f} = \dot{m}_O \int_{r_o}^{\infty} \frac{1}{4\pi r^2 (\lambda / c_p)} dr = \ln\left[1 + B_{T-O}\right] \tag{4.30}$$

where the Spalding transfer number associated with $\beta_{T-O}$ is

$$B_{T-O} = \frac{\int_{T_O}^{T} c_{p,\infty} dT - \int_{T_O}^{T} c_{p,f} dT + q_{comb} \left[ \dfrac{(Y_{O,\infty} - Y_{O,f}) W_F}{\nu_O W_O} \right]}{h_{vap} + \dfrac{Q_{rad}}{\dot{m}} + \left( \dfrac{m_p c_{v,p}}{\dot{m}} \dfrac{dT_p}{dt} \right) - \int_{T_O}^{T} \dfrac{dc_{p,f}}{d\xi_T} dT + \dfrac{Y_{O,f} W_F q_{comb}}{Le_{O,f} \nu_O W_O}} \tag{4.31}$$

Here the subscripts $f$ and $\infty$ indicate the states at the particle surface (on the gas-phase side of the interface) and the ambient far-field environment, respectively. The denominator in 4.31 represents a variety of potential sinks for the enthalpy at the droplet surface. These sinks include, in the order in which they are written, the enthalpy associated with vaporizing the particle, the radiative losses from the surface, the enthalpy conducted into the particle, the enthalpy flux to the gas-phase due to variable specific heats, and the enthalpy of combustion lost as a consequence of oxidizer leakage. Note that the radiative absorption and emission from the particle surface, $Q_{rad}$, are included here in the surface boundary condition, but the radiative losses from a flame around the droplet must be accounted for by modifying $q_{comb}$ to provide an effective heat release, the heat release decremented by the near flame radiative losses. As far as the droplet is concerned, these radiative losses are far enough away to not affect the film state except to the extent that radiative flux to the surface is affected.

The above expressions comprise a relatively complete definition of the physics of particle evaporation, combustion, and interaction with a radiative field accounting for variable thermophysical properties. These expressions are simplified by making the assumption that no oxidizer penetrates a flame to reach the surface, $Y_{O,f} = 0$, and by setting $T_f = T_O$ leading to

$$B_{T-O} = \frac{\int_{T_f}^{T_\infty} c_{p,\infty} dT + \dfrac{Y_{O,\infty} W_F q_{comb}}{\nu_O W_O}}{h_{vap} \dfrac{Q_{rad}}{\dot{m}} + \left( \dfrac{m_p c_{v,p}}{\dot{m}} \dfrac{dT_p}{dt} \right)} \tag{4.32}$$

where the denominator is referred to as the effective enthalpy

$$h_{eff} = h_{vap} + \frac{Q_{rad}}{\dot{m}} + \left( \frac{m_p c_{v,p}}{\dot{m}} \frac{dT_p}{dt} \right). \tag{4.33}$$

Equation 4.32 is employed in the numerical models. A coupling function for fuel and oxidizer is similar written

$$\beta_{T-O} = \frac{Y_F}{W_F} - \frac{Y_O}{\nu_O W_O} \tag{4.34}$$

For a species that is evaporating, the flux boundary condition is

$$\left( \left. \frac{dY_F}{d\xi_F} \right|_f \right) = Y_{F,p} - Y_{F,f} \tag{4.35}$$

and the fuel-oxidizer Spalding mass transfer number is

$$B_{F-O} = \frac{Y_{F,f} - Y_{F,\infty} + \dfrac{Y_{O,\infty} W_F}{\nu_O W_O}}{Y_{F,p} - Y_{F,f}} \tag{4.36}$$

where $Y_{O,f} = 0$ has been assumed as in Eq. 4.32. Note that only one of $Y_{O,\infty}$ or $Y_{F,\infty}$ will be non-zero based on the current assumption of zero leakage through flames; if combustion is occurring then it will be $Y_{F,\infty}$ that is zero. The diffusion coefficient appropriate for the fuel-oxidizer coupling function is that for the fuel so that the second of Eqn 4.19 and 4.20 is used with the diffusion coefficient specifically that of the fuel, and the equivalent of Eqn. 4.30 for the fuel-oxidizer system

$$\xi_{F,f} = \dot{m}_o \int_{r_o}^{\infty} \frac{1}{4\pi r^2 (\rho D_F)} = \ln[1 + B_{F-O}] \tag{4.37}$$

#### 4.2.3.2. Extension to multiple oxidizers

Multiple oxidizers is discussed in detail later, in 4.2.4.2.

#### 4.2.3.3. Correlations for finite slip velocities

The above relationships for the heat and mass transfer are derived for a spherically symmetric field around the droplet and are valid for droplets with zero slip velocity in the absence of buoyancy. Empirical correlations are available in terms of the Nusselt and Sherwood numbers parametrized by Reynolds, Schmidt, and Prandtl numbers to describe the effect of finite slip velocities in modifying the heat and mass transfer by reducing the boundary layer thickness. The Schmidt and Prandtl numbers are defined:

$$Pr = \frac{c_{p,g}\mu_g}{\lambda_g} \tag{4.38}$$

$$Sc = \frac{\mu_g}{\rho_g D_g} \tag{4.39}$$

These quantities with the subscript $g$ represent appropriate averages for transport properties in the gas phase boundary layer around the particle.These gas-phase quantities are evaluated using an appropriate averaging process that will generally be analogous to Eqn. 4.6. The alternative is to tabulate these quantities using Eqns 4.46 and 4.47. The Nusselt number describes a dimensionless heat transfer rate to the droplet for a given difference between the ambient and surface temperature,

$$Nu_f = \left( 2r_p c_{p,g} \left.\frac{dT}{dr}\right|_f \right) \Bigg/ \left( \int_{T_f}^{T_\infty} c_{p,\infty} dT + \frac{Y_{O,\infty} W_F q_{comb}}{\nu_O W_O} \right) \tag{4.40}$$

The correction to the evaporation rate employed in the present work is based on measurements by Ranz and Marshall (1952)

$$\frac{Nu_f}{Nu_{f,Re=0}} = (1 + 0.3 Re^{1/2} Pr^{1/3}) \tag{4.41}$$

This can be introduced into 4.31, and the evaporation rate can be written

$$\dot{m} = \frac{Nu_f}{Nu_{f,Re=0}} \dot{m}_o \tag{4.42}$$

The Nusselt number for zero slip velocity, $Nu_{f,Re=0}$, is included in the relations of the previous section, specifically in Eq. 4.31. A similar correlation,

$$\frac{Sh_f}{Sh_{f,Re=0}} = (1 + 0.3 Re^{1/2} Sc^{1/3}) \tag{4.43}$$

can be used for the Sherwood number,

$$Sh_f = \left( -2r_p \left.\frac{dY_F}{dr}\right|_f \right) \Bigg/ \left( Y_{F,f} - YF, \infty + \frac{Y_{O,\infty} W_F}{\nu_O W_O} \right) \tag{4.44}$$

which is the dimensionless mass transfer coefficient, so that the evaporation rate for finite slip velocities can also be written

$$\dot{m} = \frac{Sh_f}{Sh_{f,Re=0}} \dot{m}_o \tag{4.45}$$

where $\dot{m}_o$ should be taken from Eqn. 4.37.

#### 4.2.3.4. Evaporation rates and effective diffusivities

The evaporation rate is seen in Eqn 4.30 and 4.37 to depend linearly on an area-weighted gas-phase diffusivity. Since accurate evaluation of this weighting is not feasible for the particle transport model, effect diffusivities,

$$\left(\frac{\lambda}{c_p}\right)_{eff} = \left[4\pi r_p \int_{r_o}^{\infty} \frac{1}{4\pi r^2 (\lambda/c_p)} dr\right]^{-1} \tag{4.46}$$

$$(\rho D_F)_{eff} = \left[4\pi r_p \int_{r_o}^{\infty} \frac{1}{4\pi r^2 (\rho D_F)} dr\right]^{-1} \tag{4.47}$$

are defined. The one-third rule defined in Eq. 4.6 provides a rough guideline for evaluating these diffusion coefficients in some cases. The effective thermal diffusion coefficient is combined with Eqns. 4.30 and 4.42 to give an evaporating rate based on thermal diffusion

$$\dot{m} = 4\pi r_p \left(\rho D_F\right)_{eff} \frac{Sh_f}{Sh_{f,Re=0}} \ln\left[1 + B_{T-O}\right]. \tag{4.48}$$

The strong dependence of the evaporating rate on the diffusion coefficient, coupled with the fact that the diffusion coefficients depend strongly on variations in the compositions and temperature of the gases around the droplet, mean that the reasonable but judicious choice of diffusion coefficients can often match observed experimental measurements. Similarly, the mass transfer driven evaporation rate can be written with the use of Eqns. 4.37, 4.43, and 4.45

$$\dot{m} = 4\pi r_p \left(\rho D_F\right)_{eff} \frac{Sh_f}{Sh_{f,Re=0}} \ln\left[1 + B_{T-O}\right] \tag{4.49}$$

It is necessary that the evaporation rate predicted by Eqn. 4.49 be equal to that predicted by 4.48. These equations show that thermal and mass diffusion vary both through their diffusion coefficients and through difference in boundary layer thickness attributable to finite slip velocities. Both these effects are combined in an effective Lewis number, $Le_{eff}$, to give

$$Le_{eff} = \frac{\int_{r_o}^{\infty} \frac{dr}{4\pi r^2 (\rho D_f)} \frac{Nu_f}{Nu_{f,Re=0}}}{\int_{r_o}^{\infty} \frac{dr}{4\pi r^2 (\lambda/c_p)} \frac{Sh_f}{Sh_{f,Re=0}}} \tag{4.50}$$

Equations 4.48, 4.49, and 4.50 are used in the computational model for the evaporation rate. Note that energy and mass conservation must vie the same evaporation rate; this requirement determines the film state as described in the following section, 4.2.3.5.

#### 4.2.3.5. Closure of film state with effective heat transfer coefficient

The system described by Eqns. 4.48 and 4.49 is closed with two additional assumptions. First, the film conditions, $T_f$ and $Y_f$, are related through the Clausius-Clapeyron relationship

$$P_{F,f} = P_{ref} exp\left[-\frac{h_{vap}}{R}\left(\frac{1}{T_f} - \frac{1}{T_{ref}}\right)\right] \tag{4.51}$$

where the partial pressure gives the mole fraction through $X_F = P_{F,f}/P$ that can subsequently be converted to the mass fraction with the relationship $Y_F = X_F W_F / \Sigma_k X_k W_k$ . As provided by Lefebvre [118], such a relationship is

$$h_{vap} = h_{vap,ref} \left( \frac{T_{crit} - T_f}{T_{crit} - T_{ref}} \right)^{0.38} \tag{4.52}$$

for $T_f < T_{crit}$ and zero otherwise. If the critical point temperature is not provided, the code sets $T_{crit}$ to a very large value, essentially making $h_{vap}$ independent of temperature. Second, the droplet heating is related to the difference between the film temperature and the droplet temperature by assuming an effective internal heat transfer coefficient in the form of an internal Nusselt number, $Nu_p$, for the particle so that

$$m_p c_{v,p} \frac{dT_p}{dt} = 2\pi r_p N u_p \lambda_p (T_f - T_p). \tag{4.53}$$

This internal Nusselt number, which is different from that for the external heat transfer indicated in Eqn. 4.42, can be estimated based on the results of numerical studies where the internal droplet was resolved [116, 119]. There the Nusselt numbers for no circulation and for rapid circulation were identified as 6.58 and 17.9, respectively, and a transition region was also identified based on the liquid Peclet number

$$Pe_p = \frac{2\rho_p c_{v,p} U_{surface} r_p}{\lambda_p} \tag{4.54}$$

where

$$U_{surface} = \frac{12.69 |\mathbf{u}_p - \mathbf{u}_g| Re_p^{1/3}}{16} \left( \frac{\mu_g}{\mu_p} \right) \tag{4.55}$$

which is based on the maximum surface velocity, $U_{surface}$. This transition was empirically fitted [116] to

$$Nu_p = 6.58 \left[ 1.86 + 0.86 tanh \left[ 2.245 log_{10} \left( Pe_p / 30 \right) \right] \right]. \tag{4.56}$$

In [116], both the evaporation rate and the surface temperature were reproduces using Eq. 4.56 with little error compared to simulations incorporating a detailed internal droplet convection model. Naturally, the liquid Peclet number should be zero if the particle is below the freezing temperature of the particle constituent.

The evaporation rates indicated in Eqns. 4.48 and 4.49 must be equal, subject to the constraints of Eqns. 4.51 and 4.53 based on the closure approximation employed in this section. Equating Eqns. 4.48 and 4.49 leads to a nonlinear equation for the surface temperature that is to be solved. This is readily solved using Newton's method as described here. Newton's method is an iterative root-finding method written in the form

$$T_f^{n+1} = T_f^n - g(T_f^n)/g'(T_f^n) \tag{4.57}$$

where $g(T_f) = 0$ is the equation for which the root will be found and the superscript $n$ refers to the iteration number. Equating Eqns. 4.48 and 4.49 and using 4.50 gives

$$g(T_f) = (1 + B_{T-O})^{Le_{eff}} - 1 - B_{F-O} \tag{4.58}$$

Differentiation gives

$$g'(T_f) = Le_{eff} (1 + B_{T-O})^{(Le_{eff}-1)} \frac{dB_{T-O}}{dT_f} - \frac{dB_{F-O}}{dT_f} \tag{4.59}$$

$$\frac{dB_{T-O}}{dT_f} = \frac{-\left[c_{p,\infty}(T_\infty - T_f) + \frac{Y_{O,\infty}W_F q_{comb}}{\nu_O W_O}\right]\frac{dh_{eff}}{dT_f} - c_{p,\infty}h_{eff}}{h_{eff}} \tag{4.60}$$

$$\frac{dB_{F-O}}{dT_f} = \frac{\left(Y_{F,f} - Y_{F,\infty} + \frac{Y_{O,\infty}W_F}{\nu_O W_O}\right)}{(Y_{F,p} - Y_{F,f})^2}\frac{dY_{F,f}}{dT_f} \tag{4.61}$$

$$\frac{dY_{F,f}}{dT_f} = \frac{-W_F W_g \frac{P_{atm}}{P_{ref}}\frac{h_{vap}}{R}\left[\frac{1}{{T_f}^2} + \frac{0.38}{(T_{crit} - T_f)}\left(\frac{1}{T_f} - \frac{1}{T_{ref}}\right)\right] exp\left[-\frac{h_{vap}}{R}\left(\frac{1}{T_f} - \frac{1}{T_{ref}}\right)\right]}{W_F - W_g + W_g\frac{P_{atm}}{P_{ref}} exp\left[-\frac{h_{vap}}{R}\left(\frac{1}{T_f} - \frac{1}{T_{ref}}\right)\right]} \tag{4.62}$$

$$\frac{dh_{eff}}{dT_f} = \frac{2\pi r_p Nu_p \lambda_p}{\dot{m}} - \frac{Le_f\left[Q_{rad} + 2\pi r_p Nu_p \lambda_p (T_f - T_p)\right]}{\dot{m}(1 + B_{F-O})\left[\ln(1 + B_{F-O})\right]}\frac{dB_{F-O}}{dT_f} \tag{4.63}$$

In writing these expressions for $g'$, it is assumed that $Y_{O,f} = 0$ and $T_O = T_f$.

Because of the strong nonlinearities in $g(T_f)$, care must be taken in providing initial conditions to solve these relations. This is conducted through a two-stage procedure. First, the boiling point temperature is identified, then a temperature just under the boiling temperature is used as an initial guess for the iterative solution that determines the film temperature. This procedure arises from a consideration of the shape of the $g(T_f)$. For realistic temperatures, those for which $0 < Y_f < 1$, both $g(T_f)$ and $g'(T_f)$, are monotonically strongly increasing in magnitude. An initial guess with a temperature that is too low (less than $T_f$) results in a prediction, with the Newton's method, of a very high temperature on the successive iteration due to the small magnitude of the derivative, $g'$, for a small T. Typically, this second iteration will result in a temperature for which $Y_f > 1$ that leads to negative values of $B_m$ and the iteration diverges into non-physical regimes. However, an initial guess that is within the physically reasonable regime ($0 < Y_f < 1$) and yet above the final $T_f$ will reliably converge. Therefore, the initial guess of $T = 0.99999T_{boil}$ is used as an initial guess in determining the film temperature. This method has been tested for a wide range of conditions and appears robust except for those scenarios where the denominator of $B_T$ takes on negative values. (Since $T_f$ can be less than $T_p$ and this can result in the denominator of $B_T$ taking on negative values, in which case the iteration may fail.) In the event that the iteration fails, the fast conduction limit described in 4.2.3.6 is employed.

#### 4.2.3.6. Closure for surface state assuming fast conduction

The surface state described in the previous section is the most physically realistic state that can be obtained without solving a differential equation for the heat transfer with the droplet. However, determining this state involves the iterative solution of a system of nonlinear equations. In the previous section a robust method of solving these equations is provided, but a simpler approximation may provide suitable results under certain conditions. This simpler approach is to assume that the heat transfer within the droplet is fast relative to the heating of the droplet. This is equivalent to taking the zero Biot number limit, $Bi \approx Nu_f\lambda_f/Nu_p\lambda_p \to 0$, in Eqn. 4.53, in which case $T_f = T_p$. With the film state determined by the droplet temperature, the film mass fraction is directly obtained from Eqns. 4.51 and 4.52, and the mass transfer number is obtained from Eq. 4.36 with the evaporation rate following

from Eqn. 4.49. The thermal transfer number is obtained from the mass transfer number by equating Eqns. 4.49 and 4.48, and the droplet heating is obtained y solving for the enthalpy change of the particle in Eq. 4.31 to obtain

$$\frac{m_p c_{v,p}}{\dot{m}} \frac{dT_p}{dt} = -h_{vap} - \frac{Q_{rad}}{\dot{m}} + \frac{\int_{T_f}^{T_\infty} c_{p,\infty} dT + \frac{Y_{O,\infty} W_F q_{comb}}{\nu_O W_O}}{B_{T-O}} \tag{4.64}$$

Sirignano [120] and coworkers have demonstrated that for many conditions, this particular limit is a poor approximation for at least some of the particle lifetime. In the present models, this limit is employed in two situations: (a) if the particle temperature is within 1% of the wet bulb temperature, and (b) if the particle temperature exceeds the wet bulb temperature. As the droplet temperature approaches the wet bulb temperature, employing this limit is inconsequential. In the latter case, the rate of droplet cooling is likely to be over predicted, but this is a scenario for which the convergence of the film state otherwise is not guaranteed. In the interest of creating a more robust model, and because this particular situation is not anticipated to be predominant, we employ this simpler limit.

## 4.2.4. Conserved scalars and transfer numbers for various applications

In 4.2.3.3, expressions are provided for the particle evaporation rate, 4.49 and 4.48, using a model system of fuel evaporating from the particle and reactive with an oxidizer that diffuses from the ambient gas. Of significance in those expressions are the transfer numbers defined in 4.2.3.1 in Eqns. 4.32 and 4.31, and these in turn are based upon conserved scalars defined in Eqns. 4.24 and 4.34. In this appendix, expressions for alternate transfer numbers and conserved scalars are provided for two additional systems: a simpler system in which evaporating or condensation occurs without any reaction in the boundary layer (ie. water droplet evaporation or condensation) and a more complicated system in which multiple oxidizers are involved in the oxidation of the evaporated fuel (relevant to metal oxidation).

### *4.2.4.1. Simple evaporation and condensation*

When species evaporate or condense but do not otherwise react in the boundary layer surrounding the particle, the species mass fraction of the evaporating/condensing species itself is a conserved scalar. In this case, the mass transfer number corresponding to Eq. 4.31 is simply

$$B_{F-O} = \frac{Y_{F,f} - Y_{F,\infty}}{Y_{F,p} - Y_{F,f}} \tag{4.65}$$

and the heat transfer number corresponding to 4.32 is

$$B_{T-O} = \frac{\int_{T_f}^{T_\infty} c_{p,\infty} dT}{h_{vap} + \frac{Q_{rad}}{\dot{m}} + \left( \frac{m_p c_{v,p}}{\dot{m}} \frac{dT_p}{dt} \right)} \tag{4.66}$$

#### 4.2.4.2. Multiple Oxidizers

For metal oxidation, it is possible to have multiple oxidizers that simultaneously react with the evaporating metal. These can be expressed with a series of parallel single-oxidizer reactions of the form $F + \nu_{O_i} O_i \rightarrow \Sigma \nu_{P_{i,j}} P_{j,i}$. Useful conserved scalars that can be formed with this set of reactions include

$$\beta_{F-O} = \frac{Y_F}{W_F} - \sum_i \frac{Y_{O_i}}{\nu_O W_{O_i}} \tag{4.67}$$

$$\beta_{T-O} = \int_{T_f}^{T} c_p dT + \sum_i \frac{W_F q_i}{\nu_O W_{O_i}} Y_{O_i} \tag{4.68}$$

where the enthalpy of reaction of the $i$-th oxidizer, $O_i$, with $F$ is given by $q_i$. These are analogous to the conserved scalars defined in Eqns. 4.24 and 4.34. Using these conserved scalars, a mass transfer number analogous to Eqn. 4.31 is found to be

$$B_{F-O} = \frac{Y_{F,p} - Y_{F,\infty} + \sum_i \frac{W_F q_i}{\nu_{O_i} W_{O_i}}}{Y_{F,p} - Y_{F,f}} \tag{4.69}$$

and the heat transfer number is

$$B_{T-O} = \frac{\int_{T_f}^{T_\infty} c_{p,\infty} dT + \sum_i \frac{Y_{O_i,\infty} W_F}{\nu_{O_i} W_{O_i}}}{h_{vap} + \frac{Q_{rad}}{\dot{m}} + \left( \frac{m_p c_{v,p}}{\dot{m}} \frac{dT_p}{dt} \right)} \tag{4.70}$$

These can be employed in Eqn. 4.48 and 4.49 to provide expressions for the particle mass burning rate as a function of the various oxidizer mass fractions far from the particle.

### 4.2.5. Energy Exchange between Particles and the Gas Phase Without Mass Transfer

In this section, a whersimpler scenario is considered where evaporation from and condensation onto particles is presumed to be negligible. For example, metal particles in dry air at ambient temperatures are unlikely to participate in evaporation or condensation. Models are presented in this section to treat these scenarios. The models described in the previous sections are ill posed to solve these problems because the formulation is based on a balance between diffusion and the Stefan convective velocity, which is proportional to the negligible $\dot{m}$. The models in this section are triggered if the particle constituent or its product of reaction does not exist in the gas phase. They are also triggered if the vapor pressure calculated using both the droplet temperature and the gas temperature results in values for each of the three terms in the numerator of $B_{F-O}$ that are less than a tiny number (say $10^{-8}$). In these cases, the fuel-oxidizer coupling function will result in a negligibly small evaporation rate that can be bounded without requiring a solution of the actual film state.

In the event that there is no mass transfer, the particle heating rate is determined based on the balance between conductive and radiative transfer. The conductive heat transfer can be expressed using an

effective heat transfer coefficient, thereby taking advantage of available Nusselt number correlations indicated in 4.2.4.1. Then the relationship for the droplet heating is

$$m_p c_{v,p} \frac{dT_p}{dt} = 2\pi N u_f r_p \lambda_f (T_g - T_f) + 4\pi \alpha r_p^2 (G_{in}/4 - \sigma T_p^4). \tag{4.71}$$

For the particle with no mass transfer, the closure of the film temperature is obtained by equating the external heat flux, described by the right hand side of Eqn. 4.71, with the internal heat flux, as indicated on the right hand side of Eqn. 4.53 to obtain

$$T_f = \frac{N u_f \lambda_f T_g + N u_p \lambda_p T_p + 2\alpha r_p (G_{in}/4 - \sigma T_p^4)}{N u_f \lambda_f + N u_p \lambda_p} \tag{4.72}$$

## 4.2.6. Further Notes on Radiative Heat Transfer

The expression employed for radiative droplet heating in Eqn. 4.27 is appropriate for relatively large and opaque particles, referred to as the geometric optics limit. For different particles, the expression for the absorptivity will change, but otherwise the expressions remain appropriate. To identify the appropriate absorptivity, it is useful to compare absorption coefficients for particle clouds. The absorption coefficient, $\kappa$, in units of inverse length, is an effective cross-sectional area per volume. For large particles with a condense phase absorptivity give by $\alpha$, the absorption coefficient is the summation of the cross-sectional areas time the absorptivity

$$\kappa = \frac{1}{V_c} \sum\nolimits_p \left[ \pi r_p^2 \alpha \int_{V_c} f_\sigma(\mathbf{x}; \mathbf{x_o}, t) d\mathbf{x} \right]. \tag{4.73}$$

For comparison, in the small-particle or Rayleigh limit, the absorption coefficient is proportional to the volume fraction with the proportionality coming from the complex index of refraction. For intermediate particles where the particle optical depth is comparable to the particle radius, intermediate limits are appropriate, and the absorption can range from being proportional to the particle area to being proportional to the particle volume. The appropriate absorption coefficient, and from it the particle emissivity, cases can be determined as described in the available texts [117, 121].

## 4.2.7. Input Parameters for Particle Evolution

A large number of parameters are required to specify the evolution of the particles. The parameters provided in the input file are described in table 4.2-1. In addition, table 4.2-2 specifies those variables that must be obtained from the gas-phase continuum flow. Because parcels of particles have finite extent and may span several control volumes, it is sometimes necessary to interpolate values of these gas-phase variables from several control volumes.

**Table 4.2-1.. Input parameters related to the particle evolution provided through the input file**

| Variable | Input name | Units | Description |
|---|---|---|---|
| $\rho_p$ | INJP_DENp | $kg/m^3$ | Particle density |
| $T_p$ | INJP_Tp | $K$ | Particle temperature |
| $W_F$ | INJP_MWp | $kmol/kg$ | Molecular weight of fuel or particle component |
| $c_{v,p}$ | INJP_Clp | $J/kg/K$ | Particle specific heat |
| $\alpha$ | INJP_Emp | none | Particle absorptivity,emissivity |
| $Pr$ | INJP_Prf | none | Film Prandtl number |
| $Sc$ | INJP_Scf | none | Film Schmidt number |
| $P_{ref}$ | INJP_Prefvapp | $Pa$ | Reference pressure for vaporization for particle |
| $T_{ref}$ | INJP_Trefvapp | $K$ | Reference temperature for vaporization for particle |
| $h_{vap,ref}$ | INJP_Hvaprefp | $J/kg$ | Reference enthalpy of vaporization for particle |
| $T_{crit}$ | INJP_Tıcp | $K$ | Critical temperature for particle |
| $T_{fr}$ | INJP_Tıfp | $K$ | Freezing temperature for particle |
| $\sigma_p$ | INJP_STp | $N/m$ | Particle surface tension |
| $\mu_p$ | INJP_mup | $kg/m/s$ | Particle viscosity |
| $Pr_p$ | INJP_Prl | none | Particle Prandtl number |
| $q_{comb}$ | | $J/kg$ | Enthalpy of combustion for vapor species evaluated at $T_f$ |
| $\nu_i$ | | mol/ mol Fuel | Stoichiometric coefficients (relative to evaporating species) |

**Table 4.2-2.. Variables passed from the gas-phase Eulerian solver required to evolve the particles**

| Variable | Units | Description |
|---|---|---|
| $\rho_\infty$ | $kg/m^3$ | Gas-phase density |
| $P$ | $Pa$ | Pressure |
| $u_g, v_g, w_g$ | $m/s$ | Mean gas velocity |
| $k$ | $m^2/s^2$ | Turbulent kinetic energy |
| $\epsilon$ | $m^2/s^3$ | Turbulent kinetic energy dissipation rate |
| $T_\infty$ | $K$ | Gas temperature |
| $W$ | $kg/mol$ | Gas-phase mean molecular weight |
| $Y_{F,\infty}$ | N/A | Mass fraction of fuel in gas phase |
| $Y_{O,\infty}$ | N/A | Mass fraction of oxidizer in gas phase |
| $c_{p,\infty}$ | $J/kg/K$ | Gas-phase specific heat |
| $\mu_g$ | $kg/m/s$ | Gas-phase viscosity |
| $G_{in}$ | $W/m^2$ | Incident radiation |

# 4.3. COUPLING THE LAGRANGIAN AND EULERIAN FIELDS

The previous section provides a description of the particle evolution given a gas-phase environment. In this section the means by which the gas-phase environment for a particle is determined from an Eulerian solution presumed to use a control-volume or similar approach. Following this, the effect of the particle field on the Eulerian field is described. Finally, to ensure that the coupled evolution proceeds in a physically realistic manner, it is necessary to identify limits on the time step size.

## 4.3.1. Gas-phase environment for parcels

When a parcel of particles spans more than a single control volume, it is appropriate to employ a weighted average of the gas properties in the control volumes spanned by the parcel. The average is weighted by the number of particles in a given control volume, which is obtained from $f_\sigma(\mathbf{x}; \mathbf{x_o}, t)$ defined in Eqn. 4.14. Thus, the average value of a gas-phase variable, $\phi$, for a parcel is

$$\bar{\phi} = \frac{\int_V \phi(\mathbf{x}, t) f_\sigma(\mathbf{x}; \mathbf{x_o}, t) d\mathbf{x}}{\int_V f_\sigma(\mathbf{x}; \mathbf{x_o}, t) d\mathbf{x}} \tag{4.74}$$

where the integral volume may span more than one control volume. This procedure is employed for all gas phase variables that appear in the droplet evolution equations in 4.2. For transport properties (i.e.

various diffusion coefficients), the gas phase properties employed are evaluated using the '1/3 rule' as indicated for viscosity in Eqn. 4.6.

### 4.3.2. Effect of the particle phase on the gas phase

The source terms provided in the previous section show how the gas-phase properties affect the conservation of mass, momentum, and energy for the particles. From Newton's third law, the action of the gas phase on the particles must be balanced by an action of the particles on the gas phase. This action is determined directly from the change in the state of the particle phase as described in the following.

The source of the mass for a given control volume, $V_c$, located at $(\mathbf{x}, t)$ is determined by summing the changes in the masses of all the particles in that control volume over the gas-phase time step $\delta t$

$$S_{mass}(\mathbf{x}, t) = -\frac{1}{V_c}\sum\nolimits_P \left[\frac{m_p(t+\delta t) - m_p(t)}{\delta t}\int_{V_c} f_\sigma(\mathbf{x}; \mathbf{x_o}, t)d\mathbf{x}\right]. \tag{4.75}$$

Here, the summation is presumed to occur over the $P$ parcels that contribute to the control volume at $(\mathbf{x}, t)$, and the addend in the square brackets corresponds to the mass change and particle number distribution in each of the $P$ parcels. The similar equation for species concentration incorporates the conversion of the mass of the particle component to the mass of the gas-phase component. For pure evaporation, the conversion is trivial, but for combustion where the fuel evaporation corresponds to oxidizer consumption and the product formation, the expressions become complicated. The general expression for the species mass source term is

$$S_{mass,i}(\mathbf{x}, t) = -\frac{1}{V_c}\sum\nolimits_P \left[\frac{\nu_i W_i}{W_F}\frac{m_p(t+\delta t) - m_p(t)}{\delta t}\int_{V_c} f_\sigma(\mathbf{x}; \mathbf{x_o}, t)d\mathbf{x}\right]. \tag{4.76}$$

Here, $\nu_i$ is the number of moles of species $i$ produced when a mole of the particle component evaporates, and $W_i$ is the molecular weight of species $i$. The particle component is presumed to be the fuel, denoted with the subscript $F$, and for pure evaporation reduces to equation 4.75.

Similarly, the source term for the $j$-momentum equations is determined by summing the changes in the particle momentum over all of the particles in a control volume.

$$S_{mom,j}(\mathbf{x}, t) = -\frac{1}{V_c}\sum\nolimits_P \left[\left(\frac{m_p(t+\delta t)u_{p,j}(t+\delta t) - m_p(t)u_{p,j}(t)}{\delta t} - m_p(t)g_i\right)\int_{V_c} f_\sigma(\mathbf{x}; \mathbf{x_o}, t)d\mathbf{x}\right] \tag{4.77}$$

Note that the last term in Eqn. 4.77 describes the effect of gravitational acceleration on the particle field.

The energy source term must account for the enthalpy of vaporization and heat of combustion associated with the particles as well as the particle heating. It must also account for the change in the enthalpy of the gas associated with any gases that evaporated or condensed. In addition, it is necessary to separate out the contributions associated with radiative transport since these do not necessarily affect

the local control volume, but are transported over length scales determined by the radiative transport equation. The enthalpy source term is

$$S_{enthalpy}(\mathbf{x},t) = -S_{rad} - \frac{1}{V_c}\sum\nolimits_P \left[\frac{m_p(t+\delta t)\left[h_p(t+\delta t) - h_p(t)\right]}{\delta t}\int_{V_c} f_\sigma(\mathbf{x};\mathbf{x_o},t)d\mathbf{x}\right] \quad (4.78)$$

$$-\frac{1}{V_c}\sum\nolimits_P \left[\frac{\left[m_p(t+\delta t) - m_p(t)\right]\sum_i \nu_i W_i h_i(T_g)/W_F}{\delta t}\int_{V_c} f_\sigma(\mathbf{x};\mathbf{x_o},t)d\mathbf{x}\right] \quad (4.79)$$

$$-\frac{1}{V_c}\sum\nolimits_P \left[\frac{\left[m_p(t+\delta t) - m_p(t)\right]\left(q_{comb} - h_{vap}\right)}{\delta t}\int_{V_c} f_\sigma(\mathbf{x};\mathbf{x_o},t)d\mathbf{x}\right] \quad (4.80)$$

where the radiative transport source term is

$$S_{rad}(\mathbf{x},t) = \frac{1}{V_c}\sum\nolimits_P \left[Q_{rad}\int_{V_c} f_\sigma(\mathbf{x};\mathbf{x_o},t)d\mathbf{x}\right], \quad (4.81)$$

which provides the interface between the particle field and the radiation transport equation. In general, the solution of the radiant transport equation requires an absorption coefficient and a contribution to the emission. These must be defined in such a way that they agree with the radiation absorbed and emitted by the particle field. To do this, an energy balance for the particle field, but in an Eulerian frame, is employed. Within the energy conservation equation, the radiant source term appears as a divergence of the radiation heat flux $\nabla \cdot q_R$. This divergence of the radiation heat flux can be related to the radiation intensity and the radiation emitted by [117]

$$\nabla \cdot q_R = \langle 4\kappa_{net}\sigma T_{net}^4\rangle - G_{in}\langle\kappa_{net}\rangle \quad (4.82)$$

where the subscript $net$ indicates that contributions from the particles must be combined with those from gases, smoke, and anything else that participates with the radiative field. The angular brackets indicate that the quantities on the right hand side must be defined based on the appropriate summation or averaging process over all of these participating media, which may have varying temperatures. In order to balance the radiant energy flux in and out of the particle with their contribution to Eqn. 4.82, it is necessary to separate these particle contributions from the net radiant source term. To leading order, this can be done in an additive manner so that the first term in Eqn. 4.82 is split as $\langle 4\kappa_{net}\sigma T_{net}^4\rangle = \langle 4\kappa_p\sigma T_p^4\rangle + \langle 4\kappa_{others}\sigma T_{others}^4\rangle$ and the second term is split with $\langle\kappa_{net}\rangle = \langle\kappa_p\rangle + \langle\kappa_{others}\rangle$. Here the subscript $p$ indicates the contribution associated with the particle field and the subscript $others$ indicates all other contributions (gases, soot, etc.). Corrections to this leading order approximation are related to the optical thickness of the control volumes over which the radiation solve occurs. These corrections arise because a portion of the intensity is absorbed within the control volume. As long as the control volumes are all optically thin, then this correction is not important, but it must be accounted for in the radiation solve if that term is important.

Separating out only the contribution of the particle field to the radiation solve gives

$$(\nabla \cdot q_R)_p = \langle 4\kappa_p\sigma T_p^4\rangle - G_{in}\langle\kappa_p\rangle \quad (4.83)$$

where the angular brackets indicate that the quantities on the right hand side must be defined based on the appropriate summation over all of the particles in the control volume. In order to ensure energy conservation between the Lagrangian solution of the particle field and the eulerian solution of the

radiation transfer and energy equations, these appropriate sums are obtained by integrating over the particle field in a given control volume as in Eqn. 4.81. Here, the emission and absorption contributions of $Q_{rad}$ are separated

$$S_{rad}(\mathbf{x},t) = \frac{1}{V_c}\sum\nolimits_P \left[4\pi r_p^2 \alpha \sigma T_p^4 \int_{V_c} f_\sigma(\mathbf{x};\mathbf{x_o},t)d\mathbf{x}\right] - \frac{1}{V_c}\sum\nolimits_P \left[\pi r_p^2 \alpha \int_{V_c} f_\sigma(\mathbf{x};\mathbf{x_o},t)d\mathbf{x}\right] G_{in} \tag{4.84}$$

so that, equating the first and second terms on the right-hand side of Eqns. 4.83 and 4.84 gives

$$\langle 4\kappa_p \sigma T_p^4 \rangle = \frac{1}{V_c}\sum\nolimits_P \left[4\pi r_p^2 \alpha \sigma T_p^4 \int_{V_c} f_\sigma(\mathbf{x};\mathbf{x_o},t)d\mathbf{x}\right] \tag{4.85}$$

and

$$\langle \kappa_p \rangle = \frac{1}{V_c}\sum\nolimits_P \left[\pi r_p^2 \alpha \int_{V_c} f_\sigma(\mathbf{x};\mathbf{x_o},t)d\mathbf{x}\right] G_{in} \tag{4.86}$$

This information is crucial because it defines an energy conservation interface between the particle field and the radiation field.

The source terms defined in Eqns. 4.75 through 4.80 should be added directly to the gas-phase conservation equations in the manner appropriate for the chosen numerical method. In the above relations, it was presumed that the dimensions of the conservation equations solved are those of density, density time velocity, and density time enthalpy. The radiant flux source term in Eqn. 4.81 should similarly be added to the radiative transport equation, but in the consistent manner indicated in the above paragraph.

### 4.3.3. Time step control

The Lagrangian and Eulerian fields are advanced using an explicit operator splitting approach. The particles are presumed to be advanced using an ordinary differential equation solver capable of handling a stiff system. The system can be stiff because the magnitude of the forcing function, the right hand side, of the various particle evolution equations can vary over orders of magnitude. While the particle state is evolved, the gas-phase state is presumed to be constant, except as described in the following paragraph. The source terms indicated in the previous section are accumulated during the particle evolution, and then they are applied to the gas-phase conservation equations as it is advanced.

There are certain requirements that limit the particle evolution time step. Particle should not move more than the length of the control volume side without re-evaluating the gas-phase state using the methods described in 4.3.1. Particles should not evolve for longer than the eddy interaction time defined as the lesser of times defined in Eqn. 4.12 and 4.13 without determining a new value for $u_{g,i}$ in Eqn. 4.10 and 4.11. These requirements do not necessarily limit the gas-phase evolution time step.

There are also limitations to the gas-phase evolution time step imposed by the particle evolution. The source of this limitation is the requirement that the treatment of the gas-phase state as constant during the particle evolution not lead to nonphysical or inaccurate results. For example, if the particles transferred almost all of their momentum to the gas phase in a single time step (because they were small, for example), then a problem could arise. Specifically, if the subsequent momentum source term were

large, then the gas could be accelerated to velocities that exceed the initial particle velocities. In the subsequent time step, the particles would accelerate and the solution procedure could thereby destabilize. There are two means of avoiding such problems. One approach is to employ an iterative implicit coupled advance of the Eulerian and Lagrangian state. Such an approach would depend on the specific algorithms employed for the Eulerian solver. The approach that will be discussed here is a limitation on the time step size for the coupled Eulerian-Lagrangian system. This limitation is based on the idea that the change in the Eulerian state relative to the Lagrangian state should not be too dramatic. This not only provides stability, but helps limit numerical errors.

For mass conservation, the limitation on the time step is based on the idea that the mass added to the cell should not dramatically affect the pressure (or whatever variable changes to allow additional mass to entry the control volume gas phase). With $S_{mass}$ in units of density per unit time, the appropriate limitation on the time step is a time step that leads to no more than a change of $\delta_\rho \rho$ in $\rho$.

$$dt_{max} < \delta_\rho \rho / S_{mass} \tag{4.87}$$

That is, the fractional change in the density is limited to $\delta_\rho$. It is expected that $\delta_\rho$ is substantially smaller than 0.1 is appropriate, but this is subject to evaluation and will depend somewhat on the details of a given simulation. for the conservation of species, similar expressions apply, but with the second term that provides a sort of absolute tolerance in addition to the relative tolerance indicated above

$$dt_{max} < \delta_{Y_i} \rho Y_i / S_{mass,i} + \eta_{Y_i} \rho / S_{mass,i}. \tag{4.88}$$

Intuition suggests that the limitation of $\delta_{Y_i}$ do not need to be as severe as those on $\delta_\rho$, for stability and that $\eta_{Y_i}$ should be substantially smaller than typical magnitudes for $Y_i$ for accuracy. Again, the specific values will depend on how sensitive the system is on $Y_i$. For momentum, a similar expression is provided with relative and absolute tolerances

$$dt_{max} < \delta_u \rho u_i / S_{mom,i} + \eta_u \rho / S_{mom,i} \tag{4.89}$$

where $\delta_u u_i$ and $\eta_u$ are indicative of the acceptable uncertainties in the velocity field. Time-step control based on the enthalpy exchange is easiest to think of in terms of temperature

$$dt_{max} < \delta_h \rho c_{p,g} T_g / S_{enthalpy} + \eta_h \rho c_{p,g} / S_{enthalpy} \tag{4.90}$$

where $\delta_h T_g$ and $\eta_h$ are indicative of the acceptable uncertainties in the temperature field [2] .

### 4.3.4. Particle-surface interactions

Particles interacting with a surface can (1) bounce off of the surface and return to the flow with a different trajectory, (2) stick to the surface and remain as a deposit or (3) shatter so that smaller droplets

[2] Instead of using $S_{enthalpy}$ that includes both the chemical and sensible enthalpy changes, it is better to use a measure of the change in the sensible enthalpy source term (see Eqn 4.80) where the next to the last set of brackets includes only the sensible contribution of the species heating and not the chemical enthalpy contribution.

are formed that leave the surface with various trajectories. The appropriate behavior depends primarily on the ratio of the droplet kinetic energy to the surface energy as determined by the Weber number

$$We = \frac{\rho_p d_p |\mathbf{u}_{p,i}|^2}{\sigma} \tag{4.91}$$

where $\sigma$ is the surface tension of the condensed phase. A complete model description is available in [122] with criteria for droplet sticking and bouncing. That paper did not address droplet shattering in detail and the droplet shattering model is provided in the following paragraph.

Droplets will shatter if the criteria

$$We^{0.5} Re_p^{0.25} > K_{crit} = 57.7 \tag{4.92}$$

is satisfied, which occurs for relatively large droplets traveling at relatively high velocities. Satellite droplets are presumed to form with uniform sizes given by

$$d_{p,shatter} = max\left(\frac{7.9 \cdot 10^{10} \sigma We^{1.4}/Re^{2.8}}{\rho_p |\mathbf{u}_{p,i}|^2}, d_p\right) \tag{4.93}$$

where the second term in the max function ensures that the empirical relationship given as the first term does not exceed the original diameter.

If a particle sticks to the surface, it is necessary to track the mass and energy addition to that surface through a field variable added to the solid object surface set. Mass addition should be tracked on a mass per solid-element surface area basis. Similarly, the energy deposited on a surface should be tracked based on $c_{v,p}(T_p - T_{surface})$. Erikson and Gill have developed and implemented a model to provide such an interface for a Calore surface [123].

## 4.4. VERIFICATION OF PARTICLE EVOLUTION EQUATIONS

It is generally necessary to check the implementation of the equations described in 4.2 to insure that they produce the expected effect. This process is referred to as verification. There are several stages of verification, many of which are focused on software design details, but many of which are intimately linked with the physics model implementation. In this section, a series of verification tests is presented that provides a test for the correct implementation of the models described in the previous sections. For the particle transport models developed here, this is accomplished by taking certain asymptotic limits of the evolution equations for which an analytical solution exists and insuring that the particle evolution approaches that solution as the particle properties approach the appropriate limiting values. For example, in the limit of zero Reynolds number, certain behavior is expected and the particle should approach that behavior as the diameter approaches zero. The limiting behavior is based on the limiting behavior of the model equations and is not based on matching particular experimental data, although many of the limiting behaviors correspond to well known phenomena. The key objective of the verification process here is to insure that the equations are satisfied, so that when numerical examples are given, nominal values for material and transport properties are employed.

### 4.4.1. Verification of Particle Momentum and Trajectories

In this section the solutions of Eqns. 4.3 through 4.9 are tested. For all of these tests, it is presumed that the source terms indicated in 4.75 through 4.80 are set to zero so that the gas velocities and other properties are fixed. Setting the gas-phase source terms to zero is referred to as one-way coupling because the gas phase affects the particular phase, but not vice versa.

#### *4.4.1.1. Terminal Velocity*

Falling particle will reach a terminal velocity if the gas velocity is held fixed that is given by

$$u_{p,i}|_{t\to\infty} = u_{g,i} + \frac{8(\rho_p - \rho_g)r_p g_i}{3\rho_g C_D |u_{g,i} - u_{p,i}|}. \tag{4.94}$$

Because this terminal velocity includes the Reynolds number dependence of the drag coefficient, the full range of the drag coefficient can be tested by changing, for example, the particle diameter. Note that the slip velocity magnitude does appear in the drag coefficient for a range of values through the Reynolds number. For these Reynolds numbers ($Re_p < 1000$), the drag coefficient can be replaced by the first relationship in Eqn. 4.4 to give

$$u_{p,i}|_{t\to\infty} = \frac{2(\rho_p - \rho_g)r_p^2 g_i}{9\mu_g(1 + Re_p^{2/3}/6)}. \tag{4.95}$$

For $Re_p > 1000$, using the second relationship from Eqn. 4.4 in Eqn. 4.94 yields simply

$$u_{p,i}|_{t\to\infty} = u_{g,i} + \sqrt{\frac{8(\rho_p - \rho_g)r_p g_i}{3\rho_g(0.424)}}. \tag{4.96}$$

The full range of terminal velocities obtained by varying $r_p$ is shown in Fig 4.4-1. Note that while the velocities are continuous at $Re_p = 1000$, there is a discontinuity in the rate of change of the velocity with diameter of Reynolds number at this point.

#### *4.4.1.2. Small Reynolds number*

In the limit of small Reynolds number the solution of Eqn. 4.3 approaches that given by Eqn. 4.7 and 4.8 where $C_D$ simplifies to $24/Re_p$ so that $\tau_p$ approaches a constant value even as the velocity changes. In this case, the analytic solution of Eqn. 4.3, where the gas velocity is fixed, is

$$u_{p,i}|_{t\to\infty} = \left(u_{g,i}^0 + \frac{(\rho_p - \rho_g)}{\rho_p}\right)\left(1 - e^{-t/\tau_p}\right) + u_{p,i}^0 e^{-t/\tau_p} \tag{4.97}$$

where the superscript 0 indicates the initial conditions. This limit can be tested by approaching zero Reynolds numbers with successively smaller diameters. The solution to Eqn. 4.9 is obtained is similarly obtainable by integration of 4.97 once and is

$$x_{p,i} = x_{p,i}^0 + \left(u_{g,i}^0 + \frac{(\rho_p - \rho_g)}{\rho_p} g_i \tau_p\right)\left(1 - e^{-t/\tau_p}\right) + u_{p,i}^0 \tau_p \left(1 - e^{-t/\tau_p}\right) \tag{4.98}$$

**Table 4.4-1.. Input parameters related to the particle trajectory verification**

| $d_p$ | $\rho_p$ | $\mu_g$ | $\rho_g$ | $g_i$ | $\tau_p$ | $u_{p,i}\|_{t\to\infty}$ | $Re_p\|_{t\to\infty}$ |
|---|---|---|---|---|---|---|---|
| $3\cdot10^{-2}$ | 10 | 0.01 | 1.0 | $-1.0$ | $5\cdot10^{-2}$ | $5\cdot10^{-2}$ | $1.5\cdot10^{-1}$ |
| $3\cdot10^{-3}$ | 10 | 0.01 | 1.0 | $-1.0$ | $5\cdot10^{-4}$ | $5\cdot10^{-4}$ | $1.5\cdot10^{-4}$ |
| $3\cdot10^{-4}$ | 10 | 0.01 | 1.0 | $-1.0$ | $5\cdot10^{-6}$ | $5\cdot10^{-6}$ | $1.5\cdot10^{-7}$ |

Using the values indicates in table 4.4-1, and with no heat and mass transfer, particles were evolved from rest using Fuego. The resulting particle trajectories are compared with those predicted in Eqns. 4.97 and 4.98, and the velocities and positions are plotted in figure 4.4-1.

#### *4.4.1.3. Turbulent dispersion*

Verification of particle dispersion is hampered by the fact that it is a stochastic and not deterministic process. Therefore, verification is conducted by checking that the asymptotic behavior is correct for large numbers of samples. In an isotropic homogeneous turbulent field, the turbulent dispersion of a large number of particles should result in the mean square displacement of particles in proportion to the effective diffusion coefficient and the evolution time. The diffusion coefficient in the limit where $\tau_e$ and not $\tau_C$ determines the eddy interaction time is proportional to $|\mathbf{u}_g + \mathbf{u}'_g - \mathbf{u}_p|^2\tau_p$ [124]. For verification purposes, the mean gas velocity is set to zero so that the effective diffusion coefficient is proportional to $k$. Note that the early time behavior is different, but that the early time behavior decays over the time scale $\tau_p$. The long-time dispersion behavior averaged over a sufficient statistical sample should be verified to follow [98]

$$\langle x^2\rangle \propto k\tau_p t \tag{4.99}$$

Such behavior should hold true in a statistical sense for both the dispersion of the mean parcel locations and for the change in the extent of the parcel itself. Because the dispersion is driven by a Gaussian process, it can be expected that statistical differences are reduced in proportion to the inverse of the square root of the number of samples.

To verify the scaling of the particle dispersion, a large number of particles were evolved using specified fluid variables. All of the mean fluid velocities are set to zero and only the turbulent kinetic energy, $k$, combined with time and length-scale information from the turbulent energy dissipation, $\epsilon$, acted on the particles. In this case, the only force on the particle is associated with $u'_{g,i}$ as defined in Eqns. 4.10 and 4.11. The parameters related to particle dispersion used in the test are given in 4.4-2. The flow was evolved for a time of 5.0 (nearly $1000\tau_p$) and simulations were carried out for 100, 1000, and $10,000$ particles. The results are shown in 4.4-2 where the linear dependence of the mean-square displacement with time is evidence for large particle samples. The large number of samples required to get convergence is noteworthy.

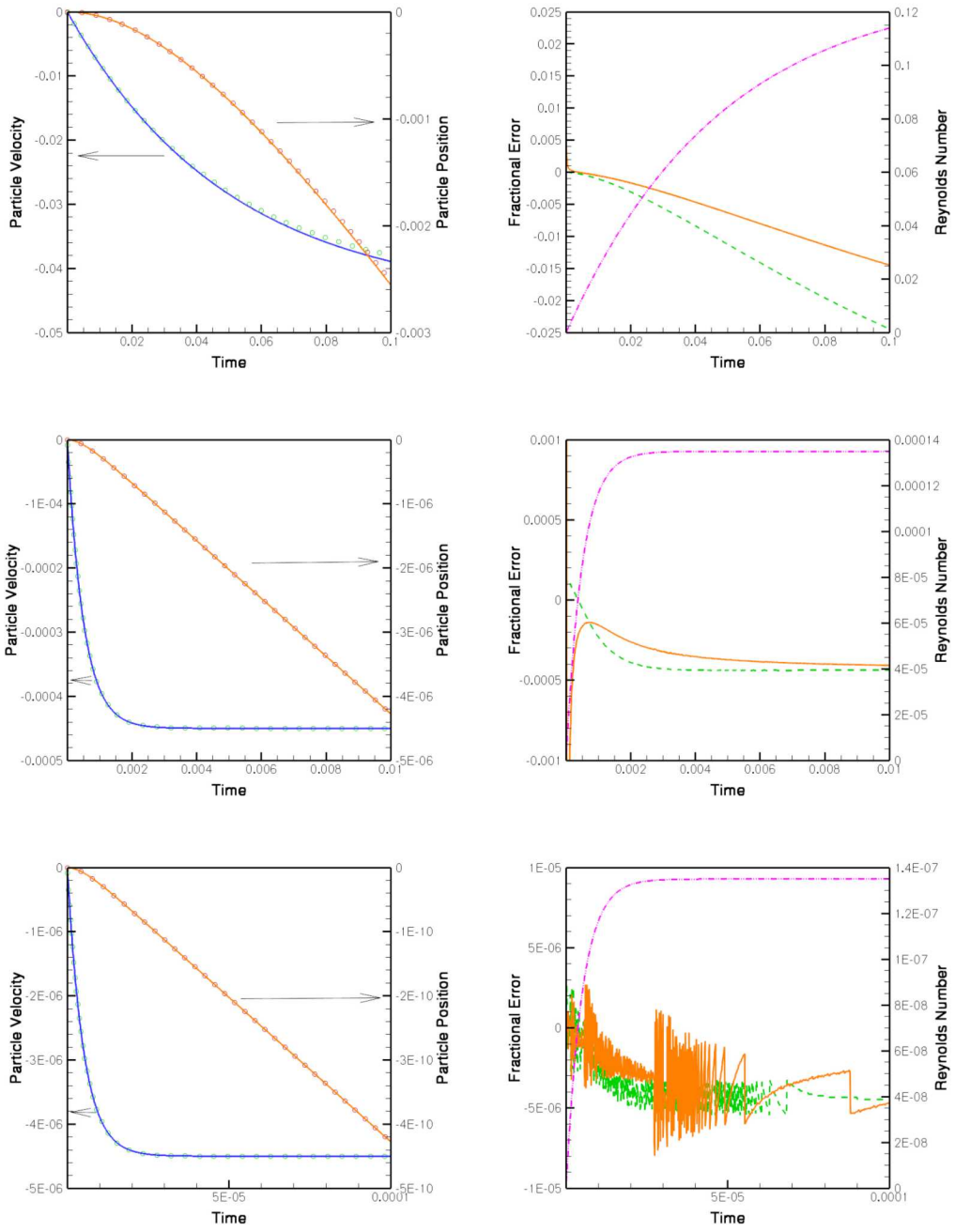


**Figure 4.4-1.. The Fuego-computed particle velocity and trajectory are compared with predictions from 4.97 and 4.98 in the left panels. The error in each prediction is shown in the right panel along with the particle Reynolds number. Parameters for the particle evolution are from 4.4-1 with $d_p$ from top row to bottom row being $d_p = 3 \cdot 10^{-2}$, $d_p = 3 \cdot 10^{-3}$, and $d_p = 3 \cdot 10^{-4}$, respectively. In the right-hand panes the solid cures represent the position error, the dashed curves represent the velocity error and the dash-dot-dot curve shows the particle Reynolds number.**

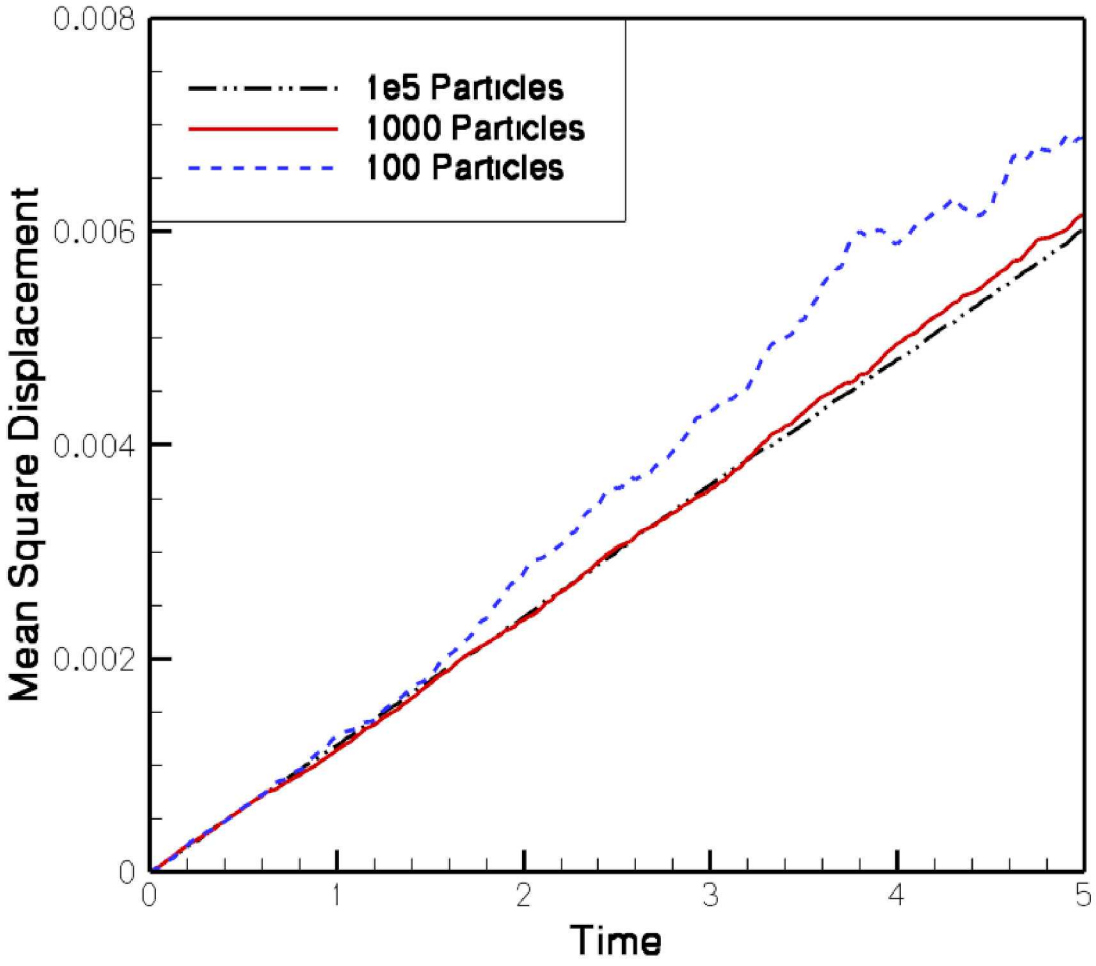


**Figure 4.4-2.. The mean square displacement of particles is shown as a function of time. Different curves show the statistical noise associated with different numbers of particles considered.**

**Table 4.4-2.. Input parameters related to the particle dispersion verification**

| Fluid properties | | Particle properties | |
|---|---|---|---|
| $k$ | 1.0 | $\rho_p$ | 10.0 |
| $\epsilon$ | 1.0 | $d_p$ | $1 \cdot 10^{-4}$ |
| $\mu_g$ | 0.01 | $\tau_p$ | $1 \cdot 10^3/18$ |

## 4.4.2. Verification of particle heat and mass transfer

There are a number of terms describing the heat and mass transfer to and from particles that must be tested. In this section, the terms that describe the particle evolution in a constant gas-phase environment will be tested. To accomplish this, the source terms indicated in Eqn. 4.75 through 4.81 are set to zero so that the gas velocities and other properties are fixed (one-way coupling).

### *4.4.2.1. Droplet heating and cooling*

If the particles have sufficiently high boiling points and enthalpies of vaporization, the evaporation rate in Eqn. 4.48 will be zero which implies $B_{T-O} = 0$. In this case, the droplet heating and cooling are governed by the equations presented in 4.2.4. There are several limits that can be tested. The most common situation that is encountered is conduction/convection dominated heating or cooling, and this can be split into high and low Reynolds number regimes. When radiant heating and cooling are the most significant processes, the system evolves according to the radiative heat flux terms resulting in different behavior.

To verify conduction dominated heating or cooling, the particle emissivities are set to zero, eliminating the radiative interaction terms. This leads to a simplification of Eqn. 4.71

$$\frac{dT_p}{dt} = \left(\frac{6}{\rho_p c_{v,p} d_p^2}\right)\left(\frac{Nu_f \lambda_f Nu_p \lambda_p}{Nu_f \lambda_f + Nu_p \lambda_p}\right)(T_g - T_p) \tag{4.100}$$

for which an analytic solution

$$\frac{T_g - T_p}{T_g - T_{p,0}} = exp\left[-\left(\frac{6}{\rho_p c_{v,p} d_p^2}\right)\left(\frac{Nu_f \lambda_f Nu_p \lambda_p}{Nu_f \lambda_f + Nu_p \lambda_p}\right)t\right] \tag{4.101}$$

is obtained if $T_g$, $Nu_f$, and $Nu_p$ are constant and $T_{p,0}$ is the initial particle temperature. To verify radiant heat flux dominated heating or cooling, the gas-phase diffusion coefficient, $\lambda_f$, can be set to zero and non-evaporating particles can be initialized with appropriately high temperatures for cooling or with an appropriate incident flux. Neglecting gas-phase conduction, Eqn. 4.71 is written

$$\frac{dT_p}{dt} = \frac{3\alpha}{\rho_p c_{v,p} r_p}(G_{in}/4 - \sigma T_p^4) \tag{4.102}$$

where the film temperature is equal to the particle temperature for $\lambda_f = 0$. In this case, the analytic solution of Eqn. 4.102 is given in implicit form

$$\frac{1}{2T_\infty^3}\left[tan^{-1}\left(\frac{T_p}{T_\infty}\right) - tan^{-1}\left(\frac{T_{p,0}}{T_\infty}\right)\right] - \frac{1}{4T_\infty^3}\left[ln\left[\left(\frac{T_p T_\infty}{T_{p,0} - T_\infty}\right)\left(\frac{T_{p,0} + T_\infty}{T_p + T_\infty}\right)\right]\right] = \frac{3\alpha\sigma t}{\rho c_{v,p} r_f}. \tag{4.103}$$

where $T_\infty = (G_{in}/4\sigma)^{1/4}$ is the temperature seen by the particle.

**Table 4.4-3.. Input parameters related to the particle heating and cooling verification**

| Fluid proprties | | Particle properties | |
|---|---|---|---|
| $k$ | 1.0 | $\rho_p$ | 10.0 |
| $\epsilon$ | 1.0 | $d_p$ | $1 \cdot 10^{-4}$ |
| $\mu_g$ | 0.01 | $\tau_p$ | $1 \cdot 10^3/18$ |

#### *4.4.2.2. The wet bulb temperature*

The wet bulb state is the state at which droplet heating is zero that occurs when the heat transfer to the droplet is perfectly balanced by the enthalpy of vaporization and no droplet heating occurs. The droplet temperature will tend to approach the wet-bulb temperature, either by being heated by excess enthalpy conducted through the surface or by being cooled when more enthalpy is used in vaporization than is conducted to the surface. The system of Eqns. 4.71, 4.48, and 4.51 with $(m_p c_{v,p}/\dot{m})dT_p/dt = 0$ in Eqn. 4.33 determines the wet bulb state. Droplets with initial temperatures set to the wet bulb temperature should evolve with no change in the droplet temperature. It is verified in the following sections that particles initially at the wet bulb temperature do not change temperature, and that droplets approach the wet bulb temperature from other initial droplet temperatures.

#### *4.4.2.3. $d^2$-Law Evaporation and Condensation*

For particles at the wet bulb temperature with no radiative losses there will be no particle heating so that $B_{T-O}$ and $B_{F-O}$ are independent of the radius. Then, if the slip velocity is negligible the particle will evaporate at a rate proportional to the diameter squared, following the well-known $d^2$-law for particle evaporation

$$\frac{d(d_p^2)}{dt} = -K. \tag{4.104}$$

To test this behavior, droplet heating should be prevented by setting the droplet temperature to the wet bulb temperature, the radiative heat transfer should be inhibited by setting the emissivity to zero, and the particle Reynolds number should approach zero in the sense that errors on the order of $Re^{1/2}$ are introduced by finite slip velocity correlations. In this case, plotting the droplet diameter squared versus time should yield a linear line with a slope given by

$$K = \frac{4\dot{m}}{\pi \rho_p d_p}. \tag{4.105}$$

To verify the $d^2$-law for particle evaporation along with the wet-bulb temperature, three water droplets are evolved in an atmosphere of humid air. The wet-bulb temperature is computed separately for those conditions to be $313.9927$, and particles are selected at that temperature and $1K$ above and below that temperature. The basic fluid and particle properties employed in the simulation are provided in 4.4-4 and the relationship between the initial particle diameter and temperature are indicated in 4.4-5. There is no flow and gravitational acceleration is not present so that there is no slip velocity maintaining the

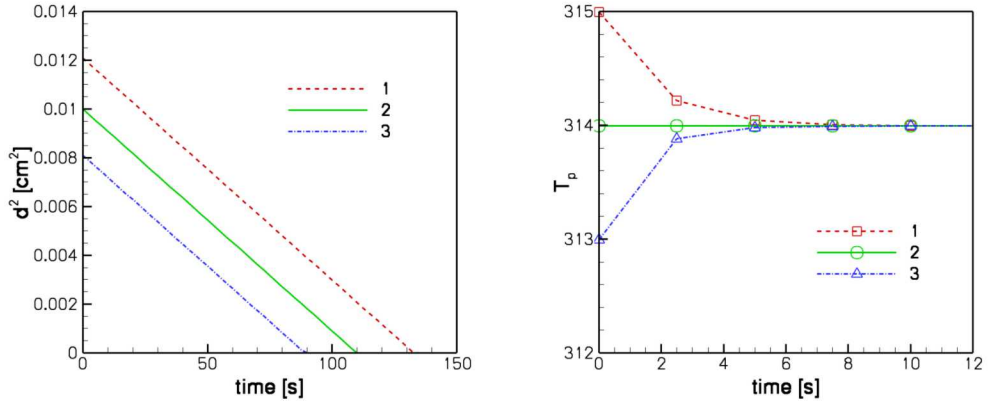


**Figure 4.4-3.. $d^2$ (left) and $T_p$ (right) as a function of time for evaporating water droplets**

**Table 4.4-4.. Input parameters related to the verification of droplet evaporation.**

| Fluid proprties | | Particle properties | | Evaporation properties | |
|---|---|---|---|---|---|
| $T_{g,in}$ | $400.0K$ | $\rho_p$ | $0.791g/cm^3$ | $h_{vap,ref}$ | $26.694 \cdot 10^{-9}$ |
| $u_{g,in}$ | $5.0g/cm^3$ | $u_{p,in}$ | $5.0cm/s$ | $T_{vap,ref}$ | $373.0K$ |
| $Y_{H_2O}$ | 0.01 | $c_{v,p}$ | $4.184 \cdot 10^7 erg/g/K$ | $P_{vap,ref}$ | $1.0Atm$ |
| $Y_{O_2}$ | 0.23 | $Y_{H_2O}$ | 1.0 | $T_{crit}$ | $647.0K$ |
| $Y_{N_2}$ | 0.76 | $Pr_f$ | 0.9 | | |
| $Nu_f$ | 2.0 | $Sc_f$ | 0.9 | | |

zero-Reynolds-number limit. For the conditions given, $K = 9.102 \cdot 10^{-5}$. Using these values the evaporation times for the three particles are 89.0, 109.9, $132.9s$ for particles labeled 1, 2, and, 3 in 4.4-5, respectively. The evaporation times predicted in Fuego are 89.1, 109.9, $123.8s$ in agreement with the predictions. The predictions assume that the deviation from the wet bulb temperature are insignificant. The evolution of $d^2$ for the particles indicated in 4.4-5 is shown in 4.4-3; the trajectories are close to linear as expected and a linear curve fit for each $d^2$ profile gives an $R^2$ coefficient of unit indicating a high degree of correlation. Also shown in 4.4-3 is the early evolution of the particle temperature. The temperature is shown to converge to the computed wet bulb temperature and to maintain itself at that temperature.

**Table 4.4-5.. Particle initial conditions for verification of droplet evaporation.**

| | 1 | 2 | 3 |
|---|---|---|---|
| $d_p[cm]$ | 0.11 | 0.1 | 0.09 |
| $T_{p,in}[K]$ | 314.9927 | 313.9927 | 312.9927 |

#### 4.4.2.4. $d^{1.5}$-law for Fast Moving Droplets

For particle droplets with large $Re_p$, the available Nusselt and Sherwood number correlations indicate that the evaporation rate should increase with $Re_p^{1/2}$ for $Re_p \gg 10$. In these cases, setting the particles to the wet-bulb temperature with no radiative losses should lead to

$$\frac{d(d_p^{1.5})}{dt} = -K_{1.5}t \tag{4.106}$$

with

$$\lim_{Re_p \to \infty} K_{1.5} = \frac{\dot{m}_o Re_p^{1/2} Sc^{1/3}}{\pi \rho_p d_p^{3/2}} \tag{4.107}$$

being the constant rate of evaporation. Note that as the particle diameter approaches zero its Reynolds number must also approach zero so that the evaporation rate should transition to a $d^2$-law behavior as it nears the fully evaporated state.

### 4.4.3. Verification of Lagrangian-Eulerian Coupling

The primary objective of this section is to verify that the source terms seen by the particle are appropriately reflected by equivalent (to the degrees appropriate) source terms in the gas-phase conservation equations. This primarily tests the source terms indicated in Eqns. 4.75 through 4.81. Because the particles affect the gas phase in the same way that the gas phase affects the particles, this is referred to as two-way coupling. The verification problems are formulated as one-dimensional problems (though the one dimension need not be aligned with the x, y, or z axis) to the maximum extent possible by employing symmetry boundary conditions on the four sides normal to the flow direction and imposing an initial uniform flow in the remaining direction. The general configuration employed is a cylindrical channel as indicated in 4.4-11. This avoids any wall effects and provides a means of identifying the transfer from inlet conditions to outlet conditions. Further, in all of these tests gravitational acceleration should be set to zero and the effects of radiation negated by setting the particle emissivity to zero.

#### 4.4.3.1. Mass conservation

The net mass flux through the system should be constant at steady state. Particles subject to evaporation will be iso-kinetically injected into the flow near the inlet (just downstream to ensure no effect of evaporation on the inlet boundary condition). These particles will be allowed to completely evaporate while they flow with the gas-phase through the domain, and the flow will be allowed to come to steady state as can be indicated by the constant exit mass flux. At steady-state, the inlet and outlet mass fluxes should be related according to the integral conservation relation

$$\int_{inlet} \rho_{in} u_{g,in} dA + \dot{M}_p = \int_{outlet} \rho_{out} u_{g,out} dA \tag{4.108}$$

where $\dot{M}$ is the mass rate of particle injection. The subscripts $in$ and $out$ indicate the state of the gas-phase fluid at the inlet to the first control volume and the exit from the last control volume. The integral across the inlet and outlet areas will average out any spatial fluctuations. To facilitate reaching steady state flow with a reasonable number of particles, it is recommended that particles with a uniform size be employed. Since particle evaporation rates typically follow the $d^2$-law behavior, it is preferable to have a relatively large number of particles injected over the evaporative lifetime of a given particle. The error in Eqn. 4.108 is expected to decrease as the frequency of particle injections is increased.

#### 4.4.3.2. Species conservation

To test the species conservation, the same verification test is employed, but the mass of the individual species is tracked.

$$\int_{inlet} \rho_{in} Y_{i,in} u_{g,in} dA + \frac{\nu_i W_i Y_{F,P}}{W_F} \dot{M}_p = \int_{outlet} \rho_{out} Y_{i,out} u_{g,out} dA \tag{4.109}$$

The mass source term is written to also account for droplet combustion as relevant. Because the total mass associated with the system changes, the mass fraction of species that do not evaporate from the particle or participate in combustion would tend to decrease; that is, the total mass flux increases while the mass flux of non-evaporating species does not increase. This can act as a second species verification test.

#### 4.4.3.3. Energy conservation

The configuration for energy conservation is the same as for mass and species conservation: one-dimensional flow with regular iso-kinetic particle injection and allowed to reach steady state. to test the coupled energy conservation, three tests are recommended to cover the range of particle behaviors. These tests would cover non-evaporating particles, evaporating but not combusting particles, and combusting particles.

The equation describing energy conservation for non-evaporating particles, assuming that the temperature of the particles equilibrates with the gas-phase before the outlet plane, is best written

$$\int_{inlet} \rho_{in} u_{g,in} h_g(T_{g,in}) dA + \dot{M}_p c_{v,p} T_{p,in} = \int_{outlet} \psi_{out} \rho_{out} u_{g,out} h_g(T_{g,in}) dA + \dot{M}_p c_{v,p} T_{eq} \tag{4.110}$$

where $h_g$ represents the gas-phase mixture enthalpy. The equilibrium temperature on the right-hand side of Eqn. 4.110 is obtained through an iterative solution of that nonlinear equation with the initial particle and gas temperatures prescribed as $T_{p,in}$ and $T_{g,in}$, respectively. Here the outflow gas-phase void fraction, $\psi_{out}$, appears in the first term on the right hand side; it is implicitly included in the other equations in this section, but is unit there based on the state assumptions. Satisfaction of the equality in Eqn. 4.110 provides a test of the first term on the right-hand side of Eqn. 4.80. Note that the approach to the equilibrium temperature follows an exponential decay of the form $exp^{-t/\tau_T}$ so that extending the domain to double the residence time from $t$ to $2t$ will tend to cause particle and gas temperatures at the

outlet to be accordingly closer to a factor of $exp^{-t/\tau_T}$. The thermal relaxation time constant for the non-evaporating particle is

$$\tau_T = \left(\frac{\rho_p c_{v,p} d_p^2}{6}\right)\left(\frac{Nu_f\lambda_f + Nu_p\lambda_p}{Nu_f\lambda_f Nu_p\lambda_p}\right). \tag{4.111}$$

This provides a additional check on the transient behavior of the system and can be used to evaluate time step control provided by Eqn. 4.90. Regardless of the time step, the enthalpy transfers should be conservative at equilibrium.

for the evaporating or combusting particles, the verification test is set up so that the evaporation of the particles is completed within the domain and that the flow reaches steady state prior to evaluation. In this sense it is the same arrangement as indicated above for mass and species conservation. These tests verify the last two terms in Eqn. 4.80. With the particles entirely evaporated, an energy balance gives

$$\int_{inlet} \rho_{in} h(T_{g,in}) u_{g,in} dA + \dot{M}_p \left[h_p(T_{p,in} + q_{comb})\right] = \int_{outlet} \rho_{out} h(T_{g,out}) u_{g,out} dA \tag{4.112}$$

The initial particle enthalpy is defined

$$h_p(T_{p,in}) = h_F(T_{p,in}) - h_{vap,ref} \tag{4.113}$$

based on the gas-phase enthalpy of the particle species denoted by the subscript $F$. Again, a nonlinear solution of this equation is required to determine $T_{g,out}$ because of the nonlinear dependence of the enthalpy on temperature. To satisfy this equation, it is necessary that the specific heat for the condense and gaseous phase of the participating species be identical; otherwise particle cooling during evaporation followed by the warming of products to the equilibrium temperature will result in energy differences that would have to be accounted for by tracking each droplet temperature in time through the domain. Further, the critical temperature, $T_{crit}$ in Eqn. 4.26, should be set essentially to an essentially infinite value of force $h_{vap} = h_{vap,ref}$.

#### 4.4.3.4. Momentum conservation

As particles with excess momentum transfer their momentum to the gas-phase flow, the net flow rate will increase. The momentum verification test described in the present section differs from the other tests in this section in the sense that a single particle or group of particles is injected at on instant and there is no continuous injection. The injected particle(s) is(are) allowed to equilibrate with the gas-phase flow and the net change in momentum is measured. For this purpose, there must be no net pressure change across the domain boundaries. Integral momentum conservation then gives the final equilibrium velocity based on

$$\int_V \rho_g u_{g,in} dV + \sum_p m_p u_{p,in} = \int_V \rho_g u_{eq} dV \sum_p m_p u_{eq}. \tag{4.114}$$

Presuming that the velocity equilibrates across the domain, $u_{eq}$ can be brought outside of the integral and summation for an explicit expression. As for the temperature equilibration test in Eqn. 4.110, there

is an exponential approach to the equilibrium value that can be used to test transient behavior if spatial fluctuations in the gas-phase velocity are not too strong. In that case, the equilibrium velocity is approached with the exponential time constant given by Eqn. 4.7 and 4.8 so that doubling the duration of the test from $t$ to $2t$ should bring the particle and gas velocities closer by a factor of $e^{-t/\tau_T}$. Note that this transient behavior can be used to evaluate the time step criteria provided in Eqn. 4.90 by assessing the error associated with liberal time steps. Regardless of the time step, though, the momentum transfer should be conservative at equilibrium.

#### 4.4.3.5. Parallel implementation

All of these tests described in the present section are implemented on both one and four processors with domains that cross multiple processors to test the passing of particle information across domain boundaries.

#### 4.4.3.6. Verification Tests for Lagrangian-Eulerian Coupling

In this section, the verification tests suggested in the previous sections are described. Verification tests for non-reacting iso-kinetic particle flows (energy conservation only), for isothermal and non-iso-kinetic particle flows (momentum conservation only) and for reacting/evaporating particle flows (mass and energy conservation) are all included. Together these test all of Eqn. 4.75 through 4.80. Not yet covered with documented verification problems is the radiation coupling in 4.81.

**Energy conservation verification** Energy conservation for non-evaporating particles was verified in Fuego using nominal parameter values for both the particles and the fluid. For the Eulerian phase, the mass, momentum, and energy equations were evolved, but not the species equations. Under these conditions, fluid properties are manually specified. The fluid properties listed in table 4.4-6 were specified as constants. Since the fluid viscosity, specific heat, and Prandtl numbers are specified, the thermal conductivity is computed from these quantities. The constant specific heat leads to a linear dependence of enthalpy on temperature, and without loss of generality the enthalpy is set to the temperature $h_g(T_g) = T_g$, corresponding to an enthalpy reference temperature of $0K$. The simulation was carried out in the cylindricalchannel.g configuration with an inlet/outlet area of 3.10583 (with unit radius, this is nominally $\pi$, but low resolution of the circular cross section leads to a smaller area) and a length of 20. Particles are inject iso-kinetically at the downstream location 1 unit from the inlet and flow 19 units to the outlet in 19 time units. With iso-kinetic flow, $Nu_f = 2$ and with zero particle viscosity $Nu_p = 6.58$ from Eqn. 4.56. The particle thermal response time from Eqn. 4.111 is $\tau_T = 3.4$ and is sufficiently small that particle thermally equilibrate while traveling the length of the channel.

There are two ways to carry out the energy conservation verification: using Eqn. 4.110 during a period in which the flow has reached steady state and integrating Eqn. 4.110 in time over the entire simulation. Both of these approaches are employed here. The surface integrals specified in the conservation equation were carried out within Ensight. To compute the flux of the particles and associated enthalpy out of the domain (last terms in 4.110), the particle deposition tracking in Fuego was employed (keywords: enthalpy_deposition_density, enthalpy_deposition_rate, etc.) and these quantities were

**Table 4.4-6.. Input parameters related to the verification of energy conservation without evaporation**

| Fluid proprties | | Particle properties | |
|---|---|---|---|
| $\rho_{in}$ | 1.0 | $\rho_p$ | 10.0 |
| $u_{g,in}$ | 1.0 | $u_{p,in}$ | 1.0 |
| $\mu_g$ | 0.01 | $\dot{M}_p$ | 0.1 |
| $c_{p,g}$ | 1.0 | $c_{v,p}$ | 100.0 |
| $T_{g,in}$ | 300.0 | $T_{p,in}$ | 1000.0 |
| $h_g(T_{g,in})$ | 300 | $\lambda_p$ | 0.1 |
| $Pr$ | 1.0 | $Pr_f$ | 1.0 |
| $Nu_f$ | 2.0 | $Nu_p$ | 6.58 |
| $\int dA$ | 3.10583 | $d_p$ | 0.01 |

also integrated over the outlet surface. The balance of the steady-state flux is described first. for the two terms on the left hand side of 4.110, the boundary conditions provided the values of $931.749$ for the fluid inlet enthalpy flux and $10,000$ for the particle inlet enthalpy. The compute equilibrium temperature is $834.1134K$ from 4.110. The outlet gas temperature was in the range of 883.4 to 834.8 with a mean of 834.1 and the exiting particles are between 832 and 835 $K$. to supplement the stead-state enthalpy flux, the integrated enthalpy flux is computed and plotted in 4.4-4. The net input enthalpy includes the initial domain enthalpy and the enthalpy of all of the injected particles over time can be compared with the enthalpy in the domain and that which has left the domain. These two quantities agree to within $0.01\%$, which is taken to be suitable (the integration of quantities within Ensight does not use the same algorithms as employed in Fuego, leading to some error). Also examined in this test is the particle mass deposition rate at the outlet. Because there is no evaporation in this scenario, the complete particle mass injected should be deposited (or pass through) the outlet. Using the keywords mass_deposition_density and mass_deposition_rate and integrating these over the outlet surface, 4.4-5 shows that the mass tracked as crossing the outlet plane matches the input particle mass, 0.1, to within statistical fluctuations.

**Momentum conservation verification** To verify the momentum transfer within Fuego, simulations were carried out with ten particles injected into a cylindrical channel. Momentum conservation for non-evaporating particles was verified using nominal parameter values for both the particle and the fluid. For the Eulerian phase, the mass and momentum were evolved, but not the species and energy equations. Under these conditions, fluid proprties are manually specified. The fluid and particle properties listed in table 4.4-6 were specified as constants. The simulation was carried out in the cylindrical_channel.g configuration with an inlet/outlet area of 3.10583 (with unit radius, this is nominally $\pi$, but low resolution of the circular cross-section leads to a smaller area), and a length of 20. The channel had symmetry boundary conditions on the side to prevent drag from affecting the momentum field. Particles are injected with an initial velocity of 10 units into a stationary fluid phase at the initial time. The particles were injected at the time origin with a diameter of 0.12, a density of 10, and a velocity of 10. The injection occurred as a line of ten particles across the channel (in the narrow

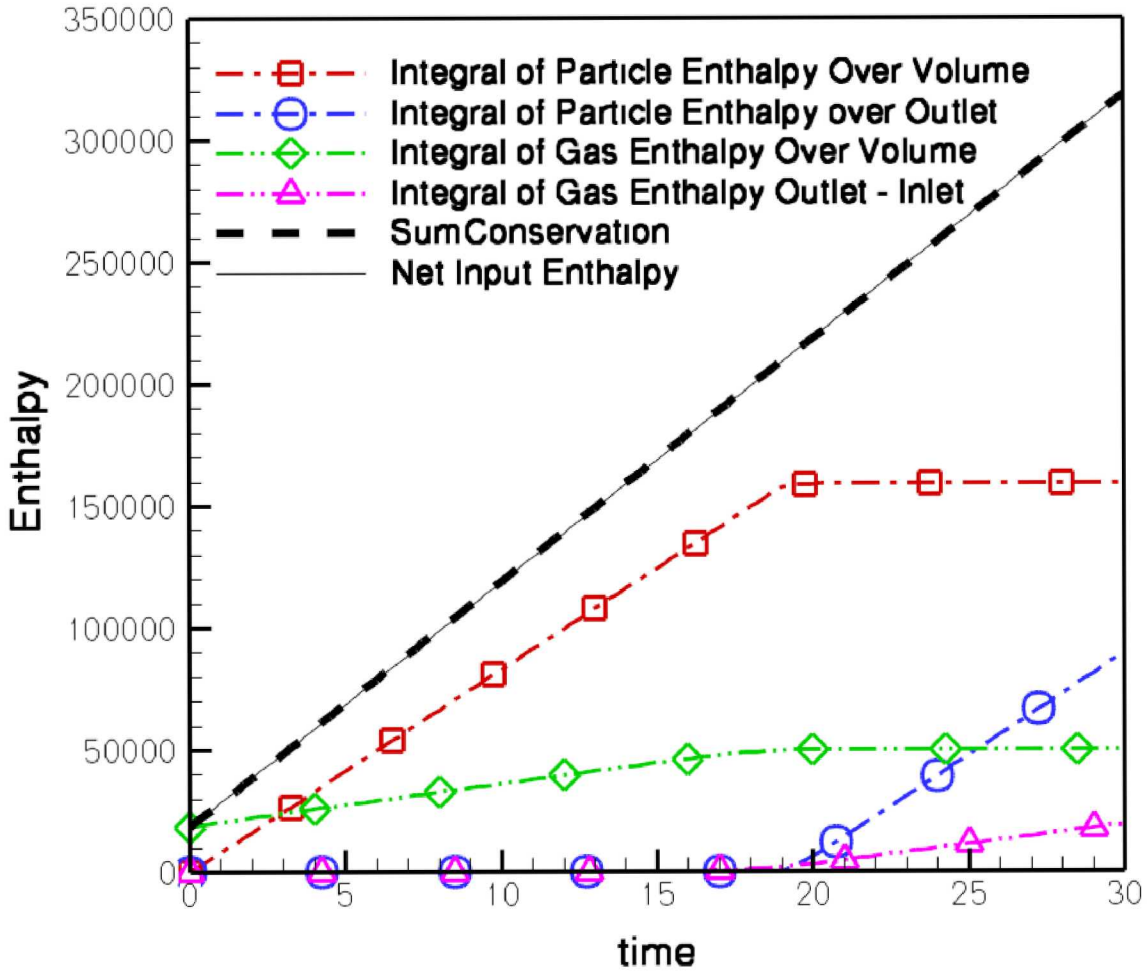


**Figure 4.4-4.. The enthalpy over time in nonreacting channel flow with hot particles injected. The net input enthalpy includes the initial domain enthalpy and the enthalpy of all of the injected particles over time. Squares show the enthalpy associated with particles in the domain. Circles show the enthalpy of the particles that have left the domain. Diamonds show the enthalpy associated with fluid in the domain. Triangles show the excess enthalpy of fluid that has left the domain (difference between the outlet enthalpy and inlet that was accounted for in the net input category). The sum of the categories indicated by symbols is shown to agree with the net input enthalpy indicating overall conservation of enthalpy.**

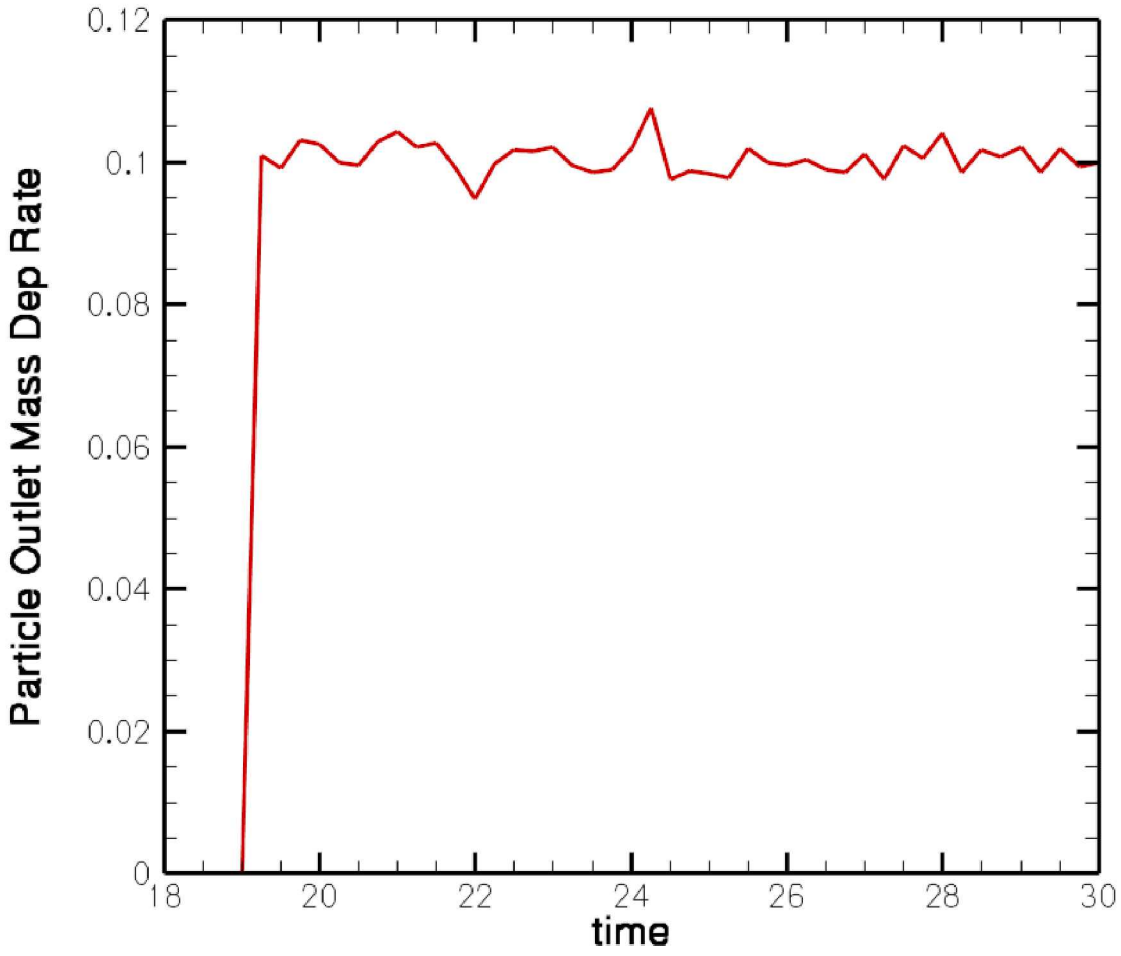


**Figure 4.4-5.. The mass deposition rate integrated over the outlet is shown as a function of time. The corresponding particle inlet mass flux is 0.1.**

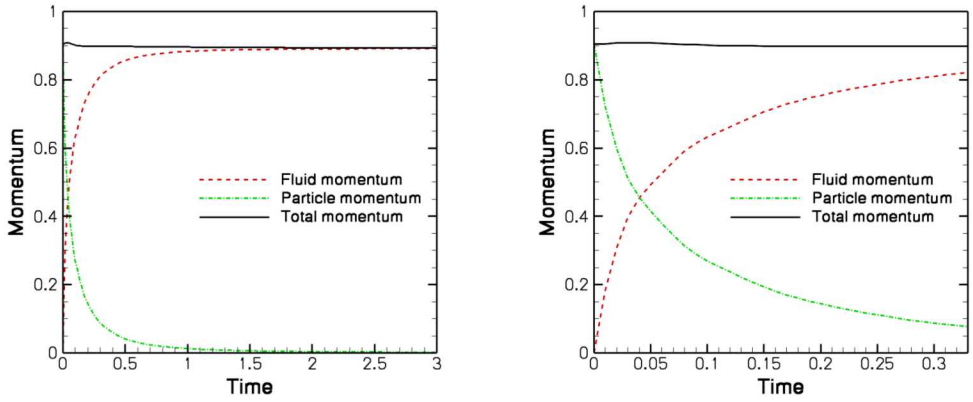


**Figure 4.4-6.. The momentum associated with the particle phase, the fluid phase, and the combined momentum shown. The right-hand panel shows the initial period in greater detail.**

direction). The particle flow direction was oriented lengthwise down the channel and the point of injection was the midpoint of the channel, 10 units from the inlet and outlet. Zero pressure boundary conditions were specified on the inlet and outlet that allowed the flow to continue in the absence of any forces (ie. particle drag).

Based on the values in 4.4-6, the initial momentum imparted by the particles is 0.904779 and the computed total momentum at the end of the Fuego simulation is 0.893059. That is, the final combined momentum is 1.3 % less than the initial momentum. The reason for this discrepancy is unclear. The final particle and fluid momentum are 0.001299 and 0.89176, respectively. The temporal evolution of the fluid, particle, and total momentum is shown in 4.4-6. There the fluid momentum has been shifted back one time step (0.01 units) to account for the staggered time-stepping algorithm: the fluid evolves based on the previous time steps' particle momentum transfer. The simulation was evolved for 30 time units, but no changes for the single precision arithmetic was observed after 21 units.

The velocity after the equilibration is computed to be $1.4379 \cdot 10^{-2}$ based on Eqn. 4.114. Carrying out the Fuego simulation results in an equilibration average velocity of $1.4356 \cdot 10^{-2}$, which is only 0.1% below the computed value. The predicted resulting momentum and velocity were computed with differing methods within Ensight: the momentum was computed using the volume integral while the velocity was computed using the spatial mean frequency. It is not clear what level of numerical error is attributable to the Ensight algorithms.

The average gas velocity had equilibrated with within 99% of its final velocity within 1.06 time units. The value of $\tau_p$ for this system is $8 \cdot 10^{-2}$ from 4.8 and the momentum and velocity equilibration times are somewhat longer than the suggested response time near the particle equilibration time of a few $\tau_p$ since the particles were not injected uniformly through the domain. The result is substantial gas-phase velocity inhomogeneities that take much longer than $\tau_p$ to dissipate.

**Mass and energy conservation verification** Mass and energy conservation for evaporating and reacting particles was verified in Fuego using fluid-phase parameter values taken from the

**Table 4.4-7.. Input parameters related to the verification of momentum conservation**

| Fluid proprties | | Particle properties | |
|---|---|---|---|
| $\rho_{in}$ | 1.0 | $\rho_p$ | 10.0 |
| $u_{g,in}$ | 0.0 | $u_{p,in}$ | 10.0 |
| $\mu_g$ | 0.01 | $N_p$ | 10 |
| $\int dA$ | 3.10583 | $d_p$ | 0.12 |

thermodynamic databases and evaluated with Cantera. For the eulerian phase, the mass, momentum, energy, and species equations were evolved. The fluid and particle properties listed in table 4.4-7 were specified.' Note that the critical temperature for water was set to a large value so that the enthalpy of the particles could be compute directly from $h_{vap,ref}$ as in Eqn. 4.113; this initial value of the particle enthalpy is also given in table 4.4-8 using the value of $h_F(T_{p,in})$ computed in Fuego, $-1.342 \cdot 10^{11} erg/g$. Since the fluid composition and enthalpy are specified as boundary conditions or evolved within Fuego, the viscosity specific heat, and thermal conductivity are computed from these quantities. The simulation was carried out in the cylindrical_channel.g configuration with an inlet/outlet area of 3.10583 (with unit radius, this is nominally $\pi$, but low resolution of the circular cross-section leads to a smaller area) and a length of 20. Particles are injected iso-kinetically at a downstream location 1 unit from the inlet flow 19 units to the outlet in 19 time units.

Net mass conservation is evaluated using Eqn. 4.108. The fluid-phase mass flux at the inlet, using the Cantera computed density of $8.776 \cdot 10^{-4} g/cm^3$, is $1.3628 \cdot 10^{-2} g/s$. The particle mass flux is $1 \cdot 10^{-4} g/s$ and combining the inlet mass fluxes gives an expected outlet mass flux of $1.3728 \cdot 10^{-2} g/s$. The value computed from the outlet using Ensight is $1.37284 \cdot 10^{-2} g/s$; there is a statistical variation in this quantity with a standard deviation of $3.7 \cdot 10^{-6} g/s$. In a similar manner, mass conservation for individual species is evaluated using Eqn. 4.109. For water, the inlet mass flux is $2.7257 \cdot 10^{-5} gH_2O/s$ and the mass injected is $1.0 \cdot 10^{-4} gH_2O/s$. The value computed at the outlet plane is $1.2657 \cdot 10^{-4} gH_2O/s$ which is approximately 1.5% below the expected value of $1.27257 \cdot 10^{-4} gH_2O/s$. The outlet water mass flux has a statistical variation associated with it characterized by a standard deviation of $5.44 \cdot 10^{-8} gH_2O/s$. While there is no source term for species like $O_2$, the inlet and outlet mass fluxes can be computed. The inlet flux of $O_2$ is $3.13466 \cdot 10^{-3} gO_2/s$ and that for the outlet is computed to be $3.13470 \cdot 10^{-3} gO_2/s$, with a standard deviation of $8.5 \cdot 10^{-7} gO_2/s$. Enthalpy conservation is evaluated using 4.112. The enthalpy flux into the domain associated with the fluid phase is $1.0477 \cdot 10^{7} erg/s$ while that associated with the particle phase is $1.568 \cdot 10^{7} erg/s$. The difference between these, $5.203 \cdot 10^{6} erg/s$, is the expected enthalpy flux at the outlet, the right hand side of 4.112. Using Ensight to evaluate the enthalpy flux at the outlet gives $5.10647 \cdot 10^{6} erg/s$; this is a 2% discrepancy, the source of which is uncertain at this point. There is a statistical variation in the outlet enthalpy flux characterized by a standard deviation of $6.38 \cdot 10^{3} erg/s$, approximately 0.04% of the total enthalpy change.

**Table 4.4-8.. Input parameters related to the verification of energy conservation with evaporation.**

| Fluid proprties | | Particle properties | | Evaporation properties | |
|---|---|---|---|---|---|
| $T_{g,in}$ | $400.0K$ | $\rho_p$ | $1.0g/cm^3$ | $h_{vap,ref}$ | $26.694 \cdot 10^{-9}$ |
| $u_{g,in}$ | $5.0g/cm^3$ | $u_{p,in}$ | $5.0cm/s$ | $T_{vap,ref}$ | $373.0K$ |
| $Y_{H_2O}$ | 0.002 | $\dot{M}$ | $1 \cdot 10^{-4}g/s$ | $P_{vap,ref}$ | $1.0Atm$ |
| $Y_{O_2}$ | 0.22 | $c_{v,p}$ | $4.184 \cdot 10^7 erg/g/K$ | $T_{crit}$ | $1 \cdot 10^9 K$ |
| $Y_{N_2}$ | 0.768 | $T_{p,in}$ | $300.0K$ | $h_p(T_{p,in})$ | $-1.568 \cdot 10^{11} erg/g$ |
| $Sc$ | 0.9 | $Y_{H_2O}$ | 1.0 | | |
| $Pr$ | 9.0 | $Pr_f$ | 0.9 | | |
| $Nu_f$ | 2.0 | $Sc_f$ | 0.9 | | |
| $\int 1dA$ | $3.10583cm^2$ | $d_p$ | $0.005cm$ | | |

**Table 4.4-9.. Verification sections and the equations tested and validated.**

| Verification subsections | Equations Covered |
|---|---|
| 4.4.1.3 | 4.3, 4.9, 4.10, 4.11, 4.12 |
| 4.4.2.1 | 4.56, 4.71, 4.72 |
| 4.4.2.2, 4.4.2.3 | 4.31, 4.33, 4.24, 4.48, 4.49, 4.51, 4.52 |
| 4.4.2.4 | 4.41 , 4.42, 4.43, 4.45 |
| 4.4.3.1, 4.4.3.2, 4.4.3.3 | 4.75, 4.76, 4.80 |
| 4.4.3.4 | 4.77 |
| Equations employed but not yet fully covered in verification tests | 4.4, 4.13, 4.27, 4.32, 4.50, 4.64, 4.81, 4.82 |

### 4.4.4. Verification summary

To summarize the verification process, the following table 4.4-9 shows the equations that are covered by verification tests described in the various sections in this document. The equations that are not fully covered are also indicated. The lack of verification coverage is only significant for the radiative terms.

## 4.5. SUMMARY

A Lagrangian model for particle transport coupled with an Eulerian solution for the gas phase is presented in detail. Models are presented for particle momentum, heat, and mass transfer, including the effects of turbulence on particle dispersion. Particular attention is paid to heat and mass transfer as these aspects are critical to the anticipated applications and they have not been well documented in other references. The heat and mass transfer models account for film temperatures that differ from particle

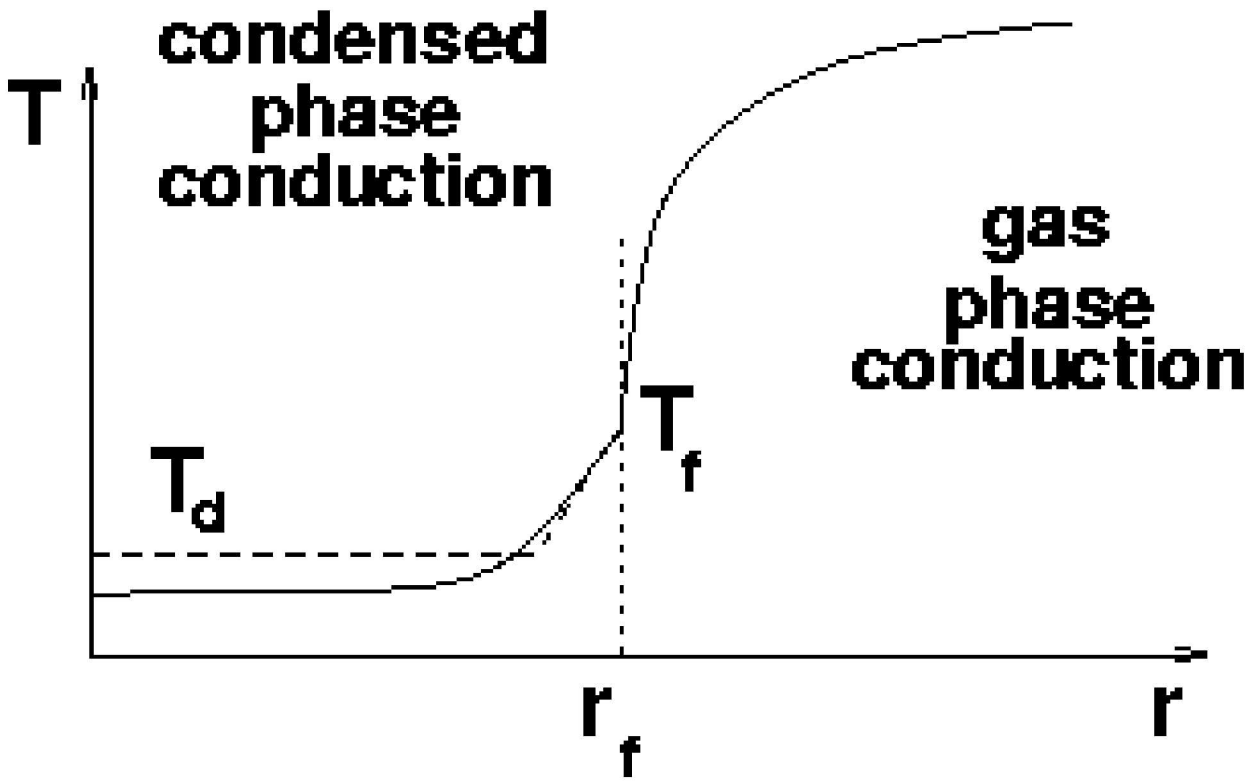


**Figure 4.4-7.. Condensed-phase conduction is approximated based on the difference between the film and mean droplet temperatures and on an estimated heat transfer coefficient that describes a boundary layer thickness. Over this boundary layer thickness, the temperature difference $T_f - T_d$ is presumed to act.**

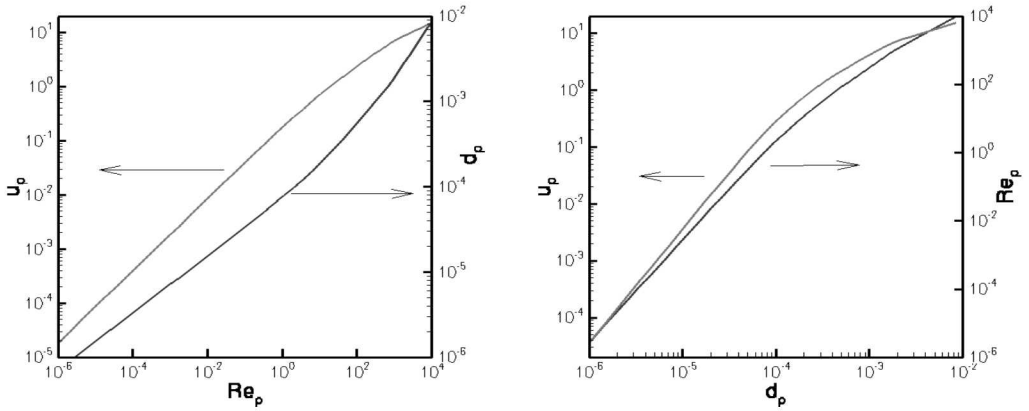


**Figure 4.4-8.. Terminal velocities for particles as a function of diameter and particle Reynolds numbers determined from 4.95 and 4.96.**

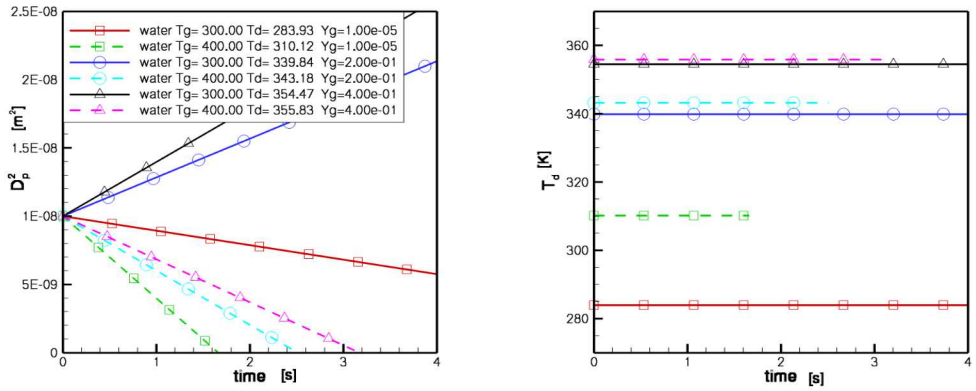


**Figure 4.4-9.. Water droplet evaporation and condensation with initial temperatures set to the wet bulb temperature. Left plot exhibits the linear $d^2$-law behavior while the right hand plot shows the droplet temperatures as constant (no heating).**

temperatures in a manner that depends on the relative magnitudes of the internal particle heat transfer, the heat transfer to the particle surface from the gas phase, the heat transfer associated with radiative fluxes, and the enthalpies associated with evaporation and combustion around the particle. Both the evaporation and condensation are permitted. A conservative algorithm for coupling the Lagrangian and Eulerian fields is presented covering mass, species, momentum, and energy transfer between two fields. Models are also specified for the interactions of the Lagrangian field with solid boundaries.

A comprehensive plan to verify the implementation of the physics models is also presented. The verification plan touches on the majority of terms in the implemented physics models. Verification tests are provide for particle momentum, trajectories, heat, and mass transfer in various limiting cases for which analytic solutions can be obtained. Verification tests to evaluate the coupling between the Lagrangian and Eulerian fields are also provided. These verification tests are based on the net conservation of mass, species, energy, and momentum.

## 4.6. EVALUATING TRANSPORT COEFFICIENTS

The droplet burning rate equations involve the area weighted diffusion coefficients as indicated in Eqn. 4.19 and subsequent equations. While the optimum method of determining the burning rate would involve the evaluation of these integrals as indicated in Eqns. 4.46 and 4.47, it is useful to estimate the effects of composition and temperature variations when such accurate evaluations are unfeasible. The kinetic theory of gases provides a starting point for such estimates, and a simplified overview of the pertinent results is provided. The single component viscosity is

$$\mu_i = \frac{5}{16}\frac{\sqrt{\pi W_k RT}}{\pi\sigma_k^2\Omega^{(2,2)*}} \tag{4.115}$$

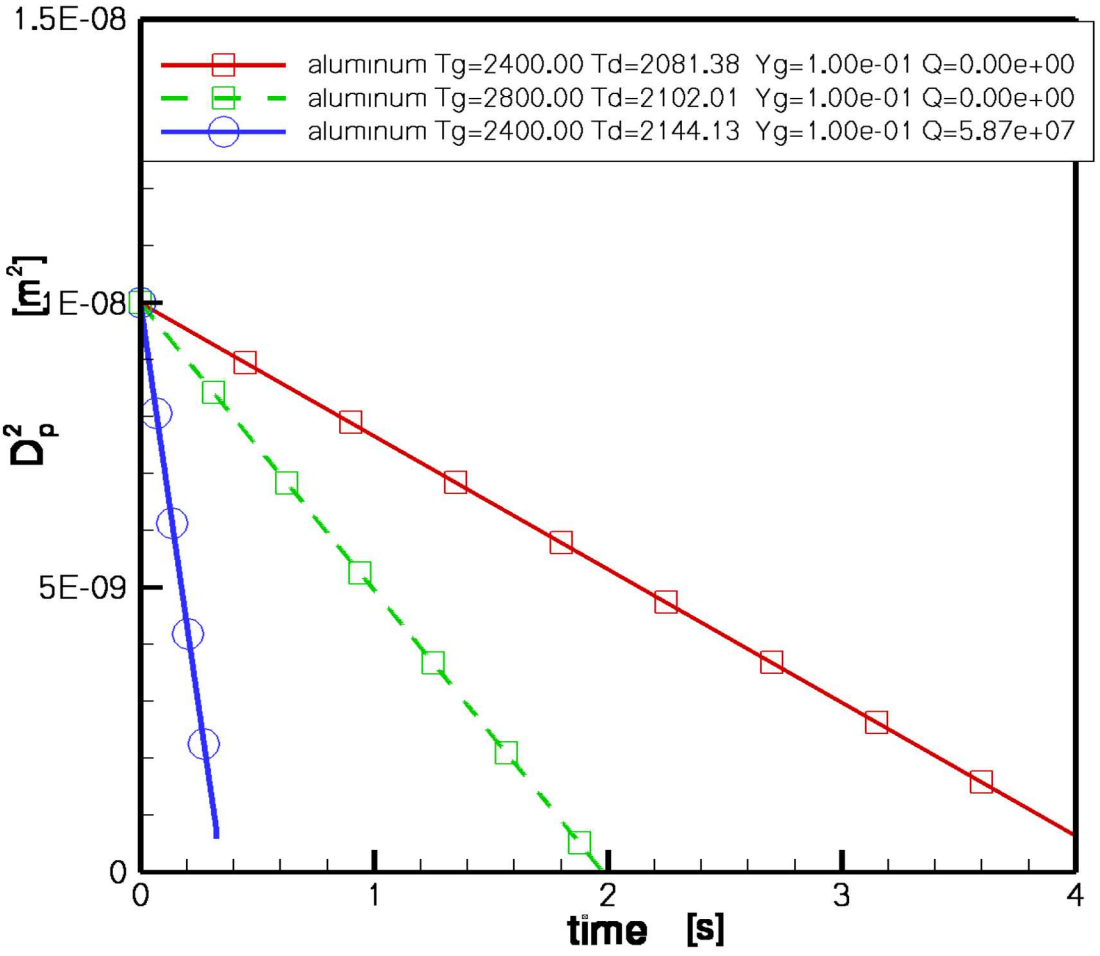


**Figure 4.4-10.. Aluminum particle evaporation with and without combustion with initial temperatures set to the wet bulb temperature showing the linear $d^2$-law behavior.**

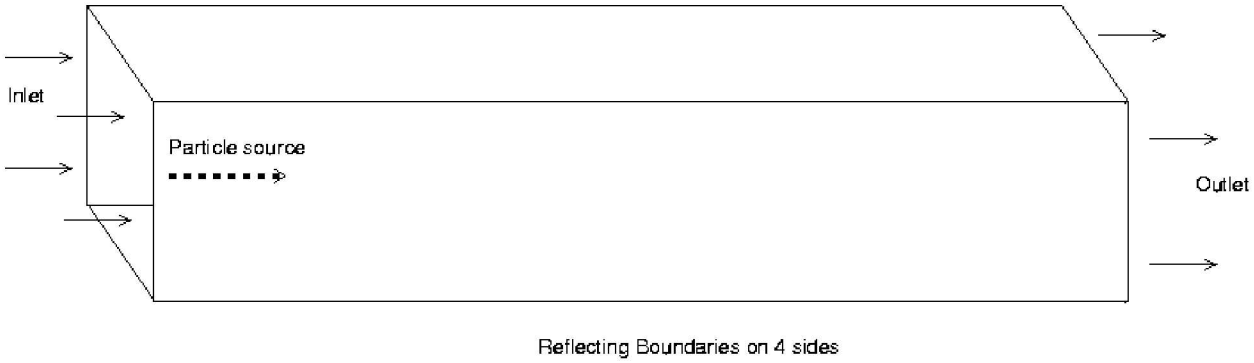


**Figure 4.4-11.. The general configuration for verification of two-way coupling. The domain should be of sufficient length that the particles equilibrate with the gas-phase flow.**

where $\sigma_k$ is the Lennard-Jones collision diameter and $\Omega^{(2,2)*}$ is the collision integral. The mixture proprties can be obtained using the Wilkes formula that averages based on mole fraction weighting to leading order. A square root dependence on the temperature is evident in 4.115, but the collision integral also includes a temperature dependence and it is found that the viscosities (and the other transport coefficients) are proportional to $T^{0.7}$, an empirical fact that is referred to as Sutherland's law. The kinetic theory of gases is only marginally successful at predicting the thermal conductivity, but the ratio of the thermal conductivity to the specific heat is closely related to the viscosity and the Prandtl number can often be approximated as constant. The binary diffusion coefficient between species $i$ and $j$ is more simply written as the product of the diffusion coefficient and the density since this removes additional pressure and temperature dependencies; this is

$$\rho D_{i,j} = \frac{3}{16} \frac{\bar{W}\sqrt{2\pi RT/W_{i,j}}}{\pi \sigma_{i,j}^2 \Omega^{(1,1)*}}. \tag{4.116}$$

Here the reduced mass and the reduced cross sections are $W_{i,j} = W_i W_j/(W_i + W_j)$ and $\sigma_{i,j}^2 = (\sigma_i + \sigma_j)^2$.

# 4.7. LAGRANGIAN PARTICLE CAPABILITIES

## 4.7.1. Lagrangian Particle Spray: Diameter Cutoffs

The Fuego Lagrangian particle spray capability has a feature which allows an upper (high) and lower (low) size (diameter) cutoff to be set for particles inserted with a specified distribution (normal, normal mass, etc.). For distribution types with infinite tails like the standard normal distribution, the particle spray can select particle sizes small enough that they do not appear in the application of interest or so large that the assumption of the dilute spray model, inherent to the Fuego Lagrangian particle implementation, is violated. In specific applications where particles experience energetic chemical reactions, such as propellant fires, particles below a certain size range react quickly and disappear without the need to resolve their dynamics. The diameter cutoff feature allows the analyst to use

standard distribution types while avoiding undesired particle size ranges. When diameter cutoffs are used, the particle pdf is adjusted accordingly to account for the lack of contribution from particle sizes outside the cutoff limits. The adjusted particle pdf is:

$$pdf_{new}\left(d\right) = pdf_{original}\left(d\right) H\left(d - d_{low}\right) H\left(d_{high} - d\right) / \int_{d_{low}}^{d_{high}} pdf_{original}\left(d\right) \tag{4.117}$$

where $pdf_{original}(d)$ is the original, uncutoff particle size pdf, $pdf_{new}(d)$ is the new particle pdf including low ($d_{low}$) and high ($d_{high}$) particle size cutoffs, the integral is take on the original particle pdf with these limits, and $H$ is the heaviside step function. This treatment properly normalizes $pdf_{new}(d)$. Figure 4.7-1 illustrates this for the case of a normal distribution of particle diameters ($< d >= 0.5, \sigma = 0.1$) with and without diameter cutoffs at $d = 0.3$ and $d = 0.65$. Figure 4.7-2 shows a section of a Fuego input deck utilizing the diameter cutoff functionality.

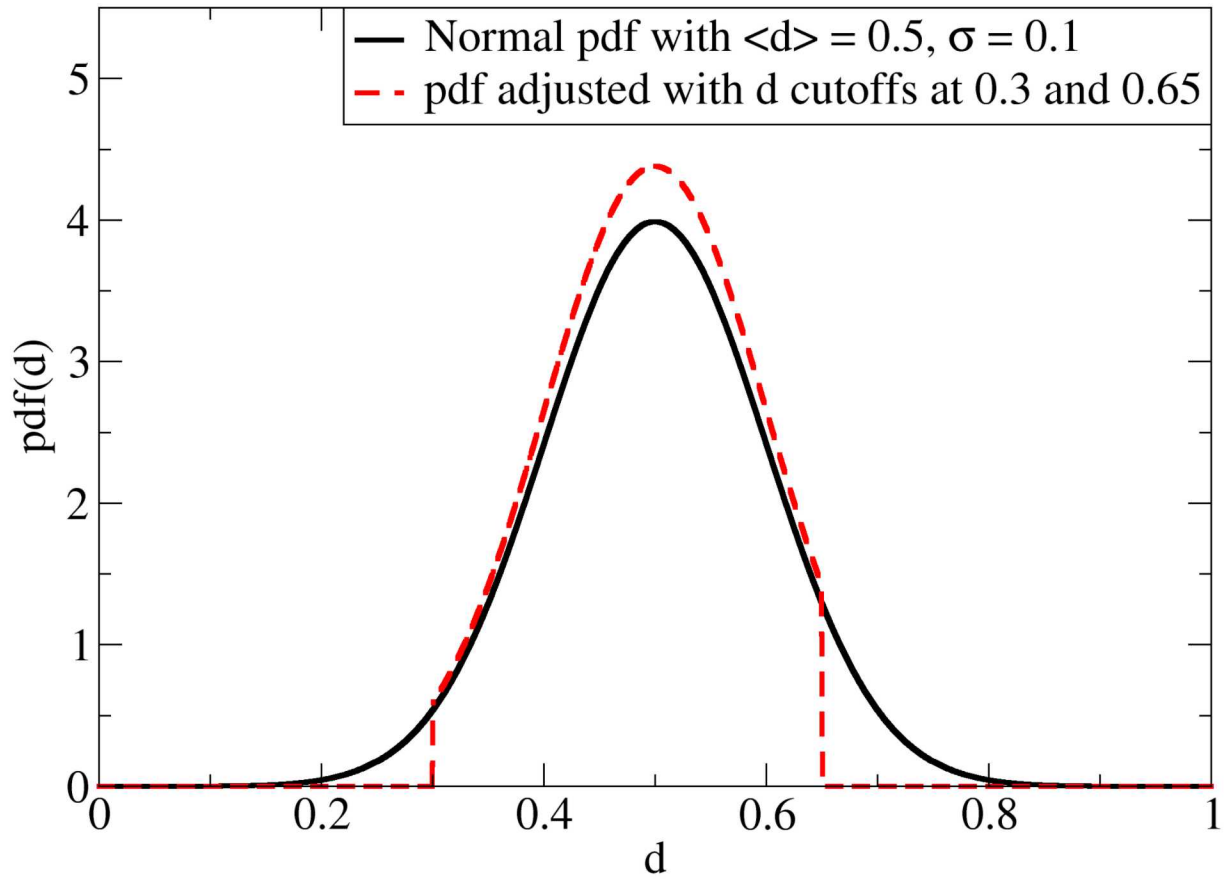


**Figure 4.7-1.. Particle size (diameter) distribution for Lagrangian particle spray with and without diameter cutoffs set at** $0.3$ **and** $0.65$

```
BEGIN PARTICLE SPRAY spray1
    PARTICLE DEFINITION = solid_particles
    #Spray Geometry
      CENTER = -100.0, 0.0, 0.0
      NOZZLE RADIUS = 10.0
      NORMAL VECTOR = 1 0 0
    #Size Distribution Parameters
      MASS_FLOW_RATE = 900.0 # g/s
      PARTICLE DIAMETER DISTRIBUTION TYPE            = NORMAL
      PARTICLE DIAMETER DISTRIBUTION PARAMETER MEAN = 1.0 # cm
      PARTICLE DIAMETER DISTRIBUTION PARAMETER STDEV = 0.5 # cm
      DIAMETER CUTOFF HIGH = 1.05 # cm
      DIAMETER CUTOFF LOW = 0.95 # cm
      NUMBER REPRESENTED VECTOR = 10 20 30 40 50 60 70
      DIAMETER NUMBER REPRESENTED VECTOR = 0.1 0.4 0.7 1.0 1.3 1.6 1.9
    #Velocity Distribution Parameters
      PARTICLE VELOCITY DISTRIBUTION TYPE            = CONSTANT
      PARTICLE VELOCITY DISTRIBUTION PARAMETER VALUE = 1.0 # cm/s
    #Particle Temperature
     TEMPERATURE = 300.0
 END PARTICLE SPRAY spray1
```

**Figure 4.7-2.. Lagrangian particle spray section of a Fuego input deck showing use of diameter cutoffs**

## 4.7.2. Lagrangian Particle Spray: Angular Spreading Sprays

The angular spreading spray algorithm was modified in version 4.30 to produce an isotropically spreading particle spray (within the angular limits specified). Previously, the particle trajectories were preferentially aligned with the spray axis. For isotropic spread, the cosine of the polar angle (measured with respect to the spray axis) rather than the angle itself is chosen randomly. The polar angle is then determined from the inverse cosine of this value.

$$\theta = cos^{-1}\left[rand\left(\right)\right] \tag{4.118}$$

## 4.7.3. Alumina Absorption Model

Fuego allows for a user to specify the radiation absorption model for alumina in reacting aluminum particle simulations like propellant fires. The alumina absorption model, using a FORTRAN subroutine, can now read from a user input file containing data for the alumina absorption coefficient as a function of particle temperature. The file contains two columns defining this function. The first column is temperature; the second is the absorption coefficient. This function is linearly interpolated to find the absorption coefficient at any temperature of interest. Figure 4.7-3 displays two standard alumina absorption models alongside a user-specified model.

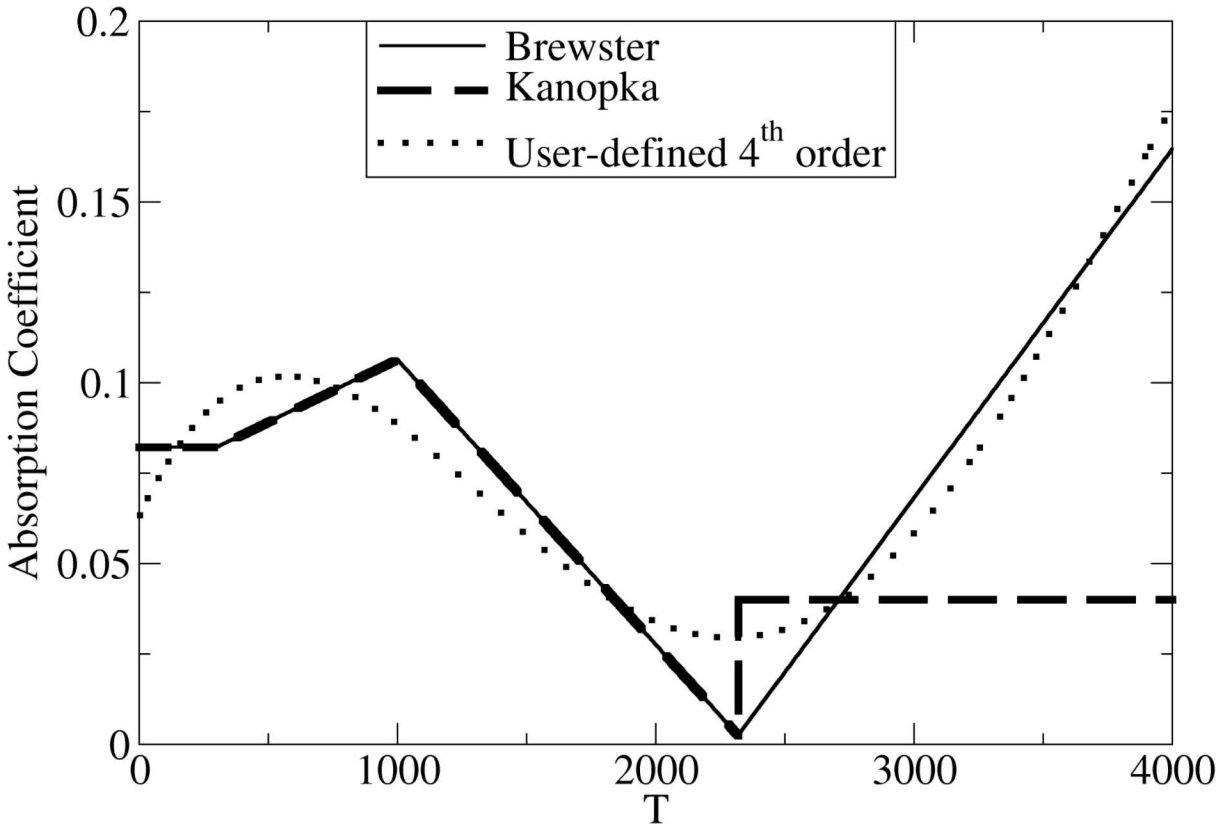


**Figure 4.7-3.. Alumina absorption coefficient for standard models Brewster and Kanopka along with a user-specified model**

### 4.7.4. Emission Multiplier

For propellant fire simulations which use the evaporating Lagrangian particle type, analysts have determined that modifying the particle-radiation coupling can be advantageous to reproducing experimental results. To address this, Fuego has a capability to modify the particle radiation emission with a constant multiplier. When the emission multiplier is not set, a default value of 1 is assumed, and emission = absorption when the particle and fluid temperatures are identical. Particle radiant emission $E_p$ and absorption $A_p$ are:

$$E_p = 4\pi\alpha R_p^2 \sigma_{SB} T_p^4 f_E \tag{4.119}$$

$$A_p = 4\pi\alpha R_p^2 \sigma_{SB} T_f^4 \tag{4.120}$$

where $\alpha$ is the particle absorptivity, $R_p$ is the particle radius, $\sigma_{SB}$ is the Stefan-Boltzmann constant, $T_{p,f}$ are the particle and fluid temperatures respectively, and $f_E$ is the emission multiplier described above.

## 4.7.5. Lagrangian Particle Spray: Number Represented Function

Lagrangian particle sprays have historically been required to use parcelling (grouping of several particles into a single parcel) with either a constant mass represented per parcel or constant number represented per parcel. In propellant fire applications and other reacting particle environments, a more sophisticated functionality between the number of particles represented per parcel and particle size can increase the efficiency of simulations. For this reason, Fuego includes a capability to allow the analyst to specify this function (parcel size vs. particle diameter). This function is specified by a vector for each (number represented per parcel and diameter). For diameters at or below the lowest specified in the vector, the number represented is constant and equal to the value at the lowest diameter specified. For diameters at or above the highest specified in the vector, the number represented is constant and equal to the value at the largest diameter specified. Intermediate values are linearly interpolated. Figure 4.7-4 diagrams the way parcelling works for each of the different parcelling schemes.

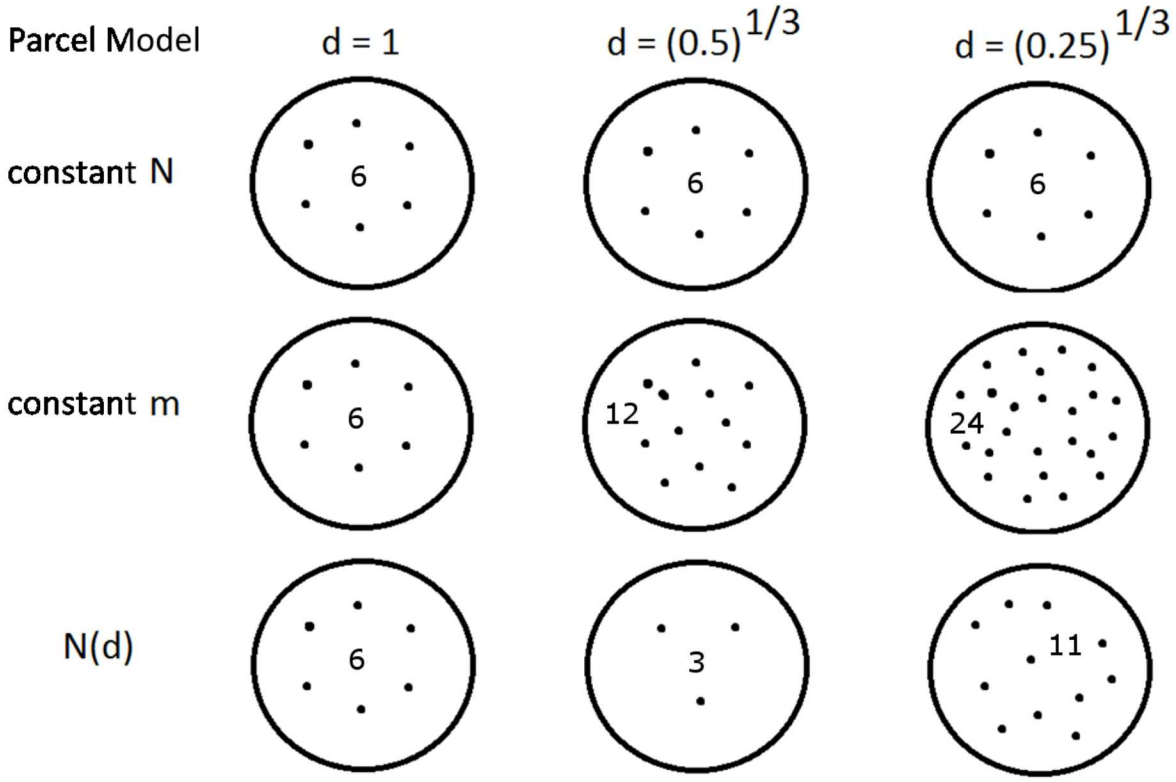


**Figure 4.7-4.. For a Lagrangian particle spray, the number of particles contained within a parcel for three representative particle diameters using constant number, constant mass, and user defined number of particles per parcel. Circles represent parcels with the points inside representing the number of particles contained in the parcel.**

### 4.7.6. Lagrangian Particle Insertion: User Definable Mechanism

Previous to version 4.30, Lagrangian particles could be inserted into the domain through two mechanisms: 1) batch introduction of a group of particles at a specified time with the particle configuration defined by a particle data file or a filled shape (i.e. cone, cylinder) with shape parameters or 2) via a particle spray with either a rectangular or circular nozzle and a specified mass flux rate. In cases where users needed a more novel insertion mechanism, Fuego lacked the capability. Fuego now includes a mechanism for particle insertion from file data in which users can specify not only particle positions, velocities, and diameters on insertion, but also particle temperature, number of particles per parcel, and insertion time. Through this method users have a full range of particle insertion options at their disposal. The dynamical form for particle introduction is contained within the file data, and does not rely on templated forms for static shapes or sprays, though those capabilities are still available. Users can, for instance, introduce particles from a very specific particle size distribution isotropically through the system with a rate of their choosing or create a particle spray with a conical nozzle with velocity vectors normal to the nozzle. The only limitation lies in the ability of the user to specify this mechanism through the particle data file. Figure 4.7-5 shows some examples of particle insertion types available with this capability. Figures 4.7-6 and 4.7-7 display a conical particle spray generated with this mechanism from two different perspectives (conical axis lying in the plane of the figure and normal to the figure) at both early and late times in the simulation. In this case, the particle temperature has been designed to be a function of the position at which the particle left the spray nozzle. Many other forms are possible.

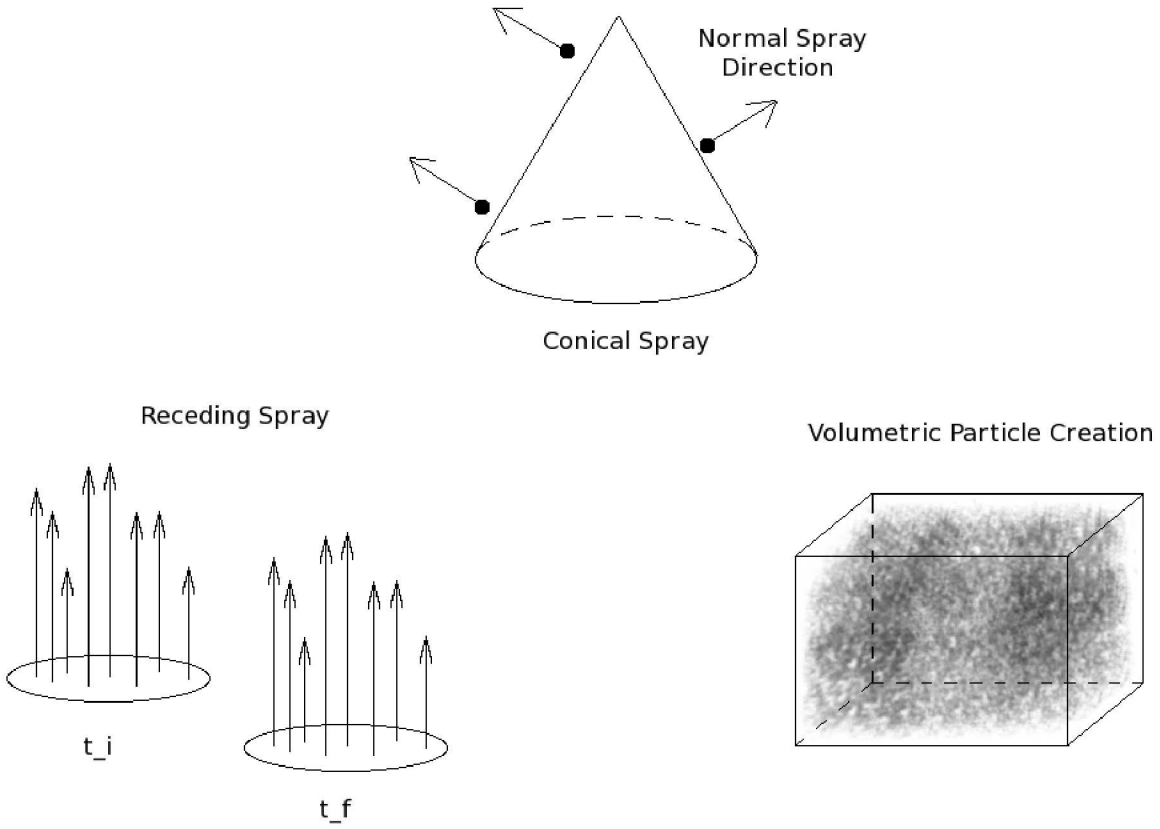


**Figure 4.7-5.. Examples of particle insertion types that can be used with particle insert from file mechanism**

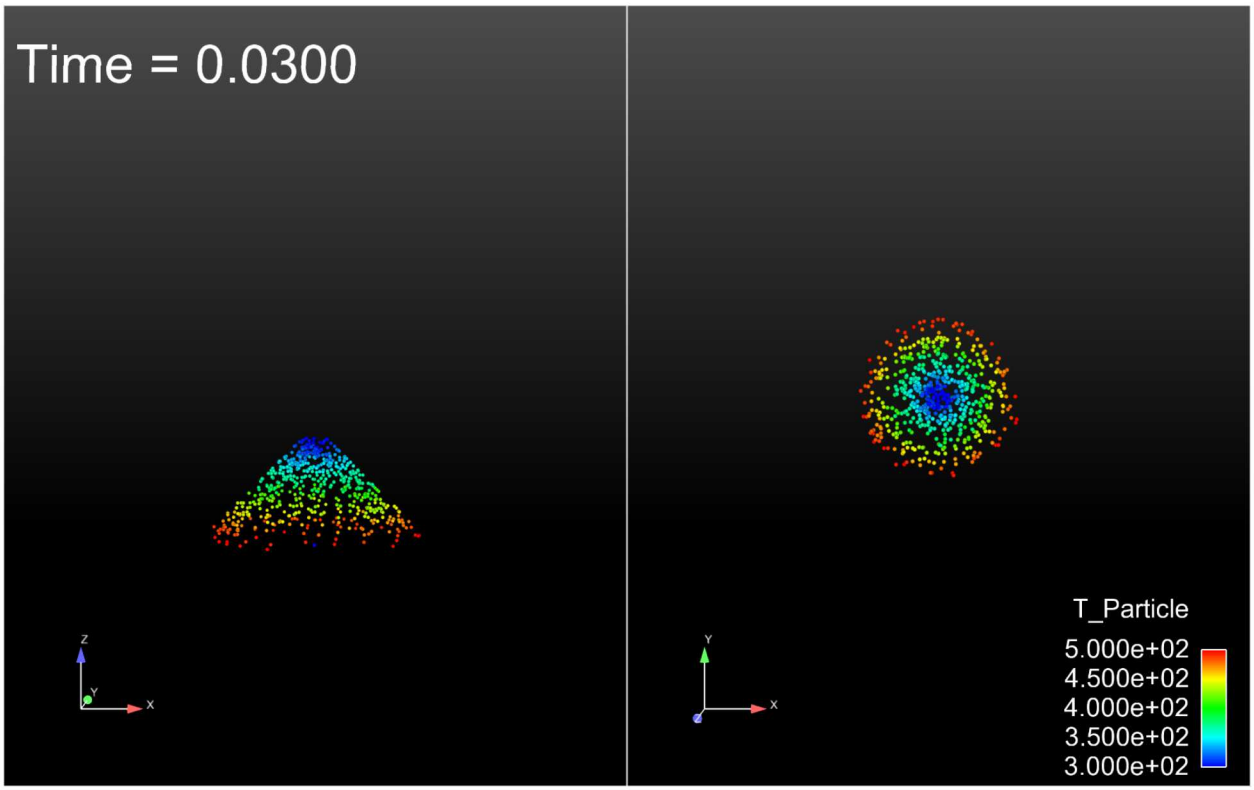


**Figure 4.7-6.. Example of particle spread from a conical shaped particle spray nozzle at early times. This nonstandard spray form was generated through the particle creation from file data mechanism. Here particle temperatures are set to be a function of their position with the hottest particles leaving the nozzle near the circular base of the cone.**

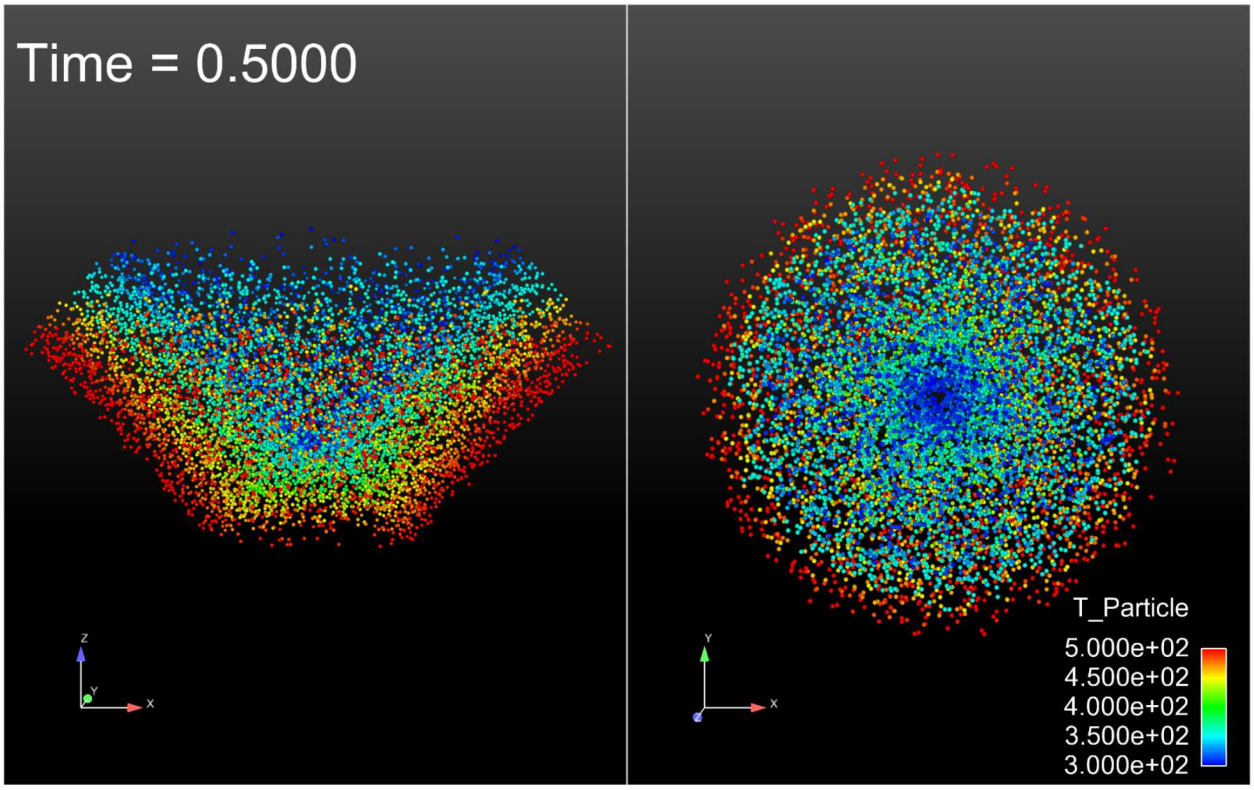


**Figure 4.7-7.. Same simulation as Fig 4.7-6 but at late time.**

# 5. NUMERICS

We surveyed commercial codes that provide turbulent combustion capabilities and discovered that most of those codes are based on finite volume methods. Between commercial evidence and our own experiences, we came to the conclusion that finite volume methods would provide a robust and stable means of solving the fire math models. Our selection of finite volume methods is constrained by the current implementation of software architecture in the SIERRA Frameworks. The mesh must be unstructured with flow variables located at the element vertices. The domain boundary is coincident with element faces. The discrete equations are assembled on an element-by-element procedure using the SIERRA workset approach for cache-use efficiency. The finite volume approach that we implement is based on the control-volume finite-element method.

Control-volume finite-element methods (CVFEM) are a class of numerical methods for solving the Navier-Stokes equations of fluid mechanics. Although the methods are applicable to the most general case of a compressible flow, they are most commonly applied to incompressible flows. This text is a discussion of the control-volume finite-element methods appropriate for numerical solutions to the low-Mach number Navier-Stokes equations with heat and mass transfer—the equations used to describe physical applications such a combustion or chemical vapor deposition.

The CVFEM's are a combination of desirable features from both the finite-element method (FEM) and the finite-volume method (FVM), though the CVFEM is truly a finite-volume method. The CVFEM differed from other FVM's at its inception in that the CVFEM used non-staggered, unstructured meshes like a FEM. Concepts from the finite-element method include: 1) the finite-element data structure and the associated shape functions or interpolation functions, 2) integral equations assembled on an element-by-element basis, an efficient process for cache-based computer architectures, and 3) unstructured meshes with arbitrary connectivity (this is not particular to FEM's, but certainly more common). Reviews for the finite-element method are given by Zienkiewicz and Taylor [125, 126], Tezduyar [64], and Gresho [127]. Concepts from the finite volume method include: 1) physically-based integral formulation constructed from physically-based interpolation functions, 2) conservation properties at the control-volume level, and 3) both a convecting and convected velocity field to avoid pressure-velocity decoupling. Some comprehensive reviews for the finite-volume method are given by Patankar [128], Shyy [129], and Ferziger and Peric [130]. An extensive literature review of control volume finite element methods (CVFEM) is given in Appendix 8.

The standard mesh configuration for vertex-centered CVFEM's has all flow variables collocated at the grid points, also called nodes. The nodes are the vertices of the finite-elements, as shown in Figure 5.0-1. The finite-volumes, also called control volumes, are centered about the nodes. Each element contains a set of sub-faces that define control-volume surfaces. The sub-faces consist of the segments or surfaces that bisect the element faces.

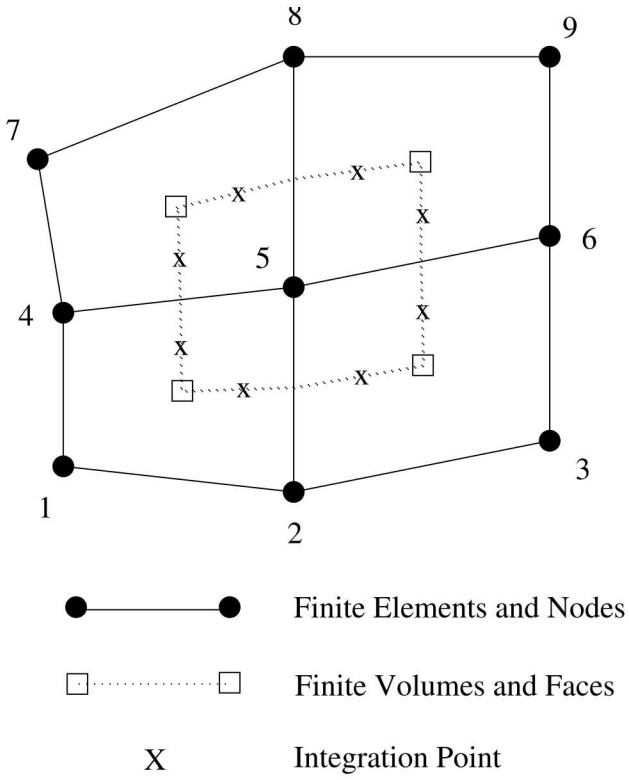


**Figure 5.0-1.. Control Volume is Centered about Finite-Element Node**

# 5.1. FLOW SOLVER

The core flow solver is based on a segregated, projection method approach. The projection method is used to compute the pressure field which is consistent with a velocity field that satisfies continuity. A pressure-smoothing approach similar to the Rhie/Chow scheme [131] is used to prevent pressure decoupling on the collocated mesh. An upwind method is used to interpolate convected values to control volume faces. Detailed descriptions of these methods are discussed in the following sections.

Another prevalent CVFEM method in the literature is the FIELDS method [132, 133]. The continuity and momentum equations are fully coupled in this approach. We experimented with this approach and found that the three-dimensional discrete equations were difficult to solve and open boundary conditions difficult to implement.

## 5.1.1. Projection Method

The role of pressure smoothing, or explicit stabilization, was first developed in the context of collocated finite volume schemes by [131]. Although this original paper did not explore the formal error introduced by this explicit stabilization, [134] later displayed the sensitivity of steady results on relaxation parameters and provided a methodology to circumvent this issue. In general, such early papers (cf. [135])

as well as other more recent papers, (cf. [136]) introduced the role of stabilization almost by happenstance as it entered only through the specific choice of the convecting velocity formula, i.e., the integration point velocity that forms the mass flow rate.

Studies of [137] and [138], each in the context of a finite element algorithm, have commented on the role of stabilization that is provided by the approximation of the derived pressure correction system, namely that $\mathbf{L} \neq \mathbf{DG}$, where $\mathbf{L}$ is the given discrete Laplacian operator and $\mathbf{D}$ and $\mathbf{G}$ are the chosen discrete divergence and gradient operators, respectively. Numerical algorithms for which the Laplacian operator does not equal the discrete divergence of gradient operator have been termed "approximate projection" algorithms (cf. [139] and [140]) in the context of solenoidal flow; in general for non-solenoidal flow the formalism of the projection derivation results in an affine projection.

Recent work by Sandia National Laboratories has cast the general approximate projection algorithm within a family of smoothing and time scaling choices. The analysis of choice that has been followed is to cast the algorithm in terms of an approximate factorization (cf. [141]), and note the added stabilization (herein also known as *pressure smoothing terms*), and splitting errors. This analysis has been extremely useful in understanding the formal accuracy, and even consistency, of a given numerical scheme.

The analysis of a given computational fluids algorithm in the context of an approximate factorization begins with the discrete momentum and continuity equations written in matrix form. The matrix $\mathbf{A}$ contains discrete, linearized contributions to the momentum equations from the time derivative, convection, and diffusion terms,

$$\begin{bmatrix} \mathbf{A} & \mathbf{G} \\ \mathbf{D} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u}^{n+1} \\ p^{n+1/2} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{b} \end{bmatrix}. \tag{5.1}$$

The discrete nodal gradient and nodal divergence are $\mathbf{G}$ and $\mathbf{D}$ respectively (note that the operator $\mathbf{D}$ may include aspects of the algorithm due to a variable density field). The function $\mathbf{f}$ contains the additional terms for the momentum equations, e.g., body force terms, lagged stress tensor terms, etc., while the function $\mathbf{b}$ contains the appropriate terms for a non-solenoidal velocity field, i.e., $-\int \frac{\partial \rho}{\partial t} dV$. The pressure is appropriately interpreted as the pressure at the $n + \frac{1}{2}$ step, (cf. [142]). The form of the matrix operators can be found in the body of literature for control-volume finite element methods (cf. [143]). Note that Equation 5.1 is not really complete as the boundary condition values are omitted, however, they are not essential in describing the bulk of the splitting and stabilization analysis as noted by [144]. The boundary conditions would simply enter through an additional vector on the right-hand side and modified entries in the matrix operators.

The approximate factorization of Equation 5.1 takes the general form of

$$\begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{D} & \mathbf{B_1} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{B_2 G} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{A B_2 G} \\ \mathbf{D} & \mathbf{B_1} + \mathbf{D B_2 G} \end{bmatrix}. \tag{5.2}$$

The factor $\mathbf{B_2}$ determines the projection time scale. The factor $\mathbf{B_1}$ defines the linear system for pressure. Ideally, $\mathbf{B_1}$ could be selected to cancel splitting errors in the continuity equation. Practically, the form of $\mathbf{B_1}$ is governed by implementation and linear solver efficiency.

A completely generalized set of incremental pressure projection methods with potential stabilization can be written by formally defining the operators $\mathbf{B_1}$ and $\mathbf{B_2}$ above, here shown as part of the sequence of equations solved,

$$\begin{aligned}
\mathbf{A}\Delta\hat{\mathbf{u}} &= \mathbf{f} - \mathbf{G}p^{n-\frac{1}{2}} - \mathbf{A}\mathbf{u}^n, && (5.3)\\
-\mathbf{L_1}\Delta p^{n+\frac{1}{2}} &= -\mathbf{D}\left(\hat{\mathbf{u}} + \tilde{\tau}_2\mathbf{G}p^{n-\frac{1}{2}}\right) + \mathbf{L_2}p^{n-\frac{1}{2}} + b, && (5.4)\\
\mathbf{u}^{n+1} &= \hat{\mathbf{u}} - \tilde{\tau}_3\mathbf{G}\Delta p^{n+\frac{1}{2}}. && (5.5)
\end{aligned}$$

Laplacian operators acting on a general scalar $\phi$, which define the approximate nature of the projection method, are given by,

$$\begin{aligned}
\mathbf{L}_1\phi &= \tau_1\nabla\phi\cdot d\mathbf{A}, && (5.6)\\
\mathbf{L_2}\phi &= \tau_2\nabla\phi\cdot d\mathbf{A}. && (5.7)\\
& && (5.8)
\end{aligned}$$

For an approximate projection method,

$$\mathbf{L_2} \neq \mathbf{D}\tilde{\tau}_2\mathbf{G}, \tag{5.9}$$

while for an exact projection,

$$\mathbf{L_2} = \mathbf{D}\tilde{\tau}_2\mathbf{G}. \tag{5.10}$$

Exact projections can be easily constructed on unstructured collocated meshes (cf. [145]), although classically this results in a wide Laplacian stencil that admits pressure oscillations yet does not add discrete errors in the continuity solve. We assume that $\tau_i$ factors defined above are represented by a diagonal matrix that corresponds to a particular time scale of choice. The relationship between $\tilde{\tau}_i$ and $\tau_i$ is normalization by a density and volume,

$$\tilde{\tau}_i = \frac{\tau_i}{\rho V}. \tag{5.11}$$

The choice of these scaling factors defines the scheme in terms of both stabilization and projection scaling. For example, the ideal form for $\tilde{\tau}_3$ is the inverse of $\mathbf{A}$. The exact choice of $\tilde{\tau}_3$ in a practical sense affects the stability of the scheme. The stabilization terms are represented by operators including both $\tilde{\tau}_1$ and $\tilde{\tau}_2$ that are required to prevent velocity and pressure decoupling in schemes for which $\mathbf{L} \neq \mathbf{DG}$.

Rearrangement of Equation 5.5, in terms of $\hat{\mathbf{u}}$, and substitution of this modified equation into Equation 5.3 and Equation 5.4 provides the full set of splitting and stabilization errors:

$$\begin{aligned}
\begin{bmatrix} \mathbf{A} & \mathbf{G} \\ \mathbf{D} & \mathbf{0} \end{bmatrix}\begin{bmatrix} \mathbf{u}^{n+1} \\ p^{n+1/2} \end{bmatrix} &= \begin{bmatrix} \mathbf{f} \\ \mathbf{b} \end{bmatrix} \\
&+ \begin{bmatrix} (\mathbf{I} - \mathbf{A}\tilde{\tau}_3)\,\mathbf{G}\Delta p^{n+\frac{1}{2}} \\ (\mathbf{L_1} - \mathbf{D}\tilde{\tau}_3\mathbf{G})\Delta p^{n+\frac{1}{2}} + (\mathbf{L_2} - \mathbf{D}\tilde{\tau}_2\mathbf{G})p^{n-\frac{1}{2}} \end{bmatrix}. && (5.12)
\end{aligned}$$

The error appearing in the momentum equation is due to splitting and generally can be repaired by non-linear iteration, although ideally single iteration methods are desired (as shown).

Again it is emphasized that for approximate projection methods, $\mathbf{L_2} \neq \mathbf{D}\tilde{\tau}_2\mathbf{G}$, whereas for exact projection methods, which are usually based on staggering velocity and pressure, $\mathbf{L_2} = \mathbf{D}\tilde{\tau}_2\mathbf{G}$ and there is no stabilization error (as there is no need to provide stabilization). Frequently, the stabilization terms within Equation 5.4 are included in a modified provisional velocity (cf. [146]), i.e., $\tilde{\mathbf{u}} = \hat{\mathbf{u}} + \tilde{\tau}_2\mathbf{G}p^{n-\frac{1}{2}}$, that can often hide the true role of stabilization.

A similar analysis for $pressure\ free$ projection methods (cf. [147]) can be carried out, in which case the equations solved are given by,

$$\begin{aligned} \mathbf{A}\hat{\mathbf{u}} &= \mathbf{f} - \mathbf{A}\mathbf{u}^n, & (5.13)\\ -\mathbf{L_1}\Delta\phi^{n+1} &= -\mathbf{D}\hat{\mathbf{u}} + \mathbf{L_1}\phi^n + b, & (5.14)\\ \mathbf{u}^{n+1} &= \hat{\mathbf{u}} - \tilde{\tau}_1\mathbf{G}\phi^{n+1}, & (5.15) \end{aligned}$$

with errors,

$$\begin{bmatrix} \mathbf{A} & \mathbf{G} \\ \mathbf{D} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u}^{n+1} \\ p^{n+1/2} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{b} \end{bmatrix} + \begin{bmatrix} -\mathbf{A}\tilde{\tau}_3\mathbf{G}\phi^{n+1} + \mathbf{G}p^{n+\frac{1}{2}} \\ (\mathbf{L_1} - \mathbf{D}\tilde{\tau}_1\mathbf{G})\,\phi^{n+1} \end{bmatrix}. \quad (5.16)$$

The error term in the continuity equation is retained to emphasize that this algorithm can be considered in the context of an approximate projection method. Assuming that the Laplacian and gradient operators commute, it is necessary to compute $p^{n+1/2} = \mathbf{A}\tau_3\phi^{n+1}$ to obtain the second-order pressure field, while the relationship $p^{n+\frac{1}{2}} = \phi^{n+1}$ will result in a first-order pressure field with splitting error $(\mathbf{I} - \mathbf{A}\tau_3)\mathbf{G}p^{n+\frac{1}{2}}$ ( [148]).

Although the above set of algorithms have been written in terms of a two step scheme, i.e., predict $\hat{\mathbf{u}}$ and correct $\hat{\mathbf{u}}$ by the appropriately scaled scalar gradient, non-linear iterations can also be taken. In this case, the $\phi^{n+1}$ and $\mathbf{u}^{n+1}$ state are replaced with the $k+1$ state, whereas the $n+\frac{1}{2}$ pressure state is replaced by the $k+\frac{1}{2}$ state. For the residual form, the $n^{th}$ state is replaced with the current iterate, $k^{th}$ state. At convergence within the time step, $\phi^{n+1} = \phi^{k+1}$, $\mathbf{u}^{n+1} = \mathbf{u}^{k+1}$, and $p^{n+\frac{1}{2}} = p^{k+\frac{1}{2}}$.

#### 5.1.1.1. CVFEM operators

SIERRA/Fuego uses the finite volume technique known as the control volume finite element method of [149]. Control volumes (the mesh dual) are constructed about the nodes, as shown in Fig. 5.0-1. Each element contains a set of subfaces that define control-volume surfaces. The subfaces consist of line segments (2-D) or surfaces (3-D). The 2-D segments are connected between the element centroid and the edge centroids. The 3-D surfaces are connected between the element centroid, the element face centroids, and the edge centroids. Integration points also exist within the subcontrol volume centroids. Such integration points are used for volume integrals such as source terms, the mass matrix, and, if chosen, gradients.

Defining $\phi_K$ to be the value of $\phi$ at node $K$, then the variation of $\phi$ within an element that contains the point location x is given by

$$\phi(\mathbf{x}) = \sum_{K\in\mathcal{N}} N_K(\mathbf{x})\phi_K, \quad (5.17)$$

where $N_K(\mathbf{x})$ is the shape function associated with node $K$ at position $\mathbf{x}$, and $\mathcal{N}$ is the set of all nodes that defines the element. For the CVFEM, either trilinear (3-D) or bilinear (2-D) shape functions are

used. Currently, Fuego supports heterogeneous element topologies consisting of hex, tet, pyramid, and wedges.

The discrete nodal gradient operator for direction $i$ can be written as a surface integral on control volume $L$,

$$\mathbf{G}\phi = (G\phi)_{Li} = \int_{\Gamma_L} \phi(\mathbf{x}) dn_i \approx \sum_{\alpha \in \mathcal{B}_L} \left( \sum_{K \in \mathcal{N}} N_K(\mathbf{x}_\alpha)\phi_K \right) n_i(\mathbf{x}_\alpha)\Delta A_\alpha, \tag{5.18}$$

where $\mathcal{B}_L$ is the set of surface integration points for control volume $L$. Similarly, the discrete divergence operator at node $L$ acting on vector $u_i$ is

$$\mathbf{Du} = (Du_i)_L = \int_{\Gamma_L} \rho(\mathbf{x}) u_i(\mathbf{x}) dn_i \approx \sum_{\alpha \in \mathcal{B}_L} \rho(\mathbf{x}_\alpha) \left( \sum_{K \in \mathcal{N}} N_K(\mathbf{x}_\alpha) u_{Ki} \right) n_i(\mathbf{x}_\alpha)\Delta A_\alpha, \tag{5.19}$$

and the Laplacian operator that includes spatially varying timescale, $\tau$, is

$$\mathbf{L}_\tau\phi = (L_\tau\phi)_L = \int_{\Gamma_L} \tau(\mathbf{x}) \frac{\partial \phi}{\partial x_j} dn_j \approx \sum_{\alpha \in \mathcal{B}_L} \tau(\mathbf{x}_\alpha) \left( \sum_{K \in \mathcal{N}} \frac{\partial N_K(\mathbf{x}_\alpha)}{\partial x_j} \phi_K \right) n_j(\mathbf{x}_\alpha)\Delta A_\alpha. \tag{5.20}$$

Note that an alternative to the gradient operator given in Equation 5.18, which is provided via the CVFEM is

$$\mathbf{G}\phi = (G\phi)_{Li} = \int_{\Gamma_L} \frac{\partial \phi}{\partial x_i} dV \approx \sum_{\alpha' \in \mathcal{B}_L} \left( \sum_{K \in \mathcal{N}} \frac{\partial N_K(\mathbf{x}_{\alpha'})}{\partial x_i} \phi_K \right) dV_{\alpha'}, \tag{5.21}$$

where $\mathcal{B}_L$ is now the set of all subcontrol volume integration points for control volume $L$ (for clarity, $\alpha'$ denotes the subcontrol volume integration point location).

The general term $\mathbf{D}\tilde{\tau}\mathbf{G}\phi$ deserves a special note in the case of variable density flows. Specifically, the interpolation is currently provided by the following equation:

$$\mathbf{D}\tilde{\tau}_i\mathbf{G}\phi = \sum_{\alpha \in \mathcal{B}_L} \rho(\mathbf{x}_\alpha) \frac{\tilde{\tau}_i(\mathbf{x}_\alpha)}{\rho(\mathbf{x}_\alpha)} \left( \sum_{K \in \mathcal{N}} N_K(\mathbf{x}_\alpha) \frac{G_{Ki}}{V_K} \right) n_i(\mathbf{x}_\alpha)\Delta A_\alpha, \tag{5.22}$$

$$= \sum_{\alpha \in \mathcal{B}_L} \tilde{\tau}_i(\mathbf{x}_\alpha) \left( \sum_{K \in \mathcal{N}} N_K(\mathbf{x}_\alpha) \frac{G_{Ki}}{V_K} \right) n_i(\mathbf{x}_\alpha)\Delta A_\alpha. \tag{5.23}$$

## 5.2. SMOOTHING ALGORITHMS DEFINED

Now that the smoothing and splitting errors have been formally defined, it is useful to consider three projection algorithms that have been implemented and verified within SIERRA/Fuego in the context of the classic two equation $k$-$\epsilon$ model, with steady method of manufactured solutions (MMS) (cf. [150]).

#### 5.2.0.1. Fourth-order smoothing with characteristic or time step scaling

In this algorithm, the projection time scales are defined by either

$$\tau = \tau_1 = \tau_2 = \tau_3 = \tau_{char}, \tag{5.24}$$

or

$$\tau = \tau_1 = \tau_2 = \tau_3 = \mathbf{I}\Delta t. \tag{5.25}$$

Here, characteristic scaling, $\tau_{char}$, is a diagonal matrix that represents a time scale based on convection and diffusion contributions, while for time step scaling, the time scale is based on the local time step. The characteristic scaling very closely follows the standard finite element method stabilization parameter.

The smoothing and splitting errors are now given by

$$\begin{bmatrix} \mathbf{A} & \mathbf{G} \\ \mathbf{D} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u}^{n+1} \\ p^{n+1/2} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{b} \end{bmatrix} + \begin{bmatrix} (\mathbf{I} - \mathbf{A}\tilde{\tau})\mathbf{G}(p^{n+\frac{1}{2}} - p^{n-\frac{1}{2}}) \\ (\mathbf{L}_\tau - \mathbf{D}\tilde{\tau}\mathbf{G})p^{n+\frac{1}{2}} \end{bmatrix}. \tag{5.26}$$

Of particular interest to this research is the role of the stabilization term, $(\mathbf{L}_\tau - \mathbf{D}\tilde{\tau}\mathbf{G})p^{n+\frac{1}{2}}$, on formal time accuracy when $\tau = \mathbf{I}\Delta t$ (a scheme that has been shown to display more appealing stability characteristics). Clearly, a scheme that uses explicit pressure stabilization with time step scaling is first-order accurate. Expanding this stabilization term shows the fourth-order pressure derivative scaled by a length scale cubed. Therefore, by refining the time step *and* mesh, one might be able to show a second-order accuracy for sufficiently resolved meshes.

In practice, the stabilization terms are carried within the mass flow rate that forms part of the right-hand side of the Pressure Poisson Equation solve and the convection term for the transport of any scalar field. The mass flow rate is defined as

$$\dot{m}^k = \left( \overline{\rho}\overline{\hat{\mathbf{u}}} + \frac{\overline{\tau \mathbf{G} p^{n-\frac{1}{2}}}}{V} - \overline{\tau}\nabla^h p^{n+\frac{1}{2}} \right) d\mathbf{A}, \tag{5.27}$$

where the introduction of the over bar is noted to represent interpolation of a nodal field to an integration point. Note that in the bulk of the collocated unstructured finite volume literature, the form of the mass flow rate defines the stabilization (the difference between the nodal gradient operator $\mathbf{G}$ and the interior element operator $\nabla^h$). Above we note the independent interpolation of the density and velocity rather than $\overline{\rho\hat{\mathbf{u}}}$, as is done in Stanford's ASC Alliance code CDP. It does seem that the full interpolation of $\rho\hat{\mathbf{u}}$ may be more consistent, although the effect of this algorithmic detail has not been explored.

#### 5.2.0.2. Stabilized smoothing

The stabilized projection algorithm is based on the work of [138], that was derived from the monolithic scheme of [137]. In this algorithm, the projection time scales are defined as

$$\tau_1 = \Delta t\mathbf{I} + \tau_{char}. \tag{5.28}$$

$$\tau_2 = \tau_{char}. \tag{5.29}$$

$$\tau_3 = \tau_{char}. \tag{5.30}$$

With the above definitions, the smoothing and splitting errors are now defined as

$$\begin{bmatrix} \mathbf{A} & \mathbf{G} \\ \mathbf{D} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u}^{n+1} \\ p^{n+\frac{1}{2}} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{b} \end{bmatrix} + \begin{bmatrix} (\mathbf{I} - \mathbf{A}\tilde{\tau}_{char})\mathbf{G}(p^{n+\frac{1}{2}} - p^{n-\frac{1}{2}}) \\ (\mathbf{L}_{\tau_{\mathbf{char}}} - \mathbf{D}\tilde{\tau}_{char}\mathbf{G})p^{n+\frac{1}{2}} + \Delta t\mathbf{L}(p^{n+\frac{1}{2}} - p^{n-\frac{1}{2}}) \end{bmatrix}. \tag{5.31}$$

The mass flow rate now includes an additional stabilization factor and is now defined as

$$\dot{m}^k = \left( \overline{\rho}\overline{\hat{\mathbf{u}}} + \overline{\frac{\tau \mathbf{G} p^{n-\frac{1}{2}}}{V}} - \overline{\tau}\nabla^h p^{n+\frac{1}{2}} - \Delta t \mathbf{L} \Delta p^{n+\frac{1}{2}} \right) d\mathbf{A}. \tag{5.32}$$

Note that at full convergence, the stabilized scheme reduces to the fourth-order characteristic scaling algorithm.

#### 5.2.0.3. Second-order smoothing with characteristic or time step scaling

In fact, the scaled nodal gradient need not be included in the mass flow rate equation, e.g.,

$$\dot{m}^k = \left( \overline{\rho}\overline{\hat{\mathbf{u}}} - \overline{\tau}\nabla^h p^{n+\frac{1}{2}} \right) d\mathbf{A}. \tag{5.33}$$

This is equivalent to neglecting the $\tilde{\tau}_2 \mathbf{G} p^{n-\frac{1}{2}}$ term in Equation 5.4, or by defining $\tilde{\mathbf{u}} = \hat{\mathbf{u}}$.

The smoothing for this algorithm is provided by the local Laplacian operator. The smoothing and splitting errors for this method are now given by

$$\begin{bmatrix} \mathbf{A} & \mathbf{G} \\ \mathbf{D} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u}^{n+1} \\ p^{n+1/2} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{b} \end{bmatrix} + \begin{bmatrix} (\mathbf{I} - \mathbf{A}\tilde{\tau})\mathbf{G}(p^{n+\frac{1}{2}} - p^{n-\frac{1}{2}}) \\ (\mathbf{L}_\tau - \mathbf{D}\tilde{\tau}\mathbf{G})\Delta p^{n+\frac{1}{2}} + \mathbf{L}_\tau p^{n-\frac{1}{2}} \end{bmatrix}. \tag{5.34}$$

#### 5.2.0.4. Zeroth-order smoothing with time step or characteristic scaling

Certainly, the pressure smoothing can be removed, i.e., $\tau_2 = 0$, that leads to the following set of errors,

$$\begin{bmatrix} \mathbf{A} & \mathbf{G} \\ \mathbf{D} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u}^{n+1} \\ p^{n+\frac{1}{2}} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{b} \end{bmatrix} + \begin{bmatrix} (\mathbf{I} - \mathbf{A}\tilde{\tau})\mathbf{G}(p^{n+\frac{1}{2}} - p^{n-\frac{1}{2}}) \\ (\mathbf{L}_\tau - \mathbf{D}\tilde{\tau}\mathbf{G})(p^{n+\frac{1}{2}} - p^{n-\frac{1}{2}}) \end{bmatrix}. \tag{5.35}$$

where $\tau$ is either the characteristic scale, $\tau_{char}$, or the simulation time step, $\mathbf{I}\Delta t$ (with $\tau_1 = \tau_3$). Although the converged error is zero, this lack of smoothing can lead to a decoupled pressure field in certain flows.

Here, the mass flow rate reduces to a simple interpolation of nodal velocities within the element

$$\dot{m}^k = \left( \overline{\rho}\overline{\hat{\mathbf{u}}} - \overline{\tau}\nabla^h \Delta p^{n+\frac{1}{2}} \right) d\mathbf{A}. \tag{5.36}$$

The unsmoothed algorithm is very similar to the staggered formulation of SIMPLE, (cf. [128]), with $\tau = A_p^{-1}$ (the inverse of the diagonal matrix from operator A). However, by design, the staggered mesh arrangement holds the property that $(\mathbf{L}_\tau - \mathbf{D}\tau\mathbf{G}) = 0$. In this method, no stabilization is added as none is required.

#### 5.2.0.5. Time integration scheme

At present, three time integration schemes are supported in the code base that include: 1) first-order backward Euler, 2) second-order BDF2 and, 3), the Crank-Nicholson method described in [151].

The general two- and three-state scheme time derivatives are simply written as:

$$\int \frac{\partial \rho \phi}{\partial t} dV = \int \frac{(\gamma_1 \rho^{n+1} \phi^{n+1} + \gamma_2 \rho^n \phi^n + \gamma_3 \rho^{n-1} \phi^{n-1})}{\Delta t} dV \tag{5.37}$$

where $\gamma_i$ represent the appropriate factors for either Backward Euler or a three-point BDF2 scheme. The above time derivative is either nodally lumped or evaluated at the subcontrol volume quadrature points. For a two-state backward Euler scheme, $\gamma_1$ is unity while $\gamma_2$ is negative unity. For a given variable time step, the BDF2 factors are,

$$\tau = \Delta t^N / \Delta t^{N-1}, \tag{5.38}$$

$$\gamma_1 = \frac{(1+2\tau)}{(1+\tau)}, \tag{5.39}$$

$$\gamma_2 = (1+\tau), \tag{5.40}$$

$$\gamma_3 = \frac{\tau^2}{(1+\tau)}. \tag{5.41}$$

For a fixed time step, the $\gamma$-factors reduce to the canocical $(\frac{3}{2}, -2, \frac{1}{2})$ set.

In the Crank-Nicholson implementation, the generalized method is written as

$$\frac{\partial \phi}{\partial t}^{n+1} = \eta \frac{\phi^{n+1} - \phi^n}{\Delta t} + (1-\eta) \frac{\partial \phi}{\partial t}^n, \tag{5.42}$$

where $\eta$ is a blending coefficient between 1 and 2. Values of $\eta$ of unity result in first-order backward Euler, while values of 2 result in second order Crank-Nicholson, i.e.,

$$\frac{\partial \phi}{\partial t}^{n+1} = 2 \frac{(\phi^{n+1} - \phi^n)}{\Delta t} - \frac{\partial \phi}{\partial t}^n. \tag{5.43}$$

A linearization is given by

$$\frac{\partial \phi}{\partial t}^{n+1} = 2 \frac{(\phi^{k} - \phi^n)}{\Delta t} - \frac{\partial \phi}{\partial t}^n, \tag{5.44}$$

where the old time derivative is computed based on the old solution of the partial differential equation of interest. The above algorithm is especially useful in that it avoids the need to evaluate complex right-hand side source terms at the n+1 and n state, e.g., simulations that include the need to compute turbulence production, reaction rate terms, etc.

#### 5.2.0.6. Variable density

In the case of variable density, the same set of options exists in the code. Specifically, the time derivative is written as (for backward Euler or BDF2),

$$\int \frac{\partial \rho}{\partial t} dV = \int \frac{(\gamma_1 \rho^{n+1} + \gamma_2 \rho^n + \gamma_3 \rho^{n-1})}{\Delta t} dV \tag{5.45}$$

For the Crank/Nicolson scheme, the full time term is

$$\frac{\partial \rho \phi}{\partial t}^{n+1} = \eta \frac{\rho^{n+1}\phi^{n+1} - \rho^n \phi^n}{\Delta t} + (1-\eta)\frac{\partial \rho \phi}{\partial t}^{n}, \tag{5.46}$$

where it is noted that the full time derivative at $n^{th}$ state is saved. The linearization is given by

$$\frac{\partial \rho \phi}{\partial t}^{n+1} = \eta \frac{\rho^{k}\phi^{k} - \rho^n \phi^n}{\Delta t} + (1-\eta)\frac{\partial \rho \phi}{\partial t}^{n}. \tag{5.47}$$

In practice, the usage of the above formula for a second-order density derivative has proven unstable. As such, $\eta$ is set to unity.

## 5.3. DISCRETE SYSTEM OF EQUATIONS

The full approximate pressure projection scheme for non-uniform density is now written as

$$\begin{aligned}
\eta M_L^k \Delta \hat{u}_i + C_L(\dot{m}^k)\Delta \hat{u}_i - T_{Lj}\Delta \hat{u}_i &= -r_i, && (5.48)\\
-L_{\tau_1 L}\Delta p^{n+\frac{1}{2}} &= -D_L(\hat{u}_i) - L_{\tau_1} p^k + (L_2 - D\tilde{\tau}_2 G)_L p^k + b, && (5.49)\\
u_{Li}^{n+1} &= \hat{u}_{Li} - \tilde{\tau} G_{Li}\Delta p^{n+\frac{1}{2}}. && (5.50)
\end{aligned}$$

The variable $-r_i$ is the residual that includes body source terms, pressure gradient, the non-symmetric part of the viscous stress term, $T_{Li}^{ns}u_j^k$, parts of the time term and the left-hand side set of operators acting on the $u_i^k$ state,

$$-ri = -\eta M_L^k u_i^k - C_L(\dot{m}^k)u_i^k + T_{Lj}\Delta u_i^k + T_{Li}^{ns}u_j^k + S_{Li} - (1-\eta)M_L(\rho \dot{u}_i^n) - G_{Li}p^{n-\frac{1}{2}}. \tag{5.51}$$

The mass matrix, $M_L^k \Delta \hat{u}_i$, is defined by

$$M_L^k \Delta \hat{u}_i = \sum_{\alpha' \in \mathcal{B}_L} \left( \sum_{K \in \mathcal{N}} N_K(\mathbf{x}_{\alpha'}) \frac{\rho_K^k}{\Delta t} \right) \left( \sum_{K \in \mathcal{N}} N_K(\mathbf{x}_{\alpha'}) \Delta \hat{u}_{Ki} \right) dV_{\alpha'}. \tag{5.52}$$

The shape function above, $N_K(\mathbf{x}_{\alpha'})$, is frequently evaluated at $\mathbf{x}_{\mathcal{N}}$, the coordinates of the vertex associated with the transport equation, i.e., the case where a lumped mass matrix is used.

For simplicity, the central difference operator is provided in $C_{Li}\Delta\hat{u}_i$ as

$$C_L\Delta\hat{u}_i = \sum_{\alpha\in\mathcal{B}_L} m_\alpha^k \left( \sum_{K\in\mathcal{N}} N_K(\mathbf{x}_\alpha)\Delta\hat{u}_{Ki} \right). \tag{5.53}$$

In the preceding equation, the mass flow rate has been linearized within the iteration step and may or may not include the explicit stabilization terms. Moreover, the shape function operator, $N_K(\mathbf{x}_\alpha)$, may be evaluated at the edge midpoints to retain the skew symmetric aspect of the operator $C_L$. By default, this term is evaluated at the subcontrol surface integration points, which retains the CVFEM canonical 27-point stencil.

The symmetric part of the stress tensor is given by

$$T_{Lj}\hat{u}_i = \sum_{\alpha\in\mathcal{B}_L} \left( \sum_{K\in\mathcal{N}} N_K(\mathbf{x}_\alpha)\mu_K \right) \left( \sum_{K\in\mathcal{N}} \frac{dN_K(\mathbf{x}_\alpha)}{dx_j}\hat{u}_{Ki} \right) n_j(\mathbf{x}_\alpha)\Delta A_\alpha, \tag{5.54}$$

while the non-symmetric stress tensor is given by

$$\begin{aligned} T_{Li}^{ns}u_j^k &= \sum_{\alpha\in\mathcal{B}_L} \left( \sum_{K\in\mathcal{N}} N_K(\mathbf{x}_\alpha)\mu_K \right) \left( \sum_{K\in\mathcal{N}} \frac{dN_K(\mathbf{x}_\alpha)}{dx_i}u_{Kj}^k \right) n_j(\mathbf{x}_\alpha)\Delta A_\alpha \\ &- \frac{2}{3}\sum_{\alpha\in\mathcal{B}_L} \left( \sum_{K\in\mathcal{N}} N_K(\mathbf{x}_\alpha)\mu_K \right) \left( \sum_{K\in\mathcal{N}} \frac{dN_K(\mathbf{x}_\alpha)}{dx_p}u_{Kp}^k \right) \delta_{ip}n_p(\mathbf{x}_\alpha)\Delta A_\alpha. \end{aligned} \tag{5.55}$$

Note that the nodal pressure gradient at node $L$ for control volume $L$ for direction $i$ is defined by Equation 5.18. The operator, $S_{Li}$, contains the gravitational term as well as the [potentially] subtracted out hydrostatic term,

$$S_{Li} = \sum_{\alpha'\in\mathcal{B}_L} \left( \sum_{K\in\mathcal{N}} N_K(\mathbf{x}_{\alpha'})(\rho_K^k - \rho^{ref}) \right) g_i dV_{\alpha'}. \tag{5.56}$$

The old time term contribution, $M_L(\rho\dot{u_i}^n)$, is defined by

$$M_L(\rho\dot{u_i}^n) = \sum_{\alpha'\in\mathcal{B}_L} \left( \sum_{K\in\mathcal{N}} N_K(\mathbf{x}_{\alpha'})\rho_K\dot{u}_{Ki}{}^n \right) dV_{\alpha'}. \tag{5.57}$$

Again, $\alpha' \in \mathcal{B}_L$ is the set of all subcontrol volume integration points for control volume $L$, $\alpha' \in \mathcal{B}_L$ is the set of all subcontrol surface integration points for control volume $L$, and $K \in \mathcal{N}$ is the set of all nodes within the element.

#### 5.3.0.1. Predictor

In general, there are a number of predictors that are supported. The easiest predictor is a simple predictor in which the old value is mapped into the current iterate. Predictors that incorporate old time

derivatives include the forward Euler and Adams-Bashforth methods, e.g.,

$$\begin{aligned} \phi^{k+1} &= \phi^n, && (5.58)\\ &= \phi^n + \Delta t \dot{\phi}^n, && (5.59)\\ &= \phi^n + \frac{\Delta t^n}{2}((2 + \frac{\Delta t^n}{\Delta t^{n-1}})\dot{\phi}^n - \frac{\Delta t^n}{\Delta t^{n-1}}\dot{\phi}^{n-1}). && (5.60) \end{aligned}$$

## 5.3.1. Upwind Interpolation for Convection

We currently support several upwind interpolations for convection. The upwind methods are blended with a centered scheme that becomes dominant below a specified cell-Peclet number.

### *5.3.1.1. First Order Upwind*

The first scheme is a simple first-order scheme that considers the two nodes adjacent to a control volume face and extrapolates from the node in the upwind direction.

$$\dot{m}\overline{\phi}_{upw} = \frac{1}{2}\left(\dot{m} + |\dot{m}|\right)\phi_L + \frac{1}{2}\left(\dot{m} - |\dot{m}|\right)\phi_R \tag{5.61}$$

The convention is that flow leaves the control volume to the left (L) and enters the control volume to the right (R). If the mass flow rate at the face is negative in value, then the node to the right will be selected.

### *5.3.1.2. Blending Function*

The user specified upwind factor controls the blending between the pure upwind operator and a blended user-chosen upwind/central operator.

$$\dot{m}\overline{\phi} = \eta\dot{m}\overline{\phi}_{upw} + (1-\eta)\left(\chi\dot{m}\overline{\phi}_{upw^{sp}} + (1-\chi)\,\dot{m}\overline{\phi}_{cen}\right), \tag{5.62}$$

where $\eta$ is the user specified first order upwind factor and $\overline{\phi}_{upw^{sp}}$ represents the user specified upwind operator, e.g., MUSCL, modified skew upwind, and even pure upwind.

The centered average of $\phi$ is computed from the shape functions, so it is based on all nodes in an element. The shape functions are evaluated at the sub-face centroid. The cell-Peclet number, $\mathrm{Pe}_{\Delta x}$, is used in the blending function (see Figure 5.3-1)

$$\chi = \frac{\left(\zeta \mathrm{Pe}_{\Delta \mathrm{x}}\right)^2}{5 + \left(\zeta \mathrm{Pe}_{\Delta \mathrm{x}}\right)^2}. \tag{5.63}$$

The hybrid upwind factor, $\zeta$, allows one to modify the functional blending function; values of unity result in the normal blending function response in Figure 5.3-1; values of zero yield a pure central operator, i.e., blending function = 0.0; values $>>$ 1 result in a blending function value of unity, i.e.,

pure upwind. The constant $A$ is implemented as above with a value of 5. This value can not be changed via the input file.

The cell-Peclet number is computed for each sub-face in the element from the two adjacent left (L) and right (R) nodes.

$$\mathrm{Pe}_{\Delta x} = \frac{\frac{1}{2}\left(u_{R,i} + u_{L,i}\right)\left(x_{R,i} - x_{L,i}\right)}{\nu} \tag{5.64}$$

A dot-product is implied by repeated indices.

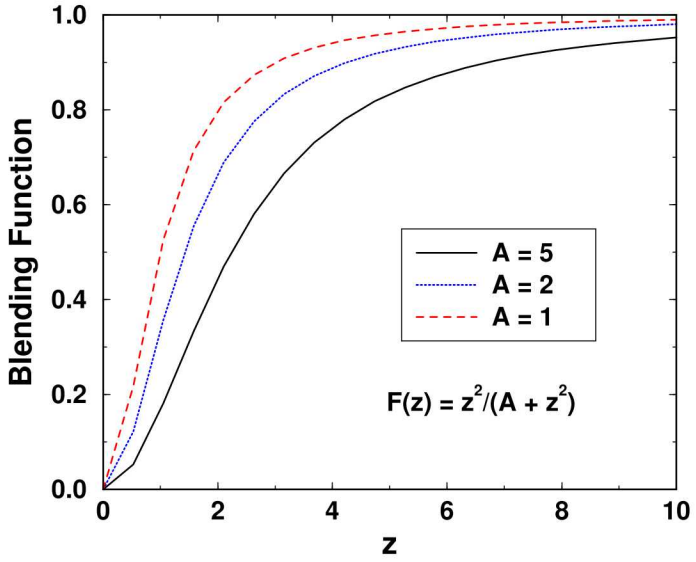


**Figure 5.3-1.. Cell-Peclet number blending function.**

### *5.3.1.3. Modified Linear Profile Skew Upwind*

Modified linear profile skew upwinding is a simplification to the skew upwinding approach in the FIELDS scheme [132, 133]. We omit the physical advection correction terms. Integration point values at control volume subfaces are interpolated from upwind intersection points on the element face. In the original skew upwind scheme, the intersection point could either be interior subface or element faces. The interpolation coefficients were computed by inverting a matrix relation between integration point values and nodal values. The linear profile skew upwinding does not use interior subface intersections – only element face intersections. The modified scheme throws out nodes on an element face that are downwind of an interior subface as shown in Figure 5.3-2.

### *5.3.1.4. MUSCL*

The MUSCL approach (see Chap. 21 of Hirsch [152]) for higher order upwinding is adapted to unstructured meshes. The upwind interpolation is constructed along each edge of an element. The interpolation makes use of the two end nodes of the edge and the centered gradient constructed at the

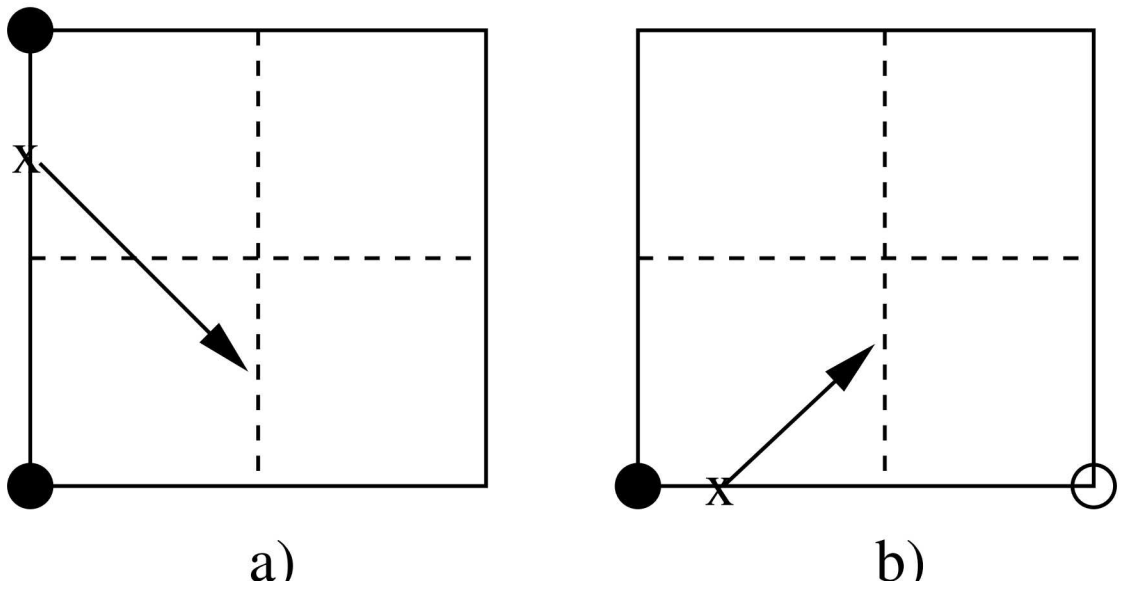


**Figure 5.3-2.. Linear profile skew upwind scheme: a) all nodes on the intersected element face are upwind of the subface, b) omit nodes on intersected element face that are downwind of the subface.**

two end nodes. The MUSCL approach constructs an interpolation in one dimension from four (or more) uniformly distributed nodal values. The two edge nodes are $\phi_i$ and $\phi_{i+1}$. The two other nodal values, $\phi_{i-1}$ and $\phi_{i+2}$, are interpolated from the unstructured mesh using the nodal gradient information.

The MUSCL scheme constructs left and right interpolants at the subface of the control volume. Without the limiter functions, the interpolation is

$$\phi^L_{i+1/2} = \phi_i + \frac{1}{4}\left[(1-\kappa)(\phi_i - \phi_{i-1}) + (1+\kappa)(\phi_{i+1} - \phi_i)\right], \quad (5.65)$$

$$\phi^R_{i+1/2} = \phi_{i+1} - \frac{1}{4}\left[(1+\kappa)(\phi_{i+1} - \phi_i) + (1-\kappa)(\phi_{i+2} - \phi_{i+1})\right], \quad (5.66)$$

where the $(i+1/2)$ location is between node $i$ and node $i+1$. On a uniform mesh, $\kappa = 1/3$ gives a third-order scheme. A second-order upwind scheme is recovered with $\kappa = -1$ and a centered scheme is recovered with $\kappa = 1$.

Limiter functions are introduced to prevent numerical oscillations from occurring.

$$\tilde{\phi}^L_{i+1/2} = \phi_i + \frac{1}{4}\left[(1-\kappa)\,\Phi\left(\frac{1}{r_L}\right)(\phi_i - \phi_{i-1}) + (1+\kappa)\,\Phi(r_L)(\phi_{i+1} - \phi_i)\right], \quad (5.67)$$

$$\tilde{\phi}^R_{i+1/2} = \phi_{i+1} - \frac{1}{4}\left[(1+\kappa)\,\Phi(r_R)(\phi_{i+1} - \phi_i) + (1-\kappa)\,\Phi\left(\frac{1}{r_R}\right)(\phi_{i+2} - \phi_{i+1})\right], \quad (5.68)$$

where

$$r^L = \frac{\phi_i - \phi_{i-1}}{\phi_{i+1} - \phi_i}, \tag{5.69}$$

$$r^R = \frac{\phi_{i+2} - \phi_{i+1}}{\phi_{i+1} - \phi_i}. \tag{5.70}$$

The limiters are selected to be symmetric such that

$$\Phi\left(r\right) = r\Phi\left(\frac{1}{r}\right). \tag{5.71}$$

The limited interpolation functions are

$$\tilde{\phi}^L_{i+1/2} = \phi_i + \frac{1}{2}\Phi\left(r^L\right)\left(\phi_{i+1} - \phi_i\right), \tag{5.72}$$

$$\tilde{\phi}^R_{i+1/2} = \phi_{i+1} - \frac{1}{2}\Phi\left(r^R\right)\left(\phi_{i+1} - \phi_i\right). \tag{5.73}$$

The interpolation for the points off the element edge is

$$\left(\phi_i - \phi_{i-1}\right) = 2\nabla\phi_i\Delta x_{i+1/2} - \left(\phi_{i+1} - \phi_i\right), \tag{5.74}$$

$$\left(\phi_{i+2} - \phi_{i+1}\right) = 2\nabla\phi_{i+1}\Delta x_{i+1/2} - \left(\phi_{i+1} - \phi_i\right), \tag{5.75}$$

where $\Delta x_{i+1/2} = x_{i+1} - x_i$ is the distance vector along the element edge. Symmetric limiter functions are:

$$\text{VanLeer}: \qquad \Phi(r) = \frac{r + |r|}{1 + |r|}, \tag{5.76}$$

$$\text{VanAlbada}: \qquad \Phi(r) = \frac{r + r^2}{1 + r^2}, \tag{5.77}$$

$$\text{superbee}: \qquad \Phi(r) = \max(0, \min(2r, 1), \min(r, 2)). \tag{5.78}$$

#### 5.3.1.5. Convection at an Inflow and Outflow Boundary

At an open boundary, the first-order and LPS upwind schemes only make use of information on the boundary.

For the MUSCL scheme with the flow leaving the domain at node $i$, the usual flux limiters are not used. The slopes are compared between $(\phi_i - \phi_{i-1})$ and $(\phi_{i-1} - \phi_{i-2})$. If the slopes are the same sign, the unlimited second order upwinding is used. If the slopes are different, then a local interpolation is used.

Estimate the slope $(\phi_{i-1} - \phi_{i-2}) = 2\Delta x \nabla\phi_2 - (\phi_i - \phi_{i-1})$, where $\Delta x_i = x_i - x_{i-1}$ is the distance vector along the element edge. For slopes of the same sign, use a second-order scheme,

$$\tilde{\phi}_i^L = \phi_i + \frac{1}{2}\left(\nabla\phi_{i-1}\Delta x_i - (\phi_i - \phi_{i-1})\right), , , , \tag{5.79}$$

else, use a first-order scheme,

$$\tilde{\phi}_i^L = \phi_i - \frac{1}{2}\left[(\phi_i - \phi_{i-1})\right]. \tag{5.80}$$

The boundary is the left (L) side. If the flow enters the domain, then use the local value of $\phi_i$.

#### 5.3.1.6. Nonlinear stabilization operator

The "nonlinear stability operator" (NSO) in Fuego is an artifical viscosity method where the added diffusivity is based on a scaled, pointwise evaluated residual. For a dual volume ($\Omega_n$), associated with a node $n$, the weak form of the NSO for a scalar variable $q$ is

$$\int_{\partial\Omega_n} \nu(R)\left(\partial_{x^i} q\right)\, g^{ij}\, \mathrm{d}S_j, \text{ where } g^{ij} = \frac{\partial x^i}{\partial \xi^k}\frac{\partial x^j}{\partial \xi^k}. \tag{5.81}$$

where $\nu$ depends on the evalaution of a local residual $R$ and the gradient of $q$ as

$$\nu = \sqrt{\frac{R^2}{\frac{\partial q}{\partial x^i} g^{ij} \frac{\partial q}{\partial x^j}}}. \tag{5.82}$$

The local residual can be taken, similar to Shakib[153] but in an incompressible context, as the full residual of the PDE. For a conserved scalar,$q$, with diffusivity $\Gamma$, $R$ would be

$$R = \left[(\text{Time})_\rho + (\text{Adv.})_{\rho\mathbf{u}} - (\text{Diff.})_{\rho\Gamma}\right] q, \tag{5.83}$$

with discrete operators representing the individual terms of the advection-diffusion equation. For an equation with a source term, it would also need appear in the local residual calculation. Another possibility for choosing $R$ would be based on the error of performing the chain-rule on the advection operator.

$$R = \tilde{G}_i\left(\rho u^i q\right) - \left[\tilde{I}\left(\rho u^i\right)\tilde{G}_i q + \left(\tilde{I} q\right) G_i\left(\rho u^i\right)\right] \tag{5.84}$$

where $\tilde{G}$ and $\tilde{I}$ represent interpolation and gradients evaluated at an integration point. Both options are available in Fuego.

The NSO computed from such residuals can add an unnecessarily large amount of dissipation in some cases. For this reason, we limit the NSO coefficient to the upwind value as

$$\nu = \min\left(\nu(R), \frac{1}{10}\left(\rho u\right)^i g_{ij}\left(\rho u\right)^j\right). \tag{5.85}$$

where $g_{ij} = \left[g^{ij}\right]^{-1}$. Additionally, as it's based on the mesh discretization error, the NSO coefficient tends to vary strongly on short length scales. For numerical robustness, we average the NSO viscosity

over control volumes, and then interpolate back to the subcontrol surfaces to evaluate the diffusion term; that is,

$$\overline{\nu}_{\text{ip}} = \tilde{I}\,\overline{\frac{\nu\|g^{ij}\|}{\|g^{ij}\|}}. \tag{5.86}$$

This operation effectively smooths the NSO viscosity over a patch of elements. The nonlinear stabilization viscosity is not included at the boundaries.

### 5.3.2. Variable Density

The discretization of the time derivative requires special attention for variable density flows. The density time-derivative in the continuity equation must be predicted in a continuous manner. The density at the new time level in the convection terms and the transport equation time terms must also be predicted.

The transport equations are solved in conservative form, so density appears in the time derivative. With a segregated solution strategy, the density at the new time level is not available until the transport equations have been solved once. A density predictor is required. A generic time term is written as

$$\frac{\partial \rho\phi}{\partial t} \approx \frac{\rho^{n+1}\phi^{n+1} - \rho^n\phi^n}{\Delta t} \approx \rho^n \frac{\phi^{n+1} - \phi^n}{\Delta t} + \phi^{n+1}\frac{\rho^* - \rho^n}{\Delta t} \tag{5.87}$$

There are two approaches to estimating the new density. The simplest approach is to use the most recent value. The other approach is to use a density predictor. The predicted value of density at the new time level, $\rho^*$, is computed from the old density and the current density time derivative. Introduce the nodal variable $\Upsilon$ for the discrete density time-derivative such that

$$\Upsilon^* = \frac{\rho^* - \rho^n}{\Delta t} \tag{5.88}$$

$$\rho^* = \rho^n + \Delta t \Upsilon^* \tag{5.89}$$

The density derivative, $\Upsilon^*$, is always updated at the bottom of the transport equation loop after a new set of temperatures and mass fractions is available. The two approaches are different for the first nonlinear sub-iteration within a time step, but yield equivalent values upon subsequent sub-iterations. The new density is also computed at the bottom of the equation loop. This value is ignored upon subsequent sub-iterations if using the density predictor. But, this new density value will get copied to the old time level when the time step is advanced. It is important to note that this new "old" velocity is not consistent with the density that was used in the old transport equations, but it seems critical to the success of this approach to do so.

For the first nonlinear iteration within a time step, the effect of the density at the new time level is predicted by carrying forward the best approximation of the density time-derivative from the last time step. The continuity equation is implemented as

$$\int \Upsilon^* dV + \int \rho^* \overline{u_i}^{n+1} n_i dS = 0, \tag{5.90}$$

where the density time-derivative is the most recent value and the density in the convection is estimated in the same manner as the transport equations. The density time-derivative, $\Upsilon$, must be stored as a persistent nodal variable in order to have a good estimate for the continuity equation from step to step.

### 5.3.3. Open Boundary Conditions

Open boundary conditions are used for boundaries where the flow can go either in or out. The direction of the flow is determined by the local force balance. In this documentation, the open boundary condition is also referred to as the outflow boundary condition. There are two parts to the outflow boundary condition. The first part concerns computing a velocity field that satisfies continuity. The second part concerns selecting the proper convected scalar value depending if the flow is in or out of the domain. Control volume balances are implemented at open boundaries for continuity, momentum, and the other transport equations.

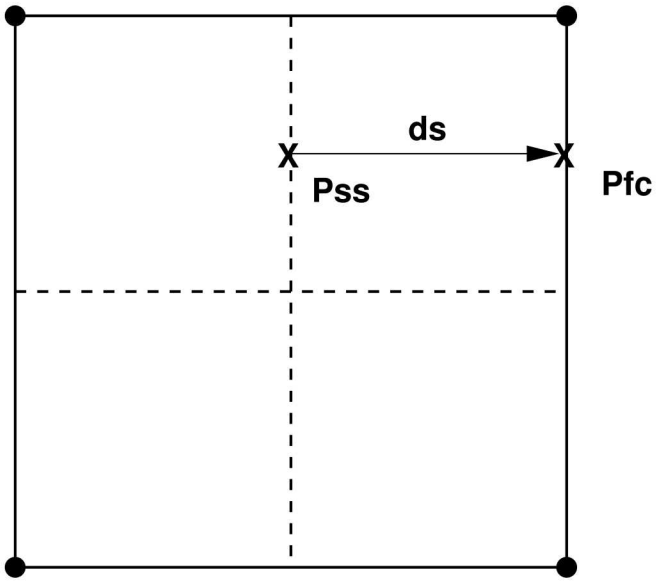


**Figure 5.3-3.. Boundary mass flux integration locations.**

A fixed pressure value is specified for the continuity and momentum equations. The nodal values of pressure on the boundary are allowed to float. A mass flux condition is formulated at the boundary in order to drive the boundary pressures towards the specified boundary pressure and to provide a boundary mass flow rate for the other transport equations. The form of the boundary mass flux is similar to the pressure-stabilized interior mass fluxes (see section 5.1). The equation for the mass flux at a boundary face, shown in Figure 5.3-3, is

$$\dot{m}_{\mathrm{bc}} = \rho \overline{u}_i^{n+1} n_i \mathrm{d}S \tag{5.91}$$

and the interpolation formula for a single velocity component is

$$\overline{u}^{n+1} = \sum_{bc} N_i U_i^{**} + f\frac{\Delta t}{\rho}\left(\sum_{bc} N_i \sum_j G_{ij} p_j^* - \left(\frac{P_{fc} - P_{ss}^{n+1}}{\Delta s}\right)\frac{\Delta x}{\Delta s}\right) + f\left(\overline{u}^n - \sum_{bc} N_i U_i^n\right) \tag{5.92}$$

The upper case velocities, $U_i$, are nodal velocities, while the lower case velocity, $\overline{u}$, is the boundary velocity. The average pressure, $P_{ss}$, is computed at the opposing subface centroid and evaluated at the new time level, $n+1$. The boundary pressure, $P_{fc}$, is evaluated at the boundary subface centroid and is the "specified" pressure. The operator, $G_{ij}$, is the discrete gradient operator for node $i$. In the case of the semi-discrete formulation, the last term is dropped in Equation 5.92 and $f = 1$.

The nodal pressure gradient is required for the momentum balance and the boundary mass flux formulation. The nodal pressure gradient is constructed by a discrete Gauss divergence relation over the control volumes. The pressure at most control volume subfaces is interpolated from the nodes of the parent element, even over inflow, wall, and symmetry boundaries. For outflow boundaries, the specified boundary pressure, $P_{fc}$, is used.

Nodal velocities on open boundaries are corrected with the projection.

On pressure-specified open boundaries, the flow will sometimes exit and reenter the domain through some sort of entrainment process. The process will look non-physical and is due to the artificially imposed constant pressure. A method of counteracting the reentrance problem is to turn off the convection terms in the momentum equations for control-volume subfaces which have reentrant flow. This condition is optional and can be set on a side-set basis.

If the flow is entrained into the domain, then far-field values must be specified for the scalar variables.

## 5.4. SEGREGATED SOLUTION PROCEDURE

The time integration method is a two-level, backward Euler scheme, requiring data at two time states. The discrete form of the nonlinear equations is

$$\frac{\phi^{n+1} - \phi^n}{\Delta t} = F\left(\phi^{n+1}, \phi^n\right). \tag{5.93}$$

Sub-iteration is required within the time step to satisfy the nonlinearities. Over one sub-iteration, the nonlinear equations are solved in a segregated manner. Each segregated equation set is linearized and solved as a linear problem. During the nonlinear iteration process, a temporary variable may be introduced to differentiate the old guess at the state (n+1) from the new guess at the state (n+1). A temporary variable (*) is introduced to hold the new estimate of the state (n+1). The temporary variable is typically only used in describing the algorithm. Functionally, the (*) variables and (n+1) variables are usually represented by the same array within the code. The only time a temporary variable would be used in the code is if the momentum equations were segregated or if the species diffusion velocities were not pre-computed.

Within the transport equations, the convection terms are linearized by freezing the mass flux (density * velocity * area).

The SIERRA framework provides services to manage the state data between the two time levels. The SIERRA framework services are insufficient because they only swap pointers. The result of the swap is that the estimate of the new solution at time (n+1) uses the solution at (n-1) instead of (n), which is too far away. After the pointer-swap, the SIERRA/Fuego code additionally copies forward the solution at (n) into the initial guess at (n+1). The array-copy occurs only at the beginning of the sub-iteration process. The SIERRA/Fuego code also manages the updating between (*) and (n+1) for the delta-form of the linear system.

The material properties are evaluated at the top of a nonlinear sub-iteration. Density is a STATE property since it has a time derivative in the continuity equation if properties are variable. Density will always be treated as a state variable, even if it is constant. All other properties are treated as TEMPORARY variables. The general workset algorithm that computes properties evaluates them at the most recent guess of the (n+1) state. There is an additional workset algorithm that evaluates state properties at both state (n) and (n+1). The state property evaluation is only performed during the initialization phase. All material properties are evaluated at the nodes. Sub-face and sub-volume values are averaged using the element shape functions.

A linear solve is performed for each equation set within a nonlinear sub-iteration. There is a solver object associated with each equation set within the SIERRA framework. The solver object contains the matrix connectivity and manages the assembly of the matrix components. There will be ten solver objects for the full turbulent combustion mechanics (the species equations all use the same solver object). There will also be ten repeated sets of connectivity information.

The ordering of the segregated equations during one nonlinear iteration is given in the following list. Reduced equation sets for simplified mechanics maintain the same relative ordering.

1. evaluate material properties using the most recent estimate of temperature and composition
2. evaluate turbulent eddy viscosity if turbulent
3. evaluate combustion model species production rates
4. evaluate soot model production rates
5. evaluate gas and soot absorptivity for radiation model
6. solve x-momentum equation, store new predicted x-velocity until all momentum equations have been evaluated
7. solve y-momentum equation, store new predicted y-velocity until all momentum equations have been evaluated
8. solve z-momentum equation, store new predicted z-velocity until all momentum equations have been evaluated
9. update predicted velocities
10. solve continuity equation using predicted velocities, update new pressure

11. update new mass fluxes at all control volume sub-faces, including boundaries, and use in subsequent transport equations
12. perform the velocity projection and correct all nodal velocities
13. assemble turbulence friction velocity
14. solve the turbulent kinetic energy equation if turbulent, store turbulent kinetic energy until turbulence dissipation equation is solved so that the production and dissipation source terms can be properly linearized
15. solve the turbulence dissipation equation if turbulent, update the turbulent kinetic energy and turbulence dissipation
16. solve the enthalpy equation
    - laminar: solve for temperature
    - turbulent: solve for enthalpy
17. solve each species equation, do not update species mass fractions until all species equations have been solved
18. solve the soot equation, store soot mass fraction until soot nuclei equation is solved
19. solve the soot nuclei equation, update soot mass fraction and soot nuclei mass fraction
20. compute Nth species mass fraction using summation rule
21. update temperature or enthalpy at new time level
    - laminar: compute enthalpy
    - turbulent: extract temperature
22. extract temperature from enthalpy if laminar
23. compute new density and time derivative of density

This procedure is repeated within a time step until the desired level of nonlinear equation convergence is achieved.

## 5.5. DISCRETE TRANSPORT EQUATIONS

The discrete form of the linearized equations are presented in this section. The nonlinear solution procedure consists of repeated approximate Newton linearizations and linear solves of the discrete equation,

$$A\delta\phi = b. \tag{5.94}$$

The matrix $A$ is based on an approximate linearization of $F$ from Equation 5.93 about a predicted value $\phi^*$,

$$A = \frac{1}{\Delta t} - \left.\frac{\partial \tilde{F}}{\partial \phi}\right|^* . \tag{5.95}$$

The right-hand side, $b$, of the linearized equation represents the residual of the nonlinear equation,

$$b = F\left(\phi^*, \phi^n\right) - \frac{\phi^* - \phi^n}{\Delta t} \tag{5.96}$$

If the nonlinear equation is converged, the right-hand side will be zero. The linear equations are solved in delta-form. The solution vector consists of the change in the unknown rather than the new value of the unknown.

There are four solution states in the nonlinear solver algorithm. The time level $n$ is the old time level. The state $*$ represents predicted values at the new time level before the linear solve. The state $**$ represents the values after the linear solve. The time level $(n+1)$ is the new time level. Within the nonlinear iteration cycle, values at the new time level $(n+1)$ are copied to the predicted level $*$ before the next iteration.

There are three stages to the assembly of the matrix that result from the linearization. The first stage is the assembly of element contributions. The elements contain control-volume sub-faces that are internal to the mesh. The second state is the assembly of flux boundary conditions. The flux boundary conditions contribute to the control-volume sub-faces on the boundary of the mesh. The flux boundary condition contributions are full element contributions because they may involve both boundary and interior nodes. The third stage is the enforcement of Dirichlet boundary conditions.

The element matrix contributions are processed by first evaluating surface integral fluxes at sub-faces and then evaluating volume integral terms at sub-volumes. The flux is evaluate at a sub-face and then added or subtracted from the two adjacent control-volumes. The sub-face area components are constructed such that the face normal direction points from the left adjacent node to the right adjacent node. Fluxes are subtracted from the left node (L) and added to the right node (R). The left and right adjacent nodes for a give sub-face number within an element are given in Tables 5.8-8, 5.8-13, and 5.8-18.

The linearization of each transport equation can be broken into contributions from the time term, convection, diffusion, and sources.

$$A = A^t + A^c + A^d + A^s \tag{5.97}$$

$$b = b^t + b^c + b^d + b^s \tag{5.98}$$

The linear system is assembled on an element-by-element basis. Each element contributes and $N \times N$ element matrix where $N$ is the number of nodes in the element. The nodal contribution from node $J$ for the control volume about node $I$ is $A_{I,J}$. Nodal variables in the following discussion are symbolized by capital letters. Linear averages of variables at face $k$ are

$$\mu_k = \sum_J N_J|_k \, \mu_J \tag{5.99}$$

$$\kappa_k = \sum_J N_J|_k \, \kappa_J \tag{5.100}$$

The density predictor (see Section 5.3.2) may be used to compute the density at the new time level for the time derivative term.

The convection operator for a face $i$ is $C_{i,J}$ and is described in Section 5.3.1.

Gradients of variables at face $k$ are:

$$p_x = \sum_J \left.\frac{\partial N_J}{\partial x}\right|_k P_J \qquad p_y = \sum_J \left.\frac{\partial N_J}{\partial y}\right|_k P_J \qquad p_z = \sum_J \left.\frac{\partial N_J}{\partial z}\right|_k P_J \tag{5.101}$$

$$u_x = \sum_J \left.\frac{\partial N_J}{\partial x}\right|_k U_J \qquad u_y = \sum_J \left.\frac{\partial N_J}{\partial y}\right|_k U_J \qquad u_z = \sum_J \left.\frac{\partial N_J}{\partial z}\right|_k U_J \tag{5.102}$$

$$v_x = \sum_J \left.\frac{\partial N_J}{\partial x}\right|_k V_J \qquad v_y = \sum_J \left.\frac{\partial N_J}{\partial y}\right|_k V_J \qquad v_z = \sum_J \left.\frac{\partial N_J}{\partial z}\right|_k V_J \tag{5.103}$$

$$w_x = \sum_J \left.\frac{\partial N_J}{\partial x}\right|_k W_J \qquad w_y = \sum_J \left.\frac{\partial N_J}{\partial y}\right|_k W_J \qquad w_z = \sum_J \left.\frac{\partial N_J}{\partial z}\right|_k W_J \tag{5.104}$$

$$t_x = \sum_J \left.\frac{\partial N_J}{\partial x}\right|_k T_J \qquad t_y = \sum_J \left.\frac{\partial N_J}{\partial y}\right|_k T_J \qquad t_z = \sum_J \left.\frac{\partial N_J}{\partial z}\right|_k T_J \tag{5.105}$$

### 5.5.1. Positive-Flow Convention and Integration Quadrature

The sign on a flux integral is defined such that flow into a control volume is positive and flow out of a control volume is negative. The equations are assembled into the implicit matrix and right-hand side such that the time derivative contribution of an unknown is positive. In reference to the model differential equation, Equation 5.93, any implicit terms that contribute to the control volume balance, $F(\phi)$, in a positive sense must be moved to the implicit left-hand side, switching signs.

The control volume balance is assembled on an element-by-element basis. Each element contributes terms from fluxes over its internal sub-control volume faces and volumetric terms from its internal sub-control volumes. A flux is computed for each sub-control volume face. The flux contribution is then summed into the two adjacent control volumes, adjusting the sign according to whether the flux is in or out of the control volume. The convention is that the sub-face normal direction between two adjacent control volumes is positive from the lower local sub-volume number to the higher sub-volume number in a local node numbering sense. The consistent treatment of fluxes is a requirement for conservation. Each sub-control volume face is numbered the same as the element edge number. The two adjacent control volumes for each edge number are given in Tables 5.8-8, 5.8-13, and 5.8-18 for different element types.

The elemental flux contributions are assembled into a global control volume matrix. Each control volume balance is written in terms of coefficients multiplying the surrounding nodal values. In terms of matrix terminology for two-dimensional elements, the matrix coefficient for Node 5 of Figure 5.0-1, associated with the control volume center, is the diagonal term and should be positive. All other nodal coefficients for the control volume balance are the off-diagonal terms and complete one row of a global flux-balance matrix.

The control volume flux integrals are evaluated using numerical quadrature. The integral term for each control volume sub-face and sub-volume is evaluated using a single quadrature point. The number of quadrature points for the surface fluxes in an element is equivalent to the number of sub-faces. For example, a quadrilateral element will have four sub-face quadratures and four sub-volume quadratures. A hexahedral element will have twelve sub-face quadratures and eight sub-volume quadratures.

In three-dimensional elements, the control-volume sub-faces may not be planar. Care must be taken to conserve surface area over a control-volume to prevent non-physical sources and sinks. The sub-faces in a three-dimensional element are defined by bilinear surfaces and the discrete surface area differential is also a bilinear function. Since the quadrature for a bilinear function is exact if evaluated at the mid-point, the current quadrature strategy will ensure surface area conservation.

The quadrature coefficients are customarily derived such that the integration ranges from $-1$ to 1, so a mapping is required to quadrature space.

$$\int_a^b F\left(\xi\right) \mathrm{d}\xi = \frac{b-a}{2} \int_{-1}^{1} f\left(\bar{\xi}\right) \mathrm{d}\bar{\xi} \tag{5.106}$$

$$\tag{5.107}$$

$$\xi = \frac{b+a}{2} + \frac{b-a}{2}\bar{\xi} \tag{5.108}$$

The integrand is evaluated at discrete points, called Gauss points, and summed using weighting functions.

$$\int_{-1}^{1} F\left(\bar{\xi}\right) \mathrm{d}\bar{\xi} = w_i F\left(\bar{\xi}_i\right) \tag{5.109}$$

For a one-point quadrature, $\bar{\xi}_1 = 0$ and $w_1 = 2$.

### 5.5.2. X-Momentum, 3D Laminar Transport

The time term is lumped. The time-term contribution is evaluated for each sub-volume.

$$A^t_{I,I} \mathrel{+}= \rho^*_I \frac{\Delta V_I}{\Delta t} \tag{5.110}$$

$$b^t_I \mathrel{-}= \left(\rho^*_I U^*_I - \rho^n_I U^n_I\right) \frac{\Delta V_I}{\Delta t} \tag{5.111}$$

The convection term is computed at each face $k$ and assembled to the left (IL) and right (IR) control volumes.

$$A^c_{IL,J} \mathrel{+}= C^*_{k,J} \tag{5.112}$$

$$A^c_{IR,J} \mathrel{-}= C^*_{k,J} \tag{5.113}$$

$$b^c_{IL} \; -= \; \sum_J C^*_{k,J} U^*_J \tag{5.114}$$

$$b^c_{IR} \; += \; \sum_J C^*_{k,J} U^*_J \tag{5.115}$$

The viscous stress term is computed at each face $k$ and assembled to the left (IL) and right (IR) control volumes. Only the solenoidal part of the stress term is used for the matrix. The stress term may or may not include the molecular viscosity, depending on the user specified model.

$$F_{k,J} \;\; = \;\; -\mu_k \left( \left.\frac{\partial N_J}{\partial x}\right|_k A_x + \left.\frac{\partial N_J}{\partial y}\right|_k A_y + \left.\frac{\partial N_J}{\partial z}\right|_k A_z \right) \tag{5.116}$$

$$A^d_{IL,J} \; += \; F_{k,J} \tag{5.117}$$

$$A^d_{IR,J} \; -= \; F_{k,J} \tag{5.118}$$

$$\tau_{xx} \;\; = \;\; \mu_k \left( u^*_x + u^*_x \right) \tag{5.119}$$

$$\tau_{xy} \;\; = \;\; \mu_k \left( u^*_y + v^*_x \right) \tag{5.120}$$

$$\tau_{xz} \;\; = \;\; \mu_k \left( u^*_z + w^*_x \right) \tag{5.121}$$

$$f_k \;\; = \;\; -\left( \tau_{xx} A_x + \tau_{xy} A_y + \tau_{xz} A_z \right) \tag{5.122}$$

$$b^d_{IL} \; -= \; f_k \tag{5.123}$$

$$b^d_{IR} \; += \; f_k \tag{5.124}$$

The pressure is assembled in the form of a volume integral. The pressure gradients have been pre-computed at nodes use a surface-integral approximation.

$$b^s_I \; -= \; \left.\frac{\partial P}{\partial x}\right|^*_I \Delta V_I \tag{5.125}$$

### 5.5.3. Y-Momentum, 3D Laminar Transport

The time term is lumped. The time-term contribution is evaluated for each sub-volume.

$$A^t_{I,I} \; += \; \rho^*_I \frac{\Delta V_I}{\Delta t} \tag{5.126}$$

$$b^t_I \; -= \; \left( \rho^*_I V^*_I - \rho^n_I V^n_I \right) \frac{\Delta V_I}{\Delta t} \tag{5.127}$$

The convection term is computed at each face $k$ and assembled to the left (IL) and right (IR) control volumes.

$$A^c_{IL,J} \quad += \quad C^*_{k,J} \tag{5.128}$$
$$A^c_{IR,J} \quad -= \quad C^*_{k,J} \tag{5.129}$$

$$b^c_{IL} \quad -= \quad \sum_J C^*_{k,J} V^*_J \tag{5.130}$$
$$b^c_{IR} \quad += \quad \sum_J C^*_{k,J} V^*_J \tag{5.131}$$

The viscous stress term is computed at each face $k$ and assembled to the left (IL) and right (IR) control volumes. Only the solenoidal part of the stress term is used for the matrix.

$$F_{k,J} \quad = \quad -\mu_k \left( \left.\frac{\partial N_J}{\partial x}\right|_k A_x + \left.\frac{\partial N_J}{\partial y}\right|_k A_y + \left.\frac{\partial N_J}{\partial z}\right|_k A_z \right) \tag{5.132}$$
$$A^d_{IL,J} \quad += \quad F_{k,J} \tag{5.133}$$
$$A^d_{IR,J} \quad -= \quad F_{k,J} \tag{5.134}$$

$$\tau_{yx} \quad = \quad \mu_k \left( v^*_x + u^*_y \right) \tag{5.135}$$
$$\tau_{yy} \quad = \quad \mu_k \left( v^*_y + v^*_y \right) \tag{5.136}$$
$$\tau_{yz} \quad = \quad \mu_k \left( v^*_z + w^*_y \right) \tag{5.137}$$
$$f_k \quad = \quad -\left( \tau_{yx} A_x + \tau_{yy} A_y + \tau_{yz} A_z \right) \tag{5.138}$$
$$b^d_{IL} \quad -= \quad f_k \tag{5.139}$$
$$b^d_{IR} \quad += \quad f_k \tag{5.140}$$

The pressure is assembled in the form of a volume integral. The pressure gradients have been pre-computed at nodes use a surface-integral approximation.

$$b^s_I \quad -= \quad \left.\frac{\partial P}{\partial y}\right|^*_I \Delta V_I \tag{5.141}$$

### 5.5.4. Z-Momentum, 3D Laminar Transport

The time term is lumped. The time-term contribution is evaluated for each sub-volume.

$$A^t_{I,I} \mathrel{+}= \rho^*_I \frac{\Delta V_I}{\Delta t} \tag{5.142}$$

$$b^t_I \mathrel{-}= \left(\rho^*_I W^*_I - \rho^n_I W^n_I\right) \frac{\Delta V_I}{\Delta t} \tag{5.143}$$

The convection term is computed at each face $k$ and assembled to the left (IL) and right (IR) control volumes.

$$A^c_{IL,J} \mathrel{+}= C^*_{k,J} \tag{5.144}$$
$$A^c_{IR,J} \mathrel{-}= C^*_{k,J} \tag{5.145}$$

$$b^c_{IL} \mathrel{-}= \sum_J C^*_{k,J} W^*_J \tag{5.146}$$
$$b^c_{IR} \mathrel{+}= \sum_J C^*_{k,J} W^*_J \tag{5.147}$$

The viscous stress term is computed at each face $k$ and assembled to the left (IL) and right (IR) control volumes. Only the solenoidal part of the stress term is used for the matrix.

$$F_{k,J} = -\mu_k \left( \left.\frac{\partial N_J}{\partial x}\right|_k A_x + \left.\frac{\partial N_J}{\partial y}\right|_k A_y + \left.\frac{\partial N_J}{\partial z}\right|_k A_z \right) \tag{5.148}$$
$$A^d_{IL,J} \mathrel{+}= F_{k,J} \tag{5.149}$$
$$A^d_{IR,J} \mathrel{-}= F_{k,J} \tag{5.150}$$

$$\tau_{zx} = \mu_k \left(w^*_x + u^*_z\right) \tag{5.151}$$
$$\tau_{zy} = \mu_k \left(w^*_y + v^*_z\right) \tag{5.152}$$
$$\tau_{zz} = \mu_k \left(w^*_z + w^*_z\right) \tag{5.153}$$
$$f_k = -\left(\tau_{zx} A_x + \tau_{zy} A_y + \tau_{zz} A_z\right) \tag{5.154}$$
$$b^d_{IL} \mathrel{-}= f_k \tag{5.155}$$
$$b^d_{IR} \mathrel{+}= f_k \tag{5.156}$$

The pressure is assembled in the form of a volume integral. The pressure gradients have been pre-computed at nodes use a surface-integral approximation.

$$b_I^s \; -= \; \left.\frac{\partial P}{\partial z}\right|_I^* \Delta V_I \tag{5.157}$$

## 5.5.5. Buoyancy, Momentum Transport

The body force imposed by the buoyancy term can be constructed in one of three ways.

### 5.5.5.1. Boussinesq Form

For the Boussinesq approximation, the body force is evaluated at the sub-volume centroid, $k$, for sub-volume $I$.

$$b_I^s -= \frac{\rho g}{T_\circ}\left(\sum_J N_J|_k \, T_J - T_\circ\right)\Delta V_I \tag{5.158}$$

### 5.5.5.2. Differential Form

For the “differential" form, the hydrostatic component of pressure has been removed. The body force is evaluated at the control-volume centroid, for sub-volume $I$.

$$b_I^s += \left(\rho_I^* - \rho_\circ\right) g \Delta V_I \tag{5.159}$$

### 5.5.5.3. Full Form

The body force is evaluated at the control-volume centroid, for sub-volume $I$.

$$b_I^s += \rho_I^* g \Delta V_I \tag{5.160}$$

### 5.5.6. Mass Transport – 3D Continuity

The time term is lumped. The time-term contribution is evaluated for each sub-volume.

$$b_I^t \mathrel{-}= (\rho_I^* - \rho_I^n)\frac{\Delta V_I}{\Delta t} \tag{5.161}$$

The convection term is computed at each face $k$ and assembled to the left (IL) and right (IR) control volumes using the Rhie/Chow scheme from Section 5.1.

$$\begin{aligned}
F_{k,J} &= -f\Delta t\left(\left.\frac{\partial N_J}{\partial x}\right|_k A_x + \left.\frac{\partial N_J}{\partial y}\right|_k A_y + \left.\frac{\partial N_J}{\partial z}\right|_k A_z\right) && (5.162)\\
A^d_{IL,J} &\mathrel{+}= F_{k,J} && (5.163)\\
A^d_{IR,J} &\mathrel{-}= F_{k,J} && (5.164)
\end{aligned}$$

$$\begin{aligned}
u_k^* &= \sum_J N_J|_k\, U_J^* + f\frac{\Delta t}{\rho}\left(\sum_J \left.\frac{\partial P}{\partial x}\right|_J^* - p_x^*\right) + f\left(u_k^n - \sum_J N_J|_J\, U_J^n\right) && (5.165)\\
v_k^* &= \sum_J N_J|_k\, V_J^* + f\frac{\Delta t}{\rho}\left(\sum_J \left.\frac{\partial P}{\partial y}\right|_J^* - p_y^*\right) + f\left(v_k^n - \sum_J N_J|_J\, V_J^n\right) && (5.166)\\
w_k^* &= \sum_J N_J|_k\, W_J^* + f\frac{\Delta t}{\rho}\left(\sum_J \left.\frac{\partial P}{\partial z}\right|_J^* - p_z^*\right) + f\left(w_k^n - \sum_J N_J|_J\, W_J^n\right) && (5.167)\\
\dot{m}_k &= \rho\left(u_k^* A_x + v_k^* A_y + w_k^* A_z\right) && (5.168)\\
b^c_{IL} &\mathrel{-}= \dot{m}_k && (5.169)\\
b^c_{IR} &\mathrel{+}= \dot{m}_k && (5.170)
\end{aligned}$$

Velocity correction and new mass flow rate.....

### 5.5.7. Energy, 3D Laminar Transport

The laminar energy equation is linearized with respect to the temperature. The time term is lumped. The time-term contribution is evaluated for each sub-volume. The density must also be linearized for stability.

$$A^t_{I,I} \mathrel{+}= \left(\rho_I^* C_{p,I}^* - \rho_I^* \frac{H_I^*}{T_I^*}\right)\frac{\Delta V_I}{\Delta t} \tag{5.171}$$

$$b_I^t \mathrel{-}= (\rho_I^* H_I^* - \rho_I^n H_I^n)\frac{\Delta V_I}{\Delta t} \tag{5.172}$$

The convection term is computed at each face $k$ and assembled to the left (IL) and right (IR) control volumes.

$$A^c_{IL,J} \quad += \quad C^{n+1}_{k,J} C^*_{p,J} \tag{5.173}$$
$$A^c_{IR,J} \quad -= \quad C^{n+1}_{k,J} C^*_{p,J} \tag{5.174}$$

$$b^c_{IL} \quad -= \quad \sum_J C^{n+1}_{k,J} H^*_J \tag{5.175}$$
$$b^c_{IR} \quad += \quad \sum_J C^{n+1}_{k,J} H^*_J \tag{5.176}$$

The heat conduction term is computed at each face $k$ and assembled to the left (IL) and right (IR) control volumes.

$$F_{k,J} \quad = \quad -\kappa_k \left( \left.\frac{\partial N_J}{\partial x}\right|_k A_x + \left.\frac{\partial N_J}{\partial y}\right|_k A_y + \left.\frac{\partial N_J}{\partial z}\right|_k A_z \right) \tag{5.177}$$
$$A^d_{IL,J} \quad += \quad F_{k,J} \tag{5.178}$$
$$A^d_{IR,J} \quad -= \quad F_{k,J} \tag{5.179}$$

$$q_k \quad = \quad -\kappa_k \left( t^*_x A_x + t^*_y A_y + t^*_z A_z \right) \tag{5.180}$$
$$b^d_{IL} \quad -= \quad q_k \tag{5.181}$$
$$b^d_{IR} \quad += \quad q_k \tag{5.182}$$

### 5.5.8. Temperature, 3D Laminar Transport

The laminar temperature equation is linearized with respect to the temperature. The time term is lumped. The time-term contribution is evaluated for each sub-volume.

$$A^t_{I,I} += \rho^*_I \frac{\Delta V_I}{\Delta t} \tag{5.183}$$

$$b^t_I -= \left( \rho^*_I T^*_I - \rho^n_I T^n_I \right) \frac{\Delta V_I}{\Delta t} \tag{5.184}$$

The convection term is computed at each face $k$ and assembled to the left (IL) and right (IR) control volumes.

$$A^c_{IL,J} \quad += \quad C^{n+1}_{k,J} \tag{5.185}$$
$$A^c_{IR,J} \quad -= \quad C^{n+1}_{k,J} \tag{5.186}$$

$$b^c_{IL} \quad -= \quad \sum_J C^{n+1}_{k,J} T^*_J \tag{5.187}$$
$$b^c_{IR} \quad += \quad \sum_J C^{n+1}_{k,J} T^*_J \tag{5.188}$$

The heat conduction term is computed at each face $k$ and assembled to the left (IL) and right (IR) control volumes.

$$F_{k,J} \quad = \quad -\frac{\kappa_k}{C_{p,k}} \left( \left.\frac{\partial N_J}{\partial x}\right|_k A_x + \left.\frac{\partial N_J}{\partial y}\right|_k A_y + \left.\frac{\partial N_J}{\partial z}\right|_k A_z \right) \tag{5.189}$$
$$A^d_{IL,J} \quad += \quad F_{k,J} \tag{5.190}$$
$$A^d_{IR,J} \quad -= \quad F_{k,J} \tag{5.191}$$

$$q_k \quad = \quad -\frac{\kappa_k}{C_{p,k}} \left( t^*_x A_x + t^*_y A_y + t^*_z A_z \right) \tag{5.192}$$
$$b^d_{IL} \quad -= \quad q_k \tag{5.193}$$
$$b^d_{IR} \quad += \quad q_k \tag{5.194}$$

A correction for variable specific heat is applied as a volume term. The correction is computed at the centroid of the sub-volume, $k$, for control volume $I$.

$$b^d_I += \frac{\kappa}{C^2_p} \left( t_x C_{p,x} + t_y C_{p,y} + t_z C_{p,z} \right) \Delta V_I \tag{5.195}$$

## 5.5.9. Species, 3D Laminar Transport

There is a species equations for each species. The mass fraction is $Y_s$, where $s$ is the species number. The time term is lumped. The time-term contribution is evaluated for each sub-volume.

$$A^t_{I,I} += \rho^*_I \frac{\Delta V_I}{\Delta t} \tag{5.196}$$

$$b_I^t \mathrel{-}= \left(\rho_I^* Y_{s,I}^* - \rho_I^n Y_{s,I}^n\right) \frac{\Delta V_I}{\Delta t} \tag{5.197}$$

The convection term is computed at each face $k$ and assembled to the left (IL) and right (IR) control volumes.

$$A_{IL,J}^c \mathrel{+}= C_{k,J}^{n+1} \tag{5.198}$$
$$A_{IR,J}^c \mathrel{-}= C_{k,J}^{n+1} \tag{5.199}$$

$$b_{IL}^c \mathrel{-}= \sum_J C_{k,J}^{n+1} Y_{s,J}^* \tag{5.200}$$
$$b_{IR}^c \mathrel{+}= \sum_J C_{k,J}^{n+1} Y_{s,J}^* \tag{5.201}$$

The mass diffusion term is computed at each face $k$ and assembled to the left (IL) and right (IR) control volumes.

$$F_{k,J} = -\rho_k D_{s,k} \left( \left.\frac{\partial N_J}{\partial x}\right|_k A_x + \left.\frac{\partial N_J}{\partial y}\right|_k A_y + \left.\frac{\partial N_J}{\partial z}\right|_k A_z \right) \tag{5.202}$$
$$A_{IL,J}^d \mathrel{+}= F_{k,J} \tag{5.203}$$
$$A_{IR,J}^d \mathrel{-}= F_{k,J} \tag{5.204}$$

$$f_k = -\rho_k D_{s,k} \left( y s_x^* A_x + y s_y^* A_y + y s_z^* A_z \right) \tag{5.205}$$
$$b_{IL}^d \mathrel{-}= f_k \tag{5.206}$$
$$b_{IR}^d \mathrel{+}= f_k \tag{5.207}$$

## 5.5.10. X-Momentum, 3D Turbulent Transport

The time term is lumped. The time-term contribution is evaluated for each sub-volume.

$$A_{I,I}^t \mathrel{+}= \rho_I^* \frac{\Delta V_I}{\Delta t} \tag{5.208}$$

$$b_I^t \mathrel{-}= \left(\rho_I^* U_I^* - \rho_I^n U_I^n\right) \frac{\Delta V_I}{\Delta t} \tag{5.209}$$

The convection term is computed at each face $k$ and assembled to the left (IL) and right (IR) control volumes.

$$A^c_{IL,J} \quad += \quad C^*_{k,J} \tag{5.210}$$
$$A^c_{IR,J} \quad -= \quad C^*_{k,J} \tag{5.211}$$

$$b^c_{IL} \quad -= \quad \sum_J C^*_{k,J} U^*_J \tag{5.212}$$
$$b^c_{IR} \quad += \quad \sum_J C^*_{k,J} U^*_J \tag{5.213}$$

The viscous stress term is computed at each face $k$ and assembled to the left (IL) and right (IR) control volumes. Only the solenoidal part of the stress term is used for the matrix.

$$F_{k,J} \quad = \quad -\left(\mu_k + \mu_{T,k}\right)\left( \left.\frac{\partial N_J}{\partial x}\right|_k A_x + \left.\frac{\partial N_J}{\partial y}\right|_k A_y + \left.\frac{\partial N_J}{\partial z}\right|_k A_z \right) \tag{5.214}$$
$$A^d_{IL,J} \quad += \quad F_{k,J} \tag{5.215}$$
$$A^d_{IR,J} \quad -= \quad F_{k,J} \tag{5.216}$$

$$\tau_{xx} \quad = \quad \left(\mu_k + \mu_{T,k}\right)\left(u^*_x + u^*_x\right) \tag{5.217}$$
$$\tau_{xy} \quad = \quad \left(\mu_k + \mu_{T,k}\right)\left(u^*_y + v^*_x\right) \tag{5.218}$$
$$\tau_{xz} \quad = \quad \left(\mu_k + \mu_{T,k}\right)\left(u^*_z + w^*_x\right) \tag{5.219}$$
$$f_k \quad = \quad -\left(\tau_{xx} A_x + \tau_{xy} A_y + \tau_{xz} A_z\right) \tag{5.220}$$
$$b^d_{IL} \quad -= \quad f_k \tag{5.221}$$
$$b^d_{IR} \quad += \quad f_k \tag{5.222}$$

The pressure is assembled in the form of a volume integral. The pressure gradients have been pre-computed at nodes use a surface-integral approximation.

$$b^s_I \quad -= \quad \left.\frac{\partial P}{\partial x}\right|^*_I \Delta V_I \tag{5.223}$$

## 5.5.11. Y-Momentum, 3D Turbulent Transport

The time term is lumped. The time-term contribution is evaluated for each sub-volume.

$$A^t_{I,I} \mathrel{+}= \rho^*_I \frac{\Delta V_I}{\Delta t} \tag{5.224}$$

$$b^t_I \mathrel{-}= \left(\rho^*_I V^*_I - \rho^n_I V^n_I\right) \frac{\Delta V_I}{\Delta t} \tag{5.225}$$

The convection term is computed at each face $k$ and assembled to the left (IL) and right (IR) control volumes.

$$A^c_{IL,J} \mathrel{+}= C^*_{k,J} \tag{5.226}$$
$$A^c_{IR,J} \mathrel{-}= C^*_{k,J} \tag{5.227}$$

$$b^c_{IL} \mathrel{-}= \sum_J C^*_{k,J} V^*_J \tag{5.228}$$
$$b^c_{IR} \mathrel{+}= \sum_J C^*_{k,J} V^*_J \tag{5.229}$$

The viscous stress term is computed at each face $k$ and assembled to the left (IL) and right (IR) control volumes. Only the solenoidal part of the stress term is used for the matrix.

$$F_{k,J} = -\left(\mu_k + \mu_{T,k}\right)\left(\left.\frac{\partial N_J}{\partial x}\right|_k A_x + \left.\frac{\partial N_J}{\partial y}\right|_k A_y + \left.\frac{\partial N_J}{\partial z}\right|_k A_z\right) \tag{5.230}$$
$$A^d_{IL,J} \mathrel{+}= F_{k,J} \tag{5.231}$$
$$A^d_{IR,J} \mathrel{-}= F_{k,J} \tag{5.232}$$

$$\tau_{yx} = \left(\mu_k + \mu_{T,k}\right)\left(v^*_x + u^*_y\right) \tag{5.233}$$
$$\tau_{yy} = \left(\mu_k + \mu_{T,k}\right)\left(v^*_y + v^*_y\right) \tag{5.234}$$
$$\tau_{yz} = \left(\mu_k + \mu_{T,k}\right)\left(v^*_z + w^*_y\right) \tag{5.235}$$
$$f_k = -\left(\tau_{yx} A_x + \tau_{yy} A_y + \tau_{yz} A_z\right) \tag{5.236}$$
$$b^d_{IL} \mathrel{-}= f_k \tag{5.237}$$
$$b^d_{IR} \mathrel{+}= f_k \tag{5.238}$$

The pressure is assembled in the form of a volume integral. The pressure gradients have been pre-computed at nodes use a surface-integral approximation.

$$b_I^s \; -= \; \left.\frac{\partial P}{\partial y}\right|_I^* \Delta V_I \tag{5.239}$$

## 5.5.12. Z-Momentum, 3D Turbulent Transport

The time term is lumped. The time-term contribution is evaluated for each sub-volume.

$$A_{I,I}^t \; += \; \rho_I^* \frac{\Delta V_I}{\Delta t} \tag{5.240}$$

$$b_I^t \; -= \; \left(\rho_I^* W_I^* - \rho_I^n W_I^n\right) \frac{\Delta V_I}{\Delta t} \tag{5.241}$$

The convection term is computed at each face $k$ and assembled to the left (IL) and right (IR) control volumes.

$$A_{IL,J}^c \; += \; C_{k,J}^* \tag{5.242}$$
$$A_{IR,J}^c \; -= \; C_{k,J}^* \tag{5.243}$$

$$b_{IL}^c \; -= \; \sum_J C_{k,J}^* W_J^* \tag{5.244}$$
$$b_{IR}^c \; += \; \sum_J C_{k,J}^* W_J^* \tag{5.245}$$

The viscous stress term is computed at each face $k$ and assembled to the left (IL) and right (IR) control volumes. Only the solenoidal part of the stress term is used for the matrix.

$$F_{k,J} \; = \; -\left(\mu_k + \mu_{T,k}\right)\left(\left.\frac{\partial N_J}{\partial x}\right|_k A_x + \left.\frac{\partial N_J}{\partial y}\right|_k A_y + \left.\frac{\partial N_J}{\partial z}\right|_k A_z\right) \tag{5.246}$$
$$A_{IL,J}^d \; += \; F_{k,J} \tag{5.247}$$
$$A_{IR,J}^d \; -= \; F_{k,J} \tag{5.248}$$

$$\tau_{zx} = (\mu_k + \mu_{T,k})\left(w_x^* + u_z^*\right) \tag{5.249}$$
$$\tau_{zy} = (\mu_k + \mu_{T,k})\left(w_y^* + v_z^*\right) \tag{5.250}$$
$$\tau_{zz} = (\mu_k + \mu_{T,k})\left(w_z^* + w_z^*\right) \tag{5.251}$$
$$f_k = -\left(\tau_{zx}A_x + \tau_{zy}A_y + \tau_{zz}A_z\right) \tag{5.252}$$
$$b_{IL}^d -= f_k \tag{5.253}$$
$$b_{IR}^d += f_k \tag{5.254}$$

The pressure is assembled in the form of a volume integral. The pressure gradients have been pre-computed at nodes use a surface-integral approximation.

$$b_I^s -= \left.\frac{\partial P}{\partial z}\right|_I^* \Delta V_I \tag{5.255}$$

## 5.5.13. Turbulent Kinetic Energy, 3D Turbulent Transport

The time term is lumped. The time-term contribution is evaluated for each sub-volume.

$$A_{I,I}^t += \rho_I^* \frac{\Delta V_I}{\Delta t} \tag{5.256}$$

$$b_I^t -= \left(\rho_I^* K_I^* - \rho_I^n K_I^n\right)\frac{\Delta V_I}{\Delta t} \tag{5.257}$$

The convection term is computed at each face $k$ and assembled to the left (IL) and right (IR) control volumes.

$$A_{IL,J}^c += C_{k,J}^{n+1} \tag{5.258}$$
$$A_{IR,J}^c -= C_{k,J}^{n+1} \tag{5.259}$$

$$b_{IL}^c -= \sum_J C_{k,J}^{n+1} K_J^* \tag{5.260}$$
$$b_{IR}^c += \sum_J C_{k,J}^{n+1} K_J^* \tag{5.261}$$

The viscous stress term is computed at each face $k$ and assembled to the left (IL) and right (IR) control volumes. Only the solenoidal part of the stress term is used for the matrix.

$$F_{k,J} = -\frac{\mu_{T,k}}{\sigma_k}\left(\left.\frac{\partial N_J}{\partial x}\right|_k A_x + \left.\frac{\partial N_J}{\partial y}\right|_k A_y + \left.\frac{\partial N_J}{\partial z}\right|_k A_z\right) \tag{5.262}$$

$$A^d_{IL,J} \mathrel{+}= F_{k,J} \tag{5.263}$$

$$A^d_{IR,J} \mathrel{-}= F_{k,J} \tag{5.264}$$

$$f_k = -\frac{\mu_{T,k}}{\sigma_k}\left(k^*_x A_x + k^*_y A_y + k^*_z A_z\right) \tag{5.265}$$

$$b^d_{IL} \mathrel{-}= f_k \tag{5.266}$$

$$b^d_{IR} \mathrel{+}= f_k \tag{5.267}$$

The turbulence production is assembled in the form of a volume integral. The velocity derivatives are computed at the sub-volume centroids.

$$\begin{aligned}\Phi &= 2\left(u_x^2 + v_y^2 + w_z^2\right) - \frac{2}{3}\left(u_x + v_y + w_z\right)^2 \\ &+ \left(u_y + v_x\right)^2 + \left(v_z + w_y\right)^2 + \left(w_x + u_z\right)^2 \end{aligned} \tag{5.268}$$

$$b^s_I \mathrel{+}= \mu_T \Phi \Delta V_I \tag{5.269}$$

The turbulence dissipation is assembled in the form of a volume integral. The terms are evaluated at the node associated with the control volume.

$$A^s_{I,I} \mathrel{+}= \rho_I \frac{E^*_I}{K^*_I}\Delta V_I \tag{5.270}$$

$$b^s_I \mathrel{-}= \rho_I E^*_I \Delta V_I \tag{5.271}$$

## 5.5.14. Turbulence Dissipation, 3D Turbulent Transport

The time term is lumped. The time-term contribution is evaluated for each sub-volume.

$$A^t_{I,I} \mathrel{+}= \rho^*_I \frac{\Delta V_I}{\Delta t} \tag{5.272}$$

$$b^t_I \mathrel{-}= \left(\rho^*_I E^*_I - \rho^n_I E^n_I\right)\frac{\Delta V_I}{\Delta t} \tag{5.273}$$

The convection term is computed at each face $k$ and assembled to the left (IL) and right (IR) control volumes.

$$A^c_{IL,J} \;+=\; C^{n+1}_{k,J} \tag{5.274}$$
$$A^c_{IR,J} \;-=\; C^{n+1}_{k,J} \tag{5.275}$$

$$b^c_{IL} \;-=\; \sum_J C^{n+1}_{k,J} E^*_J \tag{5.276}$$
$$b^c_{IR} \;+=\; \sum_J C^{n+1}_{k,J} E^*_J \tag{5.277}$$

The viscous stress term is computed at each face $k$ and assembled to the left (IL) and right (IR) control volumes. Only the solenoidal part of the stress term is used for the matrix. As with the turbulent kinetic energy transport equation, the molecular viscosity may augment the effective diffusivity.

$$F_{k,J} \;=\; -\frac{\mu_{T,k}}{\sigma_\epsilon}\left(\left.\frac{\partial N_J}{\partial x}\right|_k A_x + \left.\frac{\partial N_J}{\partial y}\right|_k A_y + \left.\frac{\partial N_J}{\partial z}\right|_k A_z\right) \tag{5.278}$$
$$A^d_{IL,J} \;+=\; F_{k,J} \tag{5.279}$$
$$A^d_{IR,J} \;-=\; F_{k,J} \tag{5.280}$$

$$f_k \;=\; -\frac{\mu_{T,k}}{\sigma_\epsilon}\left(\epsilon^*_x A_x + \epsilon^*_y A_y + \epsilon^*_z A_z\right) \tag{5.281}$$
$$b^d_{IL} \;-=\; f_k \tag{5.282}$$
$$b^d_{IR} \;+=\; f_k \tag{5.283}$$

The velocity derivatives are computed at the sub-volume centroids using velocities at the new time level $(n+1)$.

$$\begin{aligned}\Phi \;&=\; 2\left(u_x^2 + v_y^2 + w_z^2\right) - \frac{2}{3}\left(u_x + v_y + w_z\right)^2 \\ &+\; \left(u_y + v_x\right)^2 + \left(v_z + w_y\right)^2 + \left(w_x + u_z\right)^2 \end{aligned} \tag{5.284}$$
$$b^s_I \;+=\; \mu_T C_{\epsilon_1} \Phi \frac{E^*_I}{K^*_I} \Delta V_I \tag{5.285}$$

The turbulence dissipation is assembled in the form of a volume integral. The terms are evaluated at the node associated with the control volume.

$$A^s_{I,I} \;+=\; \rho_I C_{\epsilon_2} \frac{E^*_I}{K^*_I} \Delta V_I \tag{5.286}$$

$$b^s_I \;-=\; \rho_I C_{\epsilon_2} \frac{E^*_I}{K^*_I} E^*_I \Delta V_I \tag{5.287}$$

### 5.5.15. Energy, 3D Turbulent Transport

The time term is lumped. The time-term contribution is evaluated for each sub-volume.

$$A^t_{I,I} \mathrel{+}= \rho^*_I \frac{\Delta V_I}{\Delta t} \tag{5.288}$$

$$b^t_I \mathrel{-}= \left(\rho^*_I H^*_I - \rho^n_I H^n_I\right) \frac{\Delta V_I}{\Delta t} \tag{5.289}$$

The convection term is computed at each face $k$ and assembled to the left (IL) and right (IR) control volumes.

$$A^c_{IL,J} \mathrel{+}= C^{n+1}_{k,J} \tag{5.290}$$
$$A^c_{IR,J} \mathrel{-}= C^{n+1}_{k,J} \tag{5.291}$$

$$b^c_{IL} \mathrel{-}= \sum_J C^{n+1}_{k,J} H^*_J \tag{5.292}$$
$$b^c_{IR} \mathrel{+}= \sum_J C^{n+1}_{k,J} H^*_J \tag{5.293}$$

The heat conduction term is computed at each face $k$ and assembled to the left (IL) and right (IR) control volumes.

$$F_{k,J} = -\left(\frac{\mu_k}{\mathrm{Pr}} + \frac{\mu_{T,k}}{\mathrm{Pr}_T}\right)\left(\left.\frac{\partial N_J}{\partial x}\right|_k A_x + \left.\frac{\partial N_J}{\partial y}\right|_k A_y + \left.\frac{\partial N_J}{\partial z}\right|_k A_z\right) \tag{5.294}$$
$$A^d_{IL,J} \mathrel{+}= F_{k,J} \tag{5.295}$$
$$A^d_{IR,J} \mathrel{-}= F_{k,J} \tag{5.296}$$

$$q_k = -\left(\frac{\mu_k}{\mathrm{Pr}} + \frac{\mu_{T,k}}{\mathrm{Pr}_T}\right)\left(h^*_x A_x + h^*_y A_y + h^*_z A_z\right) \tag{5.297}$$
$$b^d_{IL} \mathrel{-}= q_k \tag{5.298}$$
$$b^d_{IR} \mathrel{+}= q_k \tag{5.299}$$

## 5.5.16. Species, 3D Turbulent Transport

There is a species equations for each species. The mass fraction is $Y_s$, where $s$ is the species number. The time term is lumped. The time-term contribution is evaluated for each sub-volume.

$$A^t_{I,I} \mathrel{+}= \rho^*_I \frac{\Delta V_I}{\Delta t} \tag{5.300}$$

$$b^t_I \mathrel{-}= \left(\rho^*_I Y^*_{s,I} - \rho^n_I Y^n_{s,I}\right) \frac{\Delta V_I}{\Delta t} \tag{5.301}$$

The convection term is computed at each face $k$ and assembled to the left (IL) and right (IR) control volumes.

$$A^c_{IL,J} \mathrel{+}= C^{n+1}_{k,J} \tag{5.302}$$
$$A^c_{IR,J} \mathrel{-}= C^{n+1}_{k,J} \tag{5.303}$$

$$b^c_{IL} \mathrel{-}= \sum_J C^{n+1}_{k,J} Y^*_{s,J} \tag{5.304}$$
$$b^c_{IR} \mathrel{+}= \sum_J C^{n+1}_{k,J} Y^*_{s,J} \tag{5.305}$$

The mass diffusion term is computed at each face $k$ and assembled to the left (IL) and right (IR) control volumes.

$$F_{k,J} = -\left(\frac{\mu_k}{\mathrm{Sc}} + \frac{\mu_{T,k}}{\mathrm{Sc}_T}\right)\left(\left.\frac{\partial N_J}{\partial x}\right|_k A_x + \left.\frac{\partial N_J}{\partial y}\right|_k A_y + \left.\frac{\partial N_J}{\partial z}\right|_k A_z\right) \tag{5.306}$$
$$A^d_{IL,J} \mathrel{+}= F_{k,J} \tag{5.307}$$
$$A^d_{IR,J} \mathrel{-}= F_{k,J} \tag{5.308}$$

$$f_k = -\left(\frac{\mu_k}{\mathrm{Sc}} + \frac{\mu_{T,k}}{\mathrm{Sc}_T}\right)\left(ys^*_x A_x + ys^*_y A_y + ys^*_z A_z\right) \tag{5.309}$$
$$b^d_{IL} \mathrel{-}= f_k \tag{5.310}$$
$$b^d_{IR} \mathrel{+}= f_k \tag{5.311}$$

The chemical production source terms from the EDC model are applied at the centroid of the control volume. The production term is constructed from the rate, the fine structure mass fractions, and the average mass fractions.

$$A^s_{I,I} \mathrel{+}= \dot{r}_{s,I} \Delta V_I \tag{5.312}$$

$$\begin{aligned} \dot{\omega}_{s,I} &= \dot{r}_{s,I} \left( Y^{fs}_{s,I} - Y_{s,I} \right) && (5.313) \\ b^s_I &\mathrel{+}= \dot{\omega}_{s,I} \Delta V_I && (5.314) \end{aligned}$$

## 5.5.17. Soot Transport, 3D Turbulent Transport

The time term is lumped. The time-term contribution is evaluated for each sub-volume.

$$A^t_{I,I} \mathrel{+}= \rho^*_I \frac{\Delta V_I}{\Delta t} \tag{5.315}$$

$$b^t_I \mathrel{-}= \left( \rho^*_I S^*_I - \rho^n_I S^n_I \right) \frac{\Delta V_I}{\Delta t} \tag{5.316}$$

The convection term is computed at each face $k$ and assembled to the left (IL) and right (IR) control volumes.

$$\begin{aligned} A^c_{IL,J} &\mathrel{+}= C^{n+1}_{k,J} && (5.317) \\ A^c_{IR,J} &\mathrel{-}= C^{n+1}_{k,J} && (5.318) \end{aligned}$$

$$\begin{aligned} b^c_{IL} &\mathrel{-}= \sum_J C^{n+1}_{k,J} S^*_J && (5.319) \\ b^c_{IR} &\mathrel{+}= \sum_J C^{n+1}_{k,J} S^*_J && (5.320) \end{aligned}$$

The diffusion term is computed at each face $k$ and assembled to the left (IL) and right (IR) control volumes.

$$\begin{aligned} F_{k,J} &= -\left( \frac{\mu_k}{\mathrm{Sc}} + \frac{\mu_{T,k}}{\mathrm{Sc}_T} \right) \left( \left. \frac{\partial N_J}{\partial x} \right|_k A_x + \left. \frac{\partial N_J}{\partial y} \right|_k A_y + \left. \frac{\partial N_J}{\partial z} \right|_k A_z \right) && (5.321) \\ A^d_{IL,J} &\mathrel{+}= F_{k,J} && (5.322) \\ A^d_{IR,J} &\mathrel{-}= F_{k,J} && (5.323) \end{aligned}$$

$$f_k = -\left(\frac{\mu_k}{\mathrm{Sc}} + \frac{\mu_{T,k}}{\mathrm{Sc}_T}\right)\left(s_x^* A_x + s_y^* A_y + s_z^* A_z\right) \tag{5.324}$$

$$b_{IL}^d -= f_k \tag{5.325}$$

$$b_{IR}^d += f_k \tag{5.326}$$

The soot production source term from the EDC model is applied at the centroid of the control volume.

$$b_I^s += \dot{\omega}_{soot,I} \Delta V_I \tag{5.327}$$

## 5.5.18. Soot Nuclei Transport, 3D Turbulent Transport

The time term is lumped. The time-term contribution is evaluated for each sub-volume.

$$A_{I,I}^t += \rho_I^* \frac{\Delta V_I}{\Delta t} \tag{5.328}$$

$$b_I^t -= \left(\rho_I^* N_I^* - \rho_I^n N_I^n\right) \frac{\Delta V_I}{\Delta t} \tag{5.329}$$

The convection term is computed at each face $k$ and assembled to the left (IL) and right (IR) control volumes.

$$A_{IL,J}^c += C_{k,J}^{n+1} \tag{5.330}$$

$$A_{IR,J}^c -= C_{k,J}^{n+1} \tag{5.331}$$

$$b_{IL}^c -= \sum_J C_{k,J}^{n+1} N_J^* \tag{5.332}$$

$$b_{IR}^c += \sum_J C_{k,J}^{n+1} N_J^* \tag{5.333}$$

The diffusion term is computed at each face $k$ and assembled to the left (IL) and right (IR) control volumes.

$$F_{k,J} = -\left(\frac{\mu_k}{\mathrm{Sc}} + \frac{\mu_{T,k}}{\mathrm{Sc}_T}\right)\left(\left.\frac{\partial N_J}{\partial x}\right|_k A_x + \left.\frac{\partial N_J}{\partial y}\right|_k A_y + \left.\frac{\partial N_J}{\partial z}\right|_k A_z\right) \tag{5.334}$$

$$A^d_{IL,J} \mathrel{+}= F_{k,J} \tag{5.335}$$

$$A^d_{IR,J} \mathrel{-}= F_{k,J} \tag{5.336}$$

$$f_k = -\left(\frac{\mu_k}{\mathrm{Sc}} + \frac{\mu_{T,k}}{\mathrm{Sc}_T}\right)\left(n^*_x A_x + n^*_y A_y + n^*_z A_z\right) \tag{5.337}$$

$$b^d_{IL} \mathrel{-}= f_k \tag{5.338}$$

$$b^d_{IR} \mathrel{+}= f_k \tag{5.339}$$

The soot nuclei production source term from the EDC model is applied at the centroid of the control volume.

$$b^s_I \mathrel{+}= \dot{\omega}_{nucl,I}\Delta V_I \tag{5.340}$$

# 5.6. DISCRETE BOUNDARY CONDITIONS

The Dirichlet boundary conditions are applied directly in the linear solver. The flux boundary conditions are linearized and then assembled to the linear system. The flux boundary conditions are processed on a face-by-face basis. The data available with each face includes all the data on the parent element.

## 5.6.1. Symmetry, 3D Momentum

The viscous stresses can only impart a normal force at a symmetry boundary. The only other force contribution is from the pressure. The pressure is integrated over the boundary using the boundary nodal values.

The normal viscous force component is assembled to the right hand side only for the laminar equations.

The viscous stress and sub-face normal are computed at each sub-face on the element face. The integrated sub-face force is assembled to its adjacent node.

$$F_{wi} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) n_j A_w \tag{5.341}$$

where $n_j$ is the unit sub-face normal vector and $A_w$ is the area of the sub-face.

### 5.6.2. Outflow, 3D Mass

The mass flux at a pressure-specified outflow boundary is given by Equation 5.92. The pressure at the face in the equation is $P_{fc}$ and is the specified value (see Figure 5.3-3). The interior sub-face pressure is $P_{ss}$ and is an average of nodal pressures. The fully assembled Poisson equation for pressure will have positive diagonal coefficients. Note that the form of Equation 5.92 will contribute a positive diagonal value. The nodal pressure gradient, $G_{ij}p_j^{ast}$, contains the influence of the specified pressure. The difference of the nodal pressure gradient and the boundary pressure gradient cancels the influence of the specified pressure in the outflow boundary condition. The specified pressure at the boundary only directly influences the momentum balance.

### 5.6.3. Outflow, 3D Momentum

The outflow boundary condition is applied to boundaries with either pressure-specified inflow or pressure-specified outflow. The viscous stresses are integrated over the boundary, but the viscous force normal to the boundary is neglected.

If the flow is entering the domain, the convected velocity is a combination of a specified tangential velocity (coflow) and a normal velocity. The normal velocity is constructed from the local nodal values.

If the flow exits the domain, the convected velocity values are interpolated from nodal velocities in the element adjacent to the boundary, similar to the interior scheme discussed in Section 5.3.1. The convected velocities are blended from an upwind interpolation (nearest boundary node) and centered interpolation. The shape functions for the centered interpolation are taken from the interior sub-face that is directly opposite the boundary sub-face. The upwind scheme will extrapolate from the nearest node and the linear profile skew upwind scheme will interpolate to the boundary sub-face centroid.

### 5.6.4. Outflow, 3D Energy and Temperature

The outflow boundary condition is applied to boundaries with either pressure-specified inflow or pressure-specified outflow. The heat conduction is integrated over the boundary. The transport of enthalpy by mass diffusion for a multicomponent system is not yet implemented (cdm – 9/26/10).

If the flow is entering the domain, the convected enthalpy is set to a far-field reference value.

The convected enthalpy values are interpolated from nodal enthalpies in the element adjacent to the boundary, similar to the interior scheme discussed in Section 5.3.1. The convected enthalpies are blended from an upwind interpolation (nearest boundary node) and centered interpolation. The shape functions for the centered interpolation are taken from the interior sub-face that is directly opposite the boundary sub-face. The upwind scheme will extrapolate from the nearest node and the linear profile skew upwind scheme will interpolate to the boundary sub-face centroid.

### 5.6.5. Outflow, 3D Species and Soot

The outflow boundary condition is applied to boundaries with either pressure-specified inflow or pressure-specified outflow. The mass diffusion is integrated over the boundary.

If the flow is entering the domain, the convected mass fractions are set to far-field reference values.

The convected species mass fraction values are interpolated from nodal mass fractions in the element adjacent to the boundary, similar to the interior scheme discussed in Section 5.3.1. The convected mass fractions are blended from an upwind interpolation (nearest boundary node) and centered interpolation. The shape functions for the centered interpolation are taken from the interior sub-face that is directly opposite the boundary sub-face. The upwind scheme will extrapolate from the nearest node and the linear profile skew upwind scheme will interpolate to the boundary sub-face centroid.

### 5.6.6. Outflow, 3D Turbulent Kinetic Energy

The outflow boundary condition is applied to boundaries with either pressure-specified inflow or pressure-specified outflow. If the flow is entering the domain, the convected turbulent kinetic energy is set by one of two ways:

- user specified value for turbulent kinetic energy, e.g. 0.0.,
- calculated entrainment value based on user specified turbulence intensity, $T_{in}$, and the relationship

$$k_{ip} = \frac{3}{2} \left( U_{ref} T_{in} \right)^2 . \tag{5.342}$$

The reference velocity at the integration point, $U_{ref}$, is determined by the current integration point mass flow rate divided by a characteristic area divided by the integration point density.

The convected turbulent kinetic energy is blended from an upwind interpolation (nearest boundary node) and centered interpolation. The shape functions for the centered interpolation are taken from the interior sub-face that is directly opposite the boundary sub-face. The upwind scheme will extrapolate from the nearest node and the linear profile skew upwind scheme will interpolate to the boundary sub-face centroid.

### 5.6.7. Outflow, 3D Turbulence Dissipation

The outflow boundary condition is applied to boundaries with either pressure-specified inflow or pressure-specified outflow. If the flow is entering the domain, the convected turbulence dissipation rate is set by one of two ways:

- user specified value for turbulent dissipation rate, e.g. 0.0.,
- calculated entrainment value based on user specified turbulence intensity, characteristic length and the relationship

$$\epsilon_{ip} = C_\mu{}^{3/4} \frac{k_{ip}^{3/2}}{l}, \tag{5.343}$$

where $l = 0.07L$; $L$ represents the user-specified characteristic length of large turbulent structures. The integration point turbulent kinetic energy is again based on the user specified turbulence intensity in conjunction with Equation 5.342.

The convected turbulent dissipation rate is blended from an upwind interpolation (nearest boundary node) and centered interpolation. The shape functions for the centered interpolation are taken from the interior sub-face that is directly opposite the boundary sub-face. The upwind scheme will extrapolate from the nearest node and the linear profile skew upwind scheme will interpolate to the boundary sub-face centroid.

## 5.6.8. Wall, 3D Turbulent Momentum

The effect of the wall force imparted by the wall on the fluid, as outlined in Section 2.7.5, is handled by the standard law of the wall formulation. To explain this procedure, consider a two dimensional element with two faces that consist of a wall boundary side set, Figure 5.6-1.

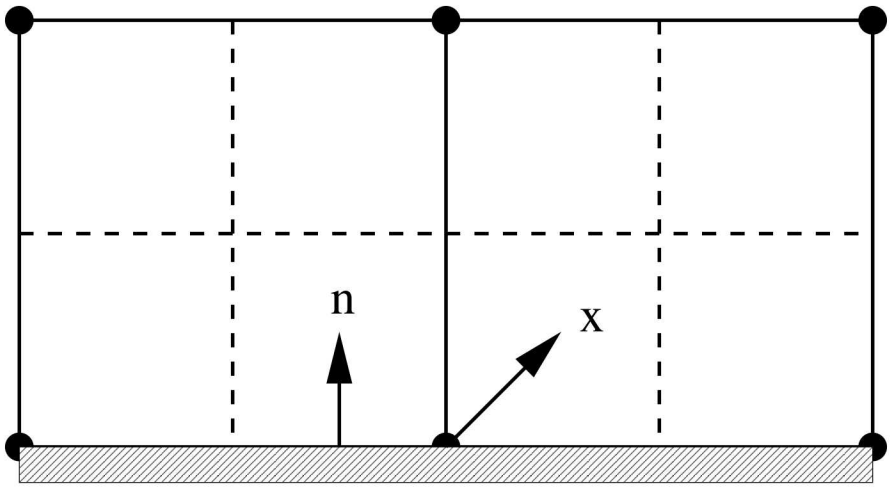


**Figure 5.6-1.. Integration locations for a wall boundary.**

The resulting discretization of the $i^{th}$-component of velocity, for the boundary face that is a wall can be expressed as follows,

$$F_{wi} = -\int \tau_{ij} n_j dS = \lambda_w A_w u_{i\|}, \tag{5.344}$$

where $A_w$ is the area, $n_j$ is the unit normal to the wall, and $\lambda_w$ is the wall shear stress factor from law of the wall,

$$\lambda_w = \frac{\rho \kappa u_\tau}{ln\left(E y^+\right)}. \tag{5.345}$$

The parallel velocity component in Equation 5.344 is determined by the projection of the nodal velocity onto each of the four (hex) or three (tet) subcontrol boundary faces (see Equation 2.182). In many respects, this procedure is very much like that of a cell-centered scheme in that the nodal velocity is assumed to act over all boundary faces. The paramount difference is the ability of one nodal velocity to be applied to a multitude of faces of potentially different orientation.

As indicated in Section 2.7.5, the friction velocity at the centroid of the boundary face is determined by a nonlinear solution procedure that will now be described. The procedure begins by use of Equation 2.171, rearranged to form the function $F$,

$$F\left(u_\tau\right) = u_{\|} - \frac{u_\tau}{\kappa} ln\left(\frac{E \rho Y_p u_\tau}{\mu}\right). \tag{5.346}$$

The objective is to determine the value of the friction velocity such that the function, $F$, is minimized. A Newton solve is therefore constructed that has the following standard iteration form,

$$u_\tau^{k+1} = u_\tau^k - \frac{F^k}{F'^k}, \tag{5.347}$$

where $F^k$ is defined by Equation 5.346 evaluated at the $k^{th}$ iteration level, and $F'^k$ is defined by

$$F'^k = -\frac{1}{\kappa}\left[1 + ln\left(\frac{E \rho Y_p {u_\tau}^k}{\mu}\right)\right]. \tag{5.348}$$

The procedure by which the normal distance to the wall is determined is based on the method outlined by the vertex-centered CFD code TASCflow [154]. In the procedure, the normal distance to the wall is linked to the grid by the evaluation of the normal distance from the subcontrol volume center to the boundary face. Therefore, the normal distance to the wall can be determined by the following steps:

• Determination of the coordinates of the subcontrol volume center by a shape function loop over all nodes. This step in the procedure mandates a SIERRA heterogeneous (face-element) workset algorithm.

• The determination of a vector, $\boldsymbol{x_i}$, from the subcontrol volume center to the respective nodal location.

• The use of the perpendicular projection operator, $P_\perp$, which is defined by,

$$P_\perp = -n_i n_j, \tag{5.349}$$

and finally,

• The determination of the normal distance by

$$Y_p = \sqrt{{x_{\perp,1}}^2 + {x_{\perp,2}}^2 + {x_{\perp,3}}^2}. \tag{5.350}$$

For convenience, the density and viscosity used in all of the above equations are nodal quantities. In other words, the physical properties are not interpolated to the centroid of the boundary face.

Once the wall shear stress factor is evaluated, it is required that the appropriate component of the velocity parallel to the boundary face is used appropriately within the respective momentum equations. As was discussed in the section on non-orthogonal momentum math models, Section 2.7.5, the parallel velocity can be written in component form (see Equation 2.183).

#### 5.6.8.1. X-Momentum

The x-momentum wall force, $F_{w1}$ is expressed as

$$F_{w1} = -\lambda_w A_w u_{1\parallel}, \tag{5.351}$$

where $u_{1\parallel}$ is defined as

$$u_{1\parallel} = \left(1 - n_1^2\right) u_{1,nd} - \left(1 - n_1 n_2\right) u_{2,nd} - \left(1 - n_1 n_2\right) u_{3,nd}. \tag{5.352}$$

Note that the form of Equation 5.352 allows for an implicit treatment of the force imparted by the wall on the fluid by the factor

$$\lambda_w A_w \left(1 - n_1^2\right). \tag{5.353}$$

#### 5.6.8.2. Y-Momentum

The y-momentum wall force, $F_{w2}$ is expressed as

$$F_{w2} = -\lambda_w A_w u_{2\parallel}, \tag{5.354}$$

where $u_{2\parallel}$ is defined as

$$u_{2\parallel} = \left(1 - n_2^2\right) u_{2,nd} - \left(1 - n_2 n_1\right) u_{1,nd} - \left(1 - n_2 n_3\right) u_{3,nd}. \tag{5.355}$$

Note that the form of Equation 5.355 allows for an implicit treatment of the force imparted by the wall on the fluid by the factor

$$\lambda_w A_w \left(1 - {n_2}^2\right). \tag{5.356}$$

#### 5.6.8.3. Z-Momentum

The z-momentum wall force, $F_{w3}$ is expressed as

$$F_{w3} = -\lambda_w A_w u_{3\|}, \tag{5.357}$$

where $u_{3\|}$ is defined as

$$u_{3\|} = \left(1 - n_3^2\right) u_{3,nd} - \left(1 - n_3 n_1\right) u_{1,nd} - \left(1 - n_3 n_2\right) u_{2,nd}. \tag{5.358}$$

Note that the form of Equation 5.358 allows for an implicit treatment of the force imparted by the wall on the fluid by the factor

$$\lambda_w A_w \left(1 - n_3^2\right). \tag{5.359}$$

### 5.6.9. Wall, 3D Turbulent Kinetic Energy

As described in Section 2.7.6, the wall boundary condition for turbulent kinetic energy can be applied in a variety of ways. In general, there are two supported methods.

The first method is specify the near-wall turbulent kinetic energy as a Dirichlet condition whose value is determined by the assumption of local equilibrium between production and dissipation of turbulent kinetic energy.

The second method is to solve a transport equation for the near wall turbulent kinetic energy whose form utilizes a modified production and dissipation term based on the assumption of local equilibrium between production and dissipation of turbulent kinetic energy. The use of a full control volume equation for the near wall turbulent kinetic energy in the presence of non-zero convection and diffusion coefficients is a violation of the very tenants of the law of the wall formulation which implicitly assumes pure shear flow behavior. Nevertheless, this method is frequently used.

The Dirichlet method consists of the determination of each integration point turbulent kinetic energy by use of the following equation,

$$k_{ip} = \frac{u_\tau^2}{C_\mu^{1/2}}, \tag{5.360}$$

The value of $u_\tau$ is determined by a nonlinear iteration solve of the law of the wall formulation. The integration point values are area weighted and assembled into the nodal location. The nodal value of the turbulent kinetic energy is given by the accumulated area weighed integration point turbulent kinetic energy divided by the total face area.

### 5.6.10. Wall, 3D Turbulence Dissipation

Consistent with all of literature, the near-wall value of turbulent dissipation is determined from iteration-lagged values of friction velocity,

$$\epsilon_{ip}^{n+1} = \frac{{u_\tau}^3}{\kappa Y_p}. \tag{5.361}$$

As with the implementation of the turbulent kinetic energy, the value computed in Equation 5.361 is area weighted and assembled to the nodal location. The Dirichlet condition is determined by the assembled quantity divided by the entire area of the boundary faces that are "owned" by the node.

## 5.7. CONJUGATE HEAT TRANSFER

### 5.7.1. General Formulation

A conjugate heat transfer problem is one in which conductive heat transfer in a solid region is coupled to the convective heat transfer in a neighboring fluid. In its most general form, the coupling at the boundary is governed by the conservation of energy, such that heat flux out of the solid is equal to heat flux into the fluid:

$$\mathbf{q}_s \cdot \mathbf{n} = \mathbf{q}_f \cdot \mathbf{n} \tag{5.362}$$

where $\mathbf{q}_s$ and $\mathbf{q}_f$ are the heat flux in the solid and fluid, respectively, and $\mathbf{n}$ is the surface normal directed into the solid and out of the fluid.

The exact form in which equation (5.362) is implemented depends on whether the fluid flow is laminar or turbulent, since different expressions must be used in these cases for $\mathbf{q}_f$. The heat flux in the solid is always due to conduction alone, but there are several possible choices that could be made for the discretization of this flux in space and time.

### 5.7.2. Time Integration

In Fuego, conjugate heat transfer is implemented through loose coupling between the fluid and solid regions, meaning that at each time step, each region is solved separately by treating information from the neighboring region as "given", and no extra iterations are done between regions to ensure convergence at a single time step. The specific algorithm used can be described as a temperature-forward, flux-back scheme. At a given time step $n$, the fluid equations are solved using the current solid temperature as a Dirichlet boundary condition; the temperature field of the fluid is thus updated to state $n+1$ everywhere except on the boundary. Then, the heat flux in the fluid at step $n+1$ is computed and transferred to the solid. Finally, the solid region is solved, updating to state $n+1$ using the information from the fluid as a flux boundary condition.

Rather than applying the fluid heat flux directly to the solid, we choose to write the solid boundary condition in the form of a convective heat flux boundary condition:

$$\mathbf{q}_s(\mathbf{x}) \cdot \mathbf{n} = h(\mathbf{x}) \left( T_\infty(\mathbf{x}) - T_s(\mathbf{x}) \right) \tag{5.363}$$

where $h$ is a convection coefficient, $T_\infty$ is the fluid temperature away from the wall, and $T_s$ is the solid surface temperature. Both $h$ and $T_\infty$ are computed from the fluid temperature field in a way that will be specified, while $T_s$ is left free in the solution of the solid region temperature. This formulation can be shown to be more stable than the alternative of simply transferring the heat flux in the fluid and applying it as a pure Neumann boundary condition to the solid.

Using superscripts to denote time step, the loosely coupled integration scheme can thus be written as:

$$T_f^{n+1} = T_s^n \qquad \text{on } \Gamma_{fs} \tag{5.364a}$$

$$\mathbf{q}_s^{n+1}(\mathbf{x}) \cdot \mathbf{n} = h^{n+1}(\mathbf{x}) \left( T_\infty^{n+1}(\mathbf{x}) - T_s^{n+1}(\mathbf{x}) \right) \qquad \text{on } \Gamma_{fs} \tag{5.364b}$$

where $\Gamma_{fs}$ is the fluid-solid interface.

### 5.7.3. Discretization of Conduction Region Boundary Condition

The quantity that is needed for a flux boundary condition condition in our CVFEM formulation is the heat flux integrated over the interface surface area associated with each node on the surface. Equation (5.364b) is applied to the conduction region at each surface node by assuming that $h$, $T_s$ and $T_\infty$ can be treated as constants on that node's sub-control surfaces:

$$Q_{s,I}^{n+1} = \int_{SCS_I} \mathbf{q}_s \cdot \mathbf{n} dA = h_I^{n+1} A_I \left( T_{\infty,I}^{n+1} - T_{s,I}^{n+1} \right) \tag{5.365}$$

where $A_I$ is the surface area associated with node $I$. The nodal data $h_I^{n+1}$ and $T_{\infty,I}^{n+1}$ are computed from the fluid solution at time step $n+1$ (see section 5.7.4), while $T_{s,I}^{n+1}$ is a degree of freedom solved during the conduction region solution.

### 5.7.4. Computation of Convection Temperature and Coefficient

On the fluid side, the total heat transfer associated with a node on the fluid-solid interface is the integral of the heat flux over that node's sub-control surfaces on the interface:

$$Q_{f,I}^{n+1} = \int_{SCS_I} \mathbf{q}_f^{n+1} \cdot \mathbf{n} dA. \tag{5.366}$$

Consider the case in which fluid and solid surfaces meshes conform exactly at the interface. Then, every fluid node can be associated with a corresponding solid node, and using Equations (5.362) and (5.365) we have:

$$Q_{f,I}^{n+1} = Q_{s,I}^{n+1} \tag{5.367a}$$

$$= h_I^{n+1} A_I \left( T_{\infty,I}^{n+1} - T_{s,I}^{n+1} \right) \tag{5.367b}$$

$$\approx h_I^{n+1} A_I \left( T_{\infty,I}^{n+1} - T_{f,I}^{n+1} \right) \tag{5.367c}$$

where the last line (where $T_{f,I}^{n+1}$ is substituted for $T_{s,I}^{n+1}$) follows approximately from (5.364a); this approximation is of the same order accuracy as the time integration scheme, and for steady state it is exact. In cases in which the surface meshes do not conform exactly, the nodal values of $h_I$ and $T_{\infty,I}$ are passed through an interpolation transfer, introducing a small amount of error.

The total heat transfer $Q_{f,I}$ must be decomposed into two components: $Q_{W,I}$ representing the variables of the fluid at nodes on the surface ("wall"), and $Q_{\infty,I}$ representing variables at nodes away from the surface:

$$Q_{f,I} = Q_{W,I} + Q_{\infty,I} \tag{5.368}$$

The way in which this decomposition is done depends on whether the flow is laminar or turbulent, as will be discussed. Comparing this decomposition with (5.367c), it is clear that:

$$Q_{W,I} = -h_I A_I T_{f,I} \tag{5.369a}$$
$$Q_{\infty,I} = h_I A_I T_{\infty,I} \tag{5.369b}$$

Rearranging:

$$h_I = -\frac{Q_{W,I}}{T_{f,I} A_I} \tag{5.370a}$$
$$T_{\infty,I} = \frac{Q_{\infty,I}}{h_I A_I} \tag{5.370b}$$

Finally, we must define the decomposition of $Q_{f,I}$ for laminar and turbulent flow. It is possible when using this approach to end up with negative values for $h_I$, which appear non-physical to the analyst and are detrimental to the numerical stability of the conduction solve since they reduce diagonal dominance of the linear system. Since the choice of these parameters is arbitrary as long as they reproduce the correct energy flux, when this occurs we reverse the sign of $h_I$ and re-compute $T_{\infty,I}$ as

$$T_{\infty,I} = \frac{Q_{f,I}}{h_I} + T_{f,I} \tag{5.371}$$

#### 5.7.4.1. Resolution of Boundary Layer

The fluid velocity at the solid surface is zero for laminar flows and turbulent flow models in which the boundary layer is resolved, so all heat transfer in the fluid near walls is due to conduction:

$$\mathbf{q}_f(\mathbf{x}) = -\kappa_f(\mathbf{x}) \nabla T(\mathbf{x}) \tag{5.372}$$

where $\kappa_f$ is the thermal conductivity of the fluid. Substituting this into (5.366) and using the finite element interpolation for $T(\mathbf{x})$ gives:

$$Q_{W,I} = -\int_{SCS_I} \kappa_f \sum_J (\mathbf{n} \cdot \nabla N_J) T_J dA \tag{5.373}$$

where $N_J$ and $T_J$ are respectively the FEM shape function and temperature degree of freedom associated with node $J$.

The most obvious way of decomposing $Q_{f,I}$ is by breaking the summation into two summations, one over boundary nodes, one over off-boundary nodes:

$$Q_{f,I} = -\int_{SCS_I} \kappa_f \sum_{J \in B} (\mathbf{n} \cdot \nabla N_J) T_J dA \tag{5.374a}$$

$$Q_{\infty,I} = -\int_{SCS_I} \kappa_f \sum_{J \notin B} (\mathbf{n} \cdot \nabla N_J) T_J dA \tag{5.374b}$$

where $B$ is the set of nodes on the wall.

These quantities, when substituted into (5.370), give the computed values of $h_I$ and $T_{\infty,I}$.

#### 5.7.4.2. Turbulent flow modeling with wall functions

In turbulent flow where the boundary layer is not resolved, wall boundary conditions are applied by assuming that the first layer of nodes in the fluid lies not exactly on the solid interface, but slightly away from the wall in the turbulent boundary layer. Various laws of the wall can then be used to relate quantities at these nodes to the wall values. The enthalpy wall boundary condition for turbulent flow can be written in the form (see section ??):

$$Q_{f,I} = c_I A_I (H_I - H_{W,I}) \tag{5.375}$$

where $H_I$ is the nodal enthalpy, $H_{W,I}$ is the corresponding enthalpy exactly at the wall, and $c_I$ is a coefficient that depends on the flow variables. The most obvious decomposition is to let $Q_{W,I} = -c_I A_I H_{W,I}$ and $Q_{\infty,I} = c_I A_I H_I$. However, this most obvious decomposition is incorrect. The difficulty is that enthalpy is measured on a relative scale, rather than an absolute scale like temperature. For example, consider the case where $H_I = 0$. This does not imply that $T_I = 0$; in Fuego, it usually corresponds to something near standard temperature and pressure. However, the obvious decomposition when substituted into (5.370) gives $T_{\infty,I} = 0$, which is clearly the wrong value for the conduction region boundary condition.

Thus, we should choose a decomposition that has $Q_{\infty,I} = 0$ only if $T_{\infty,I}$ should be zero. The correct choice is:

$$Q_{W,I} = -c_I T_{W,I} \left( \frac{H_I - H_{W,I}}{T_I - T_{W,I}} \right) \tag{5.376a}$$

$$Q_{\infty,I} = c_I T_I \left( \frac{H_I - H_{W,I}}{T_I - T_{W,I}} \right) \tag{5.376b}$$

where $T_{W,I}$ is the wall temperature (which for conjugate heat transfer has been obtained from the solid at the previous time step), and $T_I$ is the temperature value at node $I$ (slightly away from the wall). These expressions are undefined if $T_I = T_{W,I}$; in that case, the fraction $\Delta H / \Delta T$ is approximated using the limiting value given by the specific heat $c_p$.

# 5.8. ELEMENT TOPOLOGY AND SHAPE FUNCTIONS

The standard mesh configuration for cell-centered CVFEM's is to co-locate all flow variables at the nodes, also called grid points. The nodes are the vertices of the finite-elements, as shown in Figure 5.0-1. The finite-volumes, also called control volumes, are centered about the nodes. Each element contains a set of sub-faces that define control-volumes. The sub-faces consist of the segments or surfaces that bisect the element faces. For example, each control volume on an orthogonal mesh of rectangular elements is defined by four neighboring elements with contributions from the nine nodal values.

Interpolation functions are formed inside each element. In standard finite element methods, the interpolation functions are called shape functions and they are used to evaluate the integral quadratures. The same bilinear or trilinear shape functions are used in CVFEM to construct fluxes at the sub-faces. Finite-element basis functions are used as interpolation functions to integrate fluxes over control volume faces which are internal to an element. The control-volume flux interpolation functions are element based; a restriction by choice, motivated by code development considerations. In an element-based scheme, only information that defines an element may be used to assemble fluxes. Nodal information outside the element cannot be used. As a result, the global spatial accuracy is restricted to second order.

Isoparametric shape functions are used for quadrilateral and hexahedral elements. The geometry of an isoparametric element is approximated with the same shape function as the solution variables so that the bilinear/trilinear variation within remains independent of orientation. Triangular and tetrahedral elements do not require an isoparametric formulation because they are linear. The triangles and tetrahedra can be made to look like isoparametric elements in order to create a general element evaluation algorithm.

## 5.8.1. Quadrilateral Elements

The quadrilateral element has four nodes and four control volume faces. The element configuration is shown in Figure 5.8-1. The parametric variables are $\xi$ and $\eta$, and they are coincident with the faces of the control volumes. The control volume faces are formed by the straight line segments the connect the bisection points of opposing element edges. The parametric variables have the range $-1 \leq \xi \leq 1$ and $-1 \leq \eta \leq 1$.

Geometric information inside the element is interpolated from the nodal coordinates. Derivatives of the physical coordinates are the most fundamental geometric quantity, contributing to the surface areas and gradients.

$$x = N_k X_k \qquad y = N_k Y_k \tag{5.377}$$

$$\frac{\partial x}{\partial \xi} = \frac{\partial N_k}{\partial \xi} X_k \qquad \frac{\partial y}{\partial \xi} = \frac{\partial N_k}{\partial \xi} Y_k \tag{5.378}$$

$$\frac{\partial x}{\partial \eta} = \frac{\partial N_k}{\partial \eta} X_k \qquad \frac{\partial y}{\partial \eta} = \frac{\partial N_k}{\partial \eta} Y_k \tag{5.379}$$

The subscripts on the shape functions correspond to the element-local node numbering. The isoparametric shape functions and shape function derivatives for quadrilateral elements are given in Table 5.8-1.

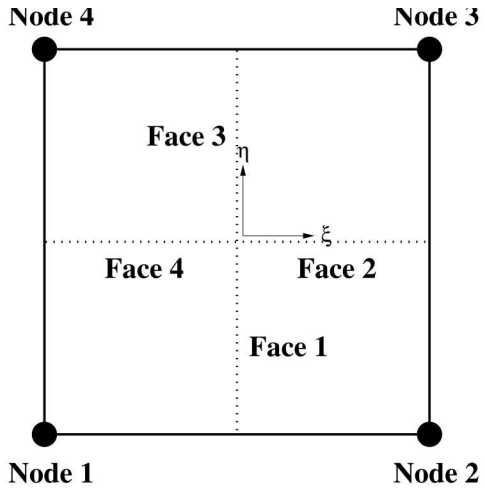


**Figure 5.8-1.. Quadrilateral element topology and numbering**

**Table 5.8-1.. Nodal shape functions and derivatives for quadrilateral elements**

| node | $N$ | $\frac{\partial N}{\partial \xi}$ | $\frac{\partial N}{\partial \eta}$ |
|---|---|---|---|
| 1 | $\frac{1}{4}(1-\xi)(1-\eta)$ | $-\frac{1}{4}(1-\eta)$ | $-\frac{1}{4}(1-\xi)$ |
| 2 | $\frac{1}{4}(1+\xi)(1-\eta)$ | $\frac{1}{4}(1-\eta)$ | $-\frac{1}{4}(1+\xi)$ |
| 3 | $\frac{1}{4}(1+\xi)(1+\eta)$ | $\frac{1}{4}(1+\eta)$ | $\frac{1}{4}(1+\xi)$ |
| 4 | $\frac{1}{4}(1-\xi)(1+\eta)$ | $-\frac{1}{4}(1+\eta)$ | $\frac{1}{4}(1-\xi)$ |

The physical surface differentials are related to differentials in parametric space. The surface area differentials, $n_i \mathrm{dA}$ are derived from their three-dimensional counterpart, Equation 5.383, where $x_\zeta = 0$, $y_\zeta = 0$, and $z_\zeta = 1$. The derivatives used in the mapping from a differential in parametric space to a differential in physical space are evaluated using Equations 5.378 and 5.379. The differential surfaces of a control-volume sub-face are surfaces of constant $\xi$ or $\eta$. Along Face 1 and 3, the differential $\mathrm{d}\xi = 0$. Along Face 2 and 4, the differential $\mathrm{d}\eta = 0$. For the purposes of constructing a general-purpose computational flux algorithm, integration over both parametric components is retained. On each face, only one surface area component will be non-zero.

$$n_i \mathrm{d}S = \begin{bmatrix} -y_\xi & x_\xi \end{bmatrix} \mathrm{d}\xi + \begin{bmatrix} y_\eta & -x_\eta \end{bmatrix} \mathrm{d}\eta \tag{5.380}$$

The usefulness of the general approach will become more apparent when triangular elements are considered.

The normals to the control-volume surfaces are positive in the direction of positive coordinate axes. The normal to a $\xi$-constant face is along the positive $\xi$-axis. The normal to a $\eta$-constant face is along the positive $\eta$-axis. The signs on the differentials are selected such that the fluxes have the proper signs relative to the control volume. The values of the element variables and the surface differentials at the control-volume faces are given in Table 5.8-2. The differential $\mathrm{d}\eta$ is negative for Face 3 because the direction for out/in flow from Node 3 to Node 4 is opposite in direction of the surface normal defined by $n_i \mathrm{d}S$.

**Table 5.8-2.. Element variable values and differentials at control-volume faces for quadrilateral elements. Face-to-edge number mapping.**

| face | Edge ( $\mathrm{Node}_{\mathrm{out}} \rightarrow \mathrm{Node}_{\mathrm{in}}$ ) | $\xi$ | $\eta$ | $\mathrm{d}\xi$ | $\mathrm{d}\eta$ |
|---|---|---|---|---|---|
| 1 | $1 \rightarrow 2$ | 0 | $-\frac{1}{2}$ | 0 | 1 |
| 2 | $2 \rightarrow 3$ | $\frac{1}{2}$ | 0 | 1 | 0 |
| 3 | $3 \rightarrow 4$ | 0 | $\frac{1}{2}$ | 0 | $-1$ |
| 4 | $1 \rightarrow 4$ | $-\frac{1}{2}$ | 0 | 1 | 0 |

Volume integrals require the volume differential, $\mathrm{d}x\mathrm{d}y$. In terms of the element parameters, the volume differential is

$$\mathrm{d}x\mathrm{d}y = J\mathrm{d}\xi\mathrm{d}\eta, \tag{5.381}$$

where

$$J = x_\xi y_\eta - x_\eta y_\xi. \tag{5.382}$$

The quadrature points and differential values are shown in Table 5.8-3.

**Table 5.8-3.. Element variable values and differentials at sub-control volume centers for quadrilateral elements.**

| sub-volume | $\xi$ | $\eta$ | $d\xi d\eta$ |
|---|---|---|---|
| 1 | $-\frac{1}{2}$ | $-\frac{1}{2}$ | 1 |
| 2 | $\frac{1}{2}$ | $-\frac{1}{2}$ | 1 |
| 3 | $\frac{1}{2}$ | $\frac{1}{2}$ | 1 |
| 4 | $-\frac{1}{2}$ | $\frac{1}{2}$ | 1 |

## 5.8.2. Triangular Elements

The triangular element has three nodes and three control volume faces. The element configuration is shown in Figure 5.8-2. The control volume faces run from the centroid of the element to the element edge bisection points. The parametric coordinate system is defined by the triangle natural coordinates,

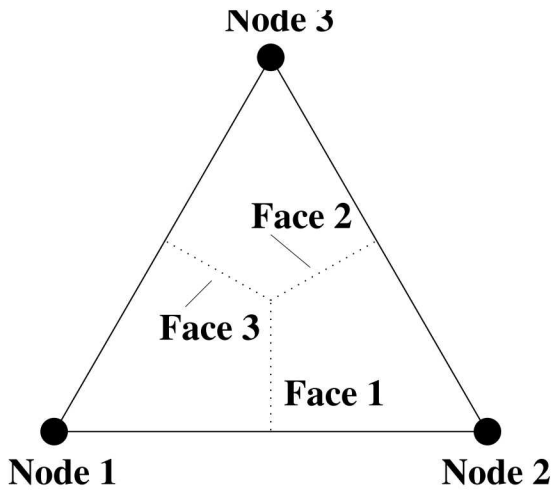


**Figure 5.8-2.. Triangular element topology and numbering**

$L_1$, $L_2$ and $L_3$, since a Cartesian mapping cannot be defined. The natural coordinates are the shape functions. As an example, the value of $L_1$ at an interpolation point is the shape function associated with Node 1. The value of $L_1$ is the fraction of the element triangle area covered by a sub-triangle, formed by the interpolation point and the edge opposite of Node 1, shown in Figure 5.8-3. For consistency with the quadrilateral element notation, the $(\xi, \eta)$ parametric variables are defined as $\xi = L_1$ and $\eta = L_2$, where $L_3$ is defined by the fact that the natural coordinates always sum to one. The linear shape functions and shape function derivatives for triangular elements are given in Table 5.8-4.

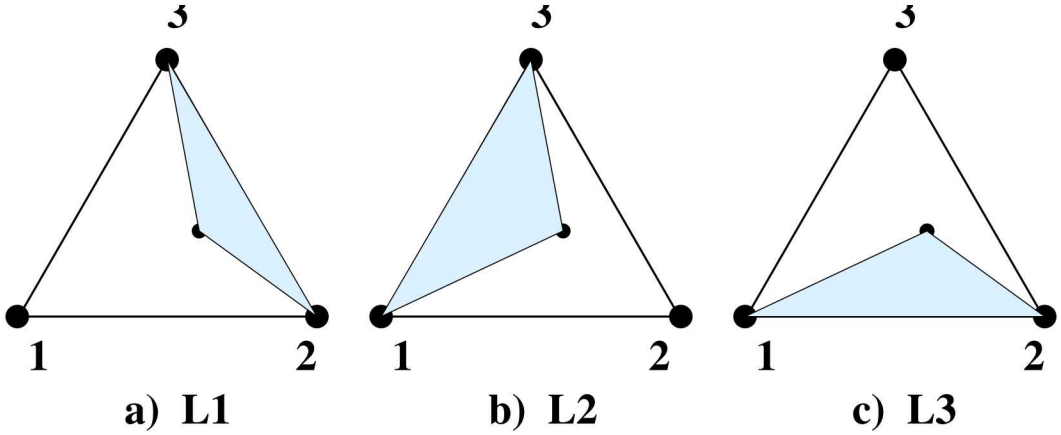


**Figure 5.8-3.. Triangular natural coordinate system, shaded area corresponds to opposite node.**

**Table 5.8-4.. Nodal shape functions and derivatives for triangular elements.**

| node | $N$ | $\frac{\partial N}{\partial \xi}$ | $\frac{\partial N}{\partial \eta}$ |
|---|---|---|---|
| 1 | $\xi$ | 1 | 0 |
| 2 | $\eta$ | 0 | 1 |
| 3 | $1-\xi-\eta$ | $-1$ | $-1$ |

The surface integrals are tricky because there is no surface that lays on a line of constant $\xi$ or $\eta$. Along Face 1, $1/2 > \xi > 1/3$ and $1/2 > \eta > 1/3$. Along Face 2, $1/3 > \xi > 0$ and $1/3 < \eta < 1/2$. Along Face 3, $1/3 < \xi < 1/2$ and $1/3 > \eta > 0$. The integrations are taken from the centroid to the element edges. The values of the element variables and the surface differentials at the control-volume faces are given in Table 5.8-5.

The form of the volume differentials are the same as with the quadrilateral elements. For volume integrals, quadrature points and differential values are shown in Table 5.8-6.

## 5.8.3. Hexahedral Elements

For hexahedral elements, there are eight nodes and twelve subfaces defining control volumes, shown in Figure 5.8-4. The shape functions are trilinear functions of the element variables, $\xi$, $\eta$, and $\zeta$. The shape functions and derivatives at each node are given in Table 5.8-7. The control volume sub-face numbering, shown in Table 5.8-8, follows the convention that the face has the same number as the element edge that connects the nodes that define the two adjacent sub-control volumes.

The surface integrals require the vector of differential surface area components, $(\mathrm{d}A_x, \mathrm{d}A_y, \mathrm{d}A_z)$, which is equivalent to the differential surface area $\mathrm{d}S$ multiplied by the unit surface normal vector $n_i$. Since the control volume surfaces are constructed using four points within an element, it is noted that

**Table 5.8-5.. Element variable values and differentials at control-volume faces for triangular elements**

| face | Edge ( $\mathrm{Node}_{\mathrm{out}} \rightarrow \mathrm{Node}_{\mathrm{in}}$ ) | $\xi$ | $\eta$ | $\mathrm{d}\xi$ | $\mathrm{d}\eta$ |
|---|---|---|---|---|---|
| 1 | $1 \rightarrow 2$ | $\frac{5}{12}$ | $\frac{5}{12}$ | $\frac{1}{6}$ | $-\frac{1}{6}$ |
| 2 | $2 \rightarrow 3$ | $\frac{1}{6}$ | $\frac{5}{12}$ | $-\frac{1}{3}$ | $-\frac{1}{6}$ |
| 3 | $3 \rightarrow 1$ | $\frac{5}{12}$ | $\frac{1}{6}$ | $\frac{1}{6}$ | $\frac{1}{3}$ |

**Table 5.8-6.. Element variable values and differentials at sub-control volume centers for triangular elements.**

| sub-volume | $\xi$ | $\eta$ | $\mathrm{d}\xi\mathrm{d}\eta$ |
|---|---|---|---|
| 1 | $\frac{7}{12}$ | $\frac{5}{24}$ | $\frac{1}{6}$ |
| 2 | $\frac{5}{24}$ | $\frac{7}{12}$ | $\frac{1}{6}$ |
| 3 | $\frac{5}{24}$ | $\frac{5}{24}$ | $\frac{1}{6}$ |

assuming the surfaces are planar results in an error. Sometimes this error is deemed acceptable, and a faster algorithm is used to compute the surface area and volume. However, when strict conservation is required, an exact algorithm using a polyhedral decomposition is employed to compute the exact volume and surface area. These are detailed below.

#### *5.8.3.1. Volume and Area Calculation Assuming Planar Surfaces*

The differential surface area, $n_i \mathrm{d}S$, is calculated in parametric space by taking the cross-product of two differential surface-tangent vectors. Let the surface be described by the collection of points $S(x, y, z)$. For example, a tangent vector in the $\xi$-direction is $\partial x_i / \partial \xi$. The normal surface area component for all three possible surface parameterizations is

$$n_k \mathrm{d}S = \left[\frac{\partial x_i}{\partial \eta} \epsilon_{ijk} \frac{\partial x_j}{\partial \zeta} \mathrm{d}\eta \mathrm{d}\zeta\right] + \left[\frac{\partial x_i}{\partial \xi} \epsilon_{ijk} \frac{\partial x_j}{\partial \eta} \mathrm{d}\xi \mathrm{d}\eta\right] + \left[\frac{\partial x_i}{\partial \zeta} \epsilon_{ijk} \frac{\partial x_j}{\partial \xi} \mathrm{d}\zeta \mathrm{d}\xi\right] \tag{5.383}$$

where $\epsilon_{ijk}$ is the alternating unit tensor and defines the cross product.

$$\epsilon_{ijk} = \begin{cases} 1 & \text{if } ijk \text{ equals an even permutation 123, 231, or 312} \\ 0 & \text{if } ijk \text{ contains a repeated index} \\ -1 & \text{if } ijk \text{ equals an odd permutation 132, 213, or 321} \end{cases} \tag{5.384}$$

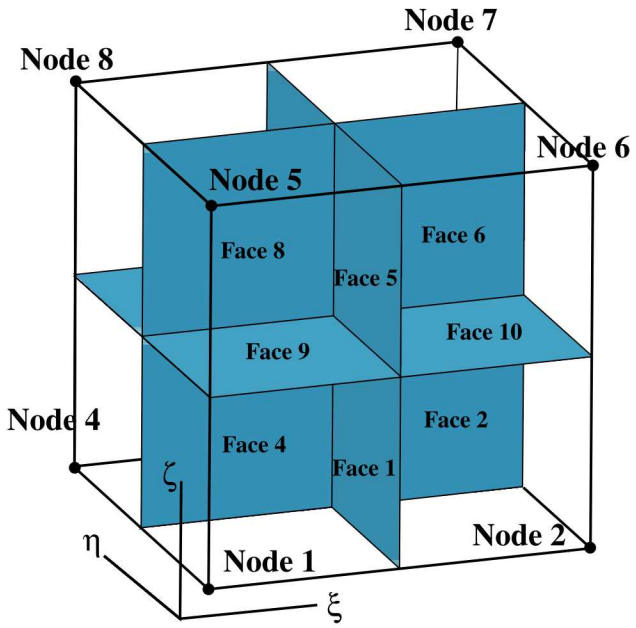


**Figure 5.8-4.. Hexahedral element topology and numbering**

Using a shortened notation,

$$\begin{aligned} n_i \mathrm{d}S &= \left[ \left|\frac{\partial(y,z)}{\partial(\eta,\zeta)}\right| \quad \left|\frac{\partial(z,x)}{\partial(\eta,\zeta)}\right| \quad \left|\frac{\partial(x,y)}{\partial(\eta,\zeta)}\right| \right] \mathrm{d}\eta\mathrm{d}\zeta \\ &+ \left[ \left|\frac{\partial(y,z)}{\partial(\zeta,\xi)}\right| \quad \left|\frac{\partial(z,x)}{\partial(\zeta,\xi)}\right| \quad \left|\frac{\partial(x,y)}{\partial(\zeta,\xi)}\right| \right] \mathrm{d}\zeta\mathrm{d}\xi \\ &+ \left[ \left|\frac{\partial(y,z)}{\partial(\xi,\eta)}\right| \quad \left|\frac{\partial(z,x)}{\partial(\xi,\eta)}\right| \quad \left|\frac{\partial(x,y)}{\partial(\xi,\eta)}\right| \right] \mathrm{d}\xi\mathrm{d}\eta, \end{aligned} \tag{5.385}$$

where the Jacobian notation is defined by

$$\frac{\partial(x,y)}{\partial(\xi,\eta)} = \begin{bmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{bmatrix}. \tag{5.386}$$

The values of the element variables and the surface differentials at the control-volume faces are given in Table 5.8-8.

Volume integrals require the volume differential, $\mathrm{d}x\mathrm{d}y\mathrm{d}z$. In terms of the element parameters, the volume differential is

$$\mathrm{d}x\mathrm{d}y\mathrm{d}z = J\mathrm{d}\xi\mathrm{d}\eta\mathrm{d}\zeta, \tag{5.387}$$

where

$$\begin{aligned} J &= x_\xi y_\eta z_\zeta - x_\zeta y_\eta z_\xi \\ &+ x_\eta y_\zeta z_\xi - x_\xi y_\zeta z_\eta \\ &+ x_\zeta y_\xi z_\eta - x_\eta y_\xi z_\zeta \end{aligned} \tag{5.388}$$

The quadrature points and differential values are shown in Table 5.8-9.

**Table 5.8-7.. Nodal shape functions and derivatives for hexahedral elements. Range is (-1,1).**

| node | $N$ | $\frac{\partial N}{\partial \xi}$ | $\frac{\partial N}{\partial \eta}$ | $\frac{\partial N}{\partial \zeta}$ |
|---|---|---|---|---|
| 1 | $\frac{1}{8}(1-\xi)(1-\eta)(1-\zeta)$ | $-\frac{1}{8}(1-\eta)(1-\zeta)$ | $-\frac{1}{8}(1-\xi)(1-\zeta)$ | $-\frac{1}{8}(1-\xi)(1-\eta)$ |
| 2 | $\frac{1}{8}(1+\xi)(1-\eta)(1-\zeta)$ | $\frac{1}{8}(1-\eta)(1-\zeta)$ | $-\frac{1}{8}(1+\xi)(1-\zeta)$ | $-\frac{1}{8}(1+\xi)(1-\eta)$ |
| 3 | $\frac{1}{8}(1+\xi)(1+\eta)(1-\zeta)$ | $\frac{1}{8}(1+\eta)(1-\zeta)$ | $\frac{1}{8}(1+\xi)(1-\zeta)$ | $-\frac{1}{8}(1+\xi)(1+\eta)$ |
| 4 | $\frac{1}{8}(1-\xi)(1+\eta)(1-\zeta)$ | $-\frac{1}{8}(1+\eta)(1-\zeta)$ | $\frac{1}{8}(1-\xi)(1-\zeta)$ | $-\frac{1}{8}(1-\xi)(1+\eta)$ |
| 5 | $\frac{1}{8}(1-\xi)(1-\eta)(1+\zeta)$ | $-\frac{1}{8}(1-\eta)(1+\zeta)$ | $-\frac{1}{8}(1-\xi)(1+\zeta)$ | $\frac{1}{8}(1-\xi)(1-\eta)$ |
| 6 | $\frac{1}{8}(1+\xi)(1-\eta)(1+\zeta)$ | $\frac{1}{8}(1-\eta)(1+\zeta)$ | $-\frac{1}{8}(1+\xi)(1+\zeta)$ | $\frac{1}{8}(1+\xi)(1-\eta)$ |
| 7 | $\frac{1}{8}(1+\xi)(1+\eta)(1+\zeta)$ | $\frac{1}{8}(1+\eta)(1+\zeta)$ | $\frac{1}{8}(1+\xi)(1+\zeta)$ | $\frac{1}{8}(1+\xi)(1+\eta)$ |
| 8 | $\frac{1}{8}(1-\xi)(1+\eta)(1+\zeta)$ | $-\frac{1}{8}(1+\eta)(1+\zeta)$ | $\frac{1}{8}(1-\xi)(1+\zeta)$ | $\frac{1}{8}(1-\xi)(1+\eta)$ |

#### *5.8.3.2. Exact Volume and Surface Area Calculation*

When the planar surface assumption for the control volumes is insufficient, the volume and surface areas can be calculated exactly. To accomplish this, a set of subcontrol points is constructed that defines the subcontrol surfaces. The locations and numbering of these subcontrol points are shown in Figure 5.8-5. The coordinates of the edge points are the average of the two adjacent vertices, the coordinates of the facial points are the average of the four vertices defining the face, and the coordinates of the interior point is the average of the eight vertices defining the volume.

The 12 subcontrol surfaces for the Hexahedron are the defined using points in counterclockwise ordering as shown in Table 5.8-10. These surfaces are further broken down into four triangles defined by the four points on the surface and a simply averaged midpoint. The four triangles are defined by points $\{5,1,2\}$, $\{5,2,3\}$, $\{5,3,4\}$, and $\{5,4,1\}$, respectively. The area vectors of each triangle are summed to calculate the total surface area vector. Noting that the triangles are planar, the area vector of each triangle is calculated exactly using half the cross product of any two right-hand oriented vectors.

The eight subcontrol volumes are defined using the points shown in Table 5.8-11. The formula to calculate the exact volume is based on the Gauss Divergence formula,

$$V = \int_\Omega \mathrm{d}V = \int_\Omega \frac{\partial x_k}{\partial x_k}\mathrm{d}V = \oint_{\partial\Omega} x_k n_k \mathrm{d}S. \tag{5.389}$$

The surfaces for the surface integral are decomposed into triangular facets as in the surface area calculation. To accomplish the decomposition, the coordinates on each face are averaged to the midpoints, and thus each hexahedral volume is constructed using 14 coordinates–eight vertices and six facial midpoints, resulting in 24 total facets. Since the triangular facets are planar, the normal is constant over the surface. Thus, the surface integral over each triangular facet is equivalent to the scalar product

**Table 5.8-8.. Element variable values and differentials at control-volume faces for hexahedral elements. Face-to-edge number mapping.**

| face | Edge ( $\mathrm{Node}_{\mathrm{out}} \rightarrow \mathrm{Node}_{\mathrm{in}}$ ) | $\xi$ | $\eta$ | $\zeta$ | $\mathrm{d}\eta\mathrm{d}\zeta$ | $\mathrm{d}\zeta\mathrm{d}\xi$ | $\mathrm{d}\xi\mathrm{d}\eta$ |
|---|---|---|---|---|---|---|---|
| 1 | $1 \rightarrow 2$ | 0 | $-\frac{1}{2}$ | $-\frac{1}{2}$ | 1 | 0 | 0 |
| 2 | $2 \rightarrow 3$ | $\frac{1}{2}$ | 0 | $-\frac{1}{2}$ | 0 | 1 | 0 |
| 3 | $3 \rightarrow 4$ | 0 | $\frac{1}{2}$ | $-\frac{1}{2}$ | $-1$ | 0 | 0 |
| 4 | $1 \rightarrow 4$ | $-\frac{1}{2}$ | 0 | $-\frac{1}{2}$ | 0 | 1 | 0 |
| 5 | $5 \rightarrow 6$ | 0 | $-\frac{1}{2}$ | $\frac{1}{2}$ | 1 | 0 | 0 |
| 6 | $6 \rightarrow 7$ | $\frac{1}{2}$ | 0 | $\frac{1}{2}$ | 0 | 1 | 0 |
| 7 | $7 \rightarrow 8$ | 0 | $\frac{1}{2}$ | $\frac{1}{2}$ | $-1$ | 0 | 0 |
| 8 | $5 \rightarrow 8$ | $-\frac{1}{2}$ | 0 | $\frac{1}{2}$ | 0 | 1 | 0 |
| 9 | $1 \rightarrow 5$ | $-\frac{1}{2}$ | $-\frac{1}{2}$ | 0 | 0 | 0 | 1 |
| 10 | $2 \rightarrow 6$ | $\frac{1}{2}$ | $-\frac{1}{2}$ | 0 | 0 | 0 | 1 |
| 11 | $3 \rightarrow 7$ | $\frac{1}{2}$ | $\frac{1}{2}$ | 0 | 0 | 0 | 1 |
| 12 | $4 \rightarrow 8$ | $-\frac{1}{2}$ | $\frac{1}{2}$ | 0 | 0 | 0 | 1 |

of the outward facing normal area vector and the centroid coordinates, $\bar{x}$. The total surface integral is the sum of the integrals on each triangular facet,

$$V = \oint_{\partial\Omega} x_k n_k \mathrm{d}S = \sum_{i=1}^{24} n_k \int_{\Delta_i} x_k \mathrm{d}S = \sum_{i=1}^{24} \bar{x}_k n_k \int_{\Delta_i} \mathrm{d}S, \quad \partial\Omega = \bigoplus_{i=1}^{24} \Delta_i. \tag{5.390}$$

The area vectors are calculated as described above. The centroid coordinates are simply the average of the three vertices constructed the triangular facet.

## 5.8.4. Tetrahedral Elements

For tetrahedral elements, there are four nodes and six subfaces defining control volumes, shown in Figure 5.8-6. The parametric coordinate system is defined by the tetrahedron natural coordinates, $L_1$, $L_2$, $L_3$, and $L_4$, since a Cartesian mapping cannot be defined. The natural coordinates are the shape functions. As an example, the value of $L_1$ at an interpolation point is the shape function associated with Node 2. The value of $L_1$ is the fraction of element tetrahedral volume covered by a sub-tetrahedron, formed by the interpolation point and the face opposite of Node 2. For consistency with the hexahedral element notation, the $(\xi, \eta, \zeta)$ parametric variables are defined as $\xi = L_1$, $\eta = L_2$, and $\zeta = L_3$, where

**Table 5.8-9.. Element variable values and differentials at sub-control volume centers for hexahedral elements.**

| sub-volume | $\xi$ | $\eta$ | $\zeta$ | $d\xi d\eta d\zeta$ |
|---|---|---|---|---|
| 1 | $-\frac{1}{2}$ | $-\frac{1}{2}$ | $-\frac{1}{2}$ | 1 |
| 2 | $\frac{1}{2}$ | $-\frac{1}{2}$ | $-\frac{1}{2}$ | 1 |
| 3 | $\frac{1}{2}$ | $\frac{1}{2}$ | $-\frac{1}{2}$ | 1 |
| 4 | $-\frac{1}{2}$ | $\frac{1}{2}$ | $-\frac{1}{2}$ | 1 |
| 5 | $-\frac{1}{2}$ | $-\frac{1}{2}$ | $\frac{1}{2}$ | 1 |
| 6 | $\frac{1}{2}$ | $-\frac{1}{2}$ | $\frac{1}{2}$ | 1 |
| 7 | $\frac{1}{2}$ | $\frac{1}{2}$ | $\frac{1}{2}$ | 1 |
| 8 | $-\frac{1}{2}$ | $\frac{1}{2}$ | $\frac{1}{2}$ | 1 |

$L_4$ is defined by the fact that the natural coordinates always sum to one. The control volume sub-face numbering, shown in Table 5.8-13, follows the convention that the face has the same number as the element edge that connects the nodes that define the two adjacent sub-control volumes.

The values of the element variables and the surface differentials at the control-volume faces are given in Table 5.8-13.

Again the control volumes are constructed using surfaces defined with four points and two methods are available to define the surface area and volume.

#### *5.8.4.1. Volume and Area Calculation Assuming Planar Surfaces*

The form of the volume differentials are the same as with the hexahedral elements. For volume integrals, quadrature points and differential values are shown in Table 5.8-14.

#### *5.8.4.2. Exact Volume and Surface Area Calculation*

Following the approach in Section 5.8.3.2, a set of subcontrol coordinates is defined to decompose the tetrahedral element, which are shown in Figure 5.8-7.

Six subcontrol surfaces for the tetrahedron are the defined using points in counterclockwise ordering as shown in Table 5.8-15. Surface area vectors are calculated using the same approach as in Section 5.8.3.2.

Four subcontrol volumes are defined using the points shown in Table 5.8-16. Since the subcontrol volumes are hexahedrons, the same volume calculation is used as above.

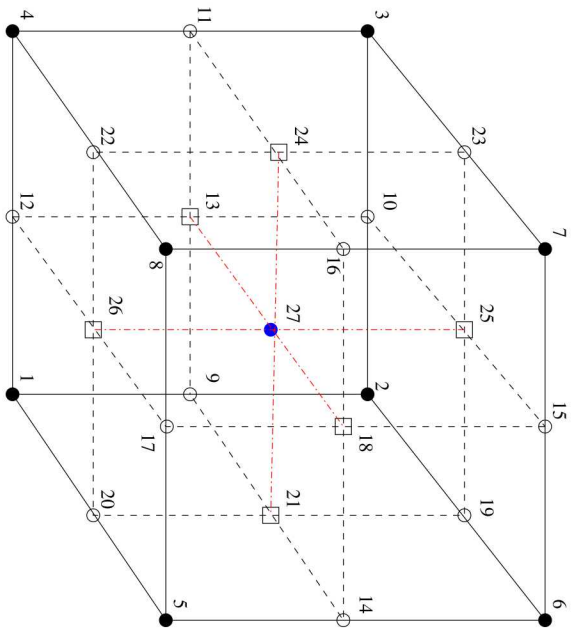


**Figure 5.8-5.. Hexahedron subcontrol points numbering**

## 5.8.5. Wedge Elements

For wedge elements, there are six nodes and nine subfaces defining control volumes. The parametric coordinate system is a linear hybrid of triangular natural coordinates. The natural coordinates are the shape functions. The local coordinates $\xi$ and $\eta$ are in the plane of the triangular surfaces while $\zeta$ is in the normal direction. The control volume sub-face numbering, shown in Table 5.8-18, follows the convention that the face has the same number as the element edge that connects the nodes that define the two adjacent sub-control volumes.

### *5.8.5.1. Volume and Area Calculation Assuming Planar Surfaces*

The values of the element variables and the surface differentials at the control-volume faces are given in Table 5.8-18.

For volume integrals, quadrature points and differential values are shown in Table 5.8-19.

### *5.8.5.2. Exact Volume and Surface Area Calculation*

Following the approach in Section 5.8.3.2, a set of subcontrol coordinates is defined to decompose the wedge element, which are shown in Figure 5.8-8.

Nine subcontrol surfaces for the tetrahedron are the defined using points in counterclockwise ordering as shown in Table 5.8-20. Surface area vectors are calculated using the same approach as in Section 5.8.3.2.

**Table 5.8-10.. Subcontrol face definitions for exact surface area calculation on hexahedral elements.**

| Face | Point Set | | | |
|---|---|---|---|---|
| 1 | 21 | 9 | 13 | 27 |
| 2 | 25 | 10 | 13 | 27 |
| 3 | 11 | 13 | 27 | 24 |
| 4 | 12 | 26 | 27 | 13 |
| 5 | 14 | 21 | 27 | 18 |
| 6 | 18 | 15 | 25 | 27 |
| 7 | 18 | 16 | 24 | 27 |
| 8 | 17 | 18 | 27 | 26 |
| 9 | 20 | 21 | 27 | 26 |
| 10 | 21 | 19 | 25 | 27 |
| 11 | 23 | 24 | 27 | 25 |
| 12 | 22 | 26 | 27 | 24 |

**Table 5.8-11.. Subcontrol volume definitions for exact volume calculation on hexahedral elements.**

| Volume | Point Set | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 9 | 13 | 12 | 20 | 21 | 27 | 26 |
| 2 | 9 | 2 | 10 | 13 | 21 | 19 | 25 | 27 |
| 3 | 13 | 10 | 3 | 11 | 27 | 25 | 23 | 24 |
| 4 | 12 | 13 | 11 | 4 | 26 | 27 | 24 | 22 |
| 5 | 20 | 21 | 27 | 26 | 5 | 14 | 18 | 17 |
| 6 | 21 | 19 | 25 | 27 | 14 | 6 | 15 | 18 |
| 7 | 27 | 25 | 23 | 24 | 18 | 15 | 7 | 16 |
| 8 | 26 | 27 | 24 | 22 | 17 | 18 | 16 | 8 |

Six subcontrol volumes are defined using the points shown in Table 5.8-21. Since the subcontrol volumes are hexahedrons, the same volume calculation is used as above.

## 5.8.6. Pyramid Elements

For pyramid elements, there are five nodes and eight subfaces defining control volumes. The local coordinates $\xi$ and $\eta$ are in the plane of the quadrilateral surfaces while $\zeta$ is in the normal direction. The control volume sub-face numbering, shown in Table 5.8-23, follows the convention that the face has the same number as the element edge that connects the nodes that define the two adjacent sub-control volumes.

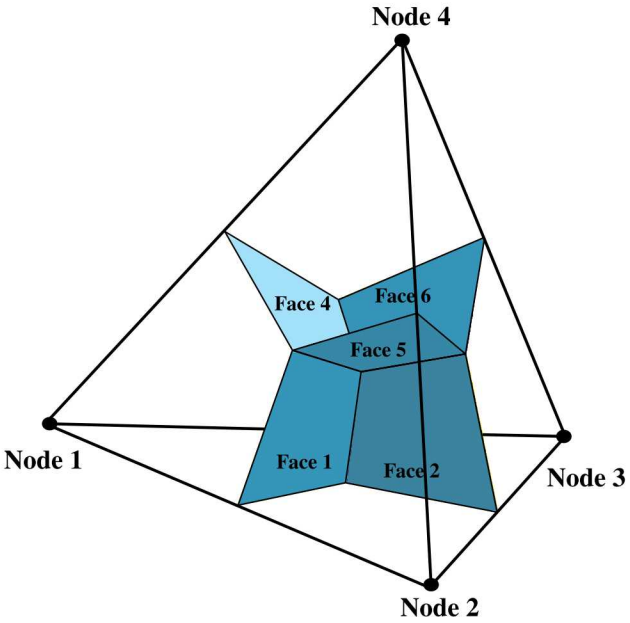


**Figure 5.8-6.. Tetrahedral element topology and numbering**

**Table 5.8-12.. Nodal shape functions and derivatives for tetrahedral elements. Range is (0,1).**

| node | $N$ | $\frac{\partial N}{\partial \xi}$ | $\frac{\partial N}{\partial \eta}$ | $\frac{\partial N}{\partial \zeta}$ |
|---|---|---|---|---|
| 1 | $1-\xi-\eta-\zeta$ | $-1$ | $-1$ | $-1$ |
| 2 | $\xi$ | 1 | 0 | 0 |
| 3 | $\eta$ | 0 | 1 | 0 |
| 4 | $\zeta$ | 0 | 0 | 1 |

#### *5.8.6.1. Volume and Area Calculation Assuming Planar Surfaces*

The values of the element variables and the surface differentials at the control-volume faces are given in Table 5.8-23.

For volume integrals, quadrature points and differential values are shown in Table 5.8-24.

#### *5.8.6.2. Exact Volume and Surface Area Calculation*

It is noted here that for pyramid elements, the planar assumption is not good even on the reference element. The volume that composes the tip is an octohedron four planar faces and four highly skewed faces. Computations have shown that the planar assumption results in severe conservation errors.

**Table 5.8-13.. Element variable values and differentials at control-volume faces for tetrahedral elements. Face-to-edge number mapping.**

| face | Edge ( $\text{Node}_{\text{out}} \rightarrow \text{Node}_{\text{in}}$ ) | $\xi$ | $\eta$ | $\zeta$ | $d\eta d\zeta$ | $d\zeta d\xi$ | $d\xi d\eta$ |
|---|---|---|---|---|---|---|---|
| 1 | $1 \rightarrow 2$ | $\frac{17}{48}$ | $\frac{7}{48}$ | $\frac{7}{48}$ | | | |
| 2 | $2 \rightarrow 3$ | $\frac{17}{48}$ | $\frac{17}{48}$ | $\frac{7}{48}$ | | | |
| 3 | $1 \rightarrow 3$ | $\frac{7}{48}$ | $\frac{17}{48}$ | $\frac{7}{48}$ | | | |
| 4 | $1 \rightarrow 4$ | $\frac{7}{48}$ | $\frac{7}{48}$ | $\frac{17}{48}$ | | | |
| 5 | $2 \rightarrow 4$ | $\frac{17}{48}$ | $\frac{7}{48}$ | $\frac{17}{48}$ | | | |
| 6 | $3 \rightarrow 4$ | $\frac{7}{48}$ | $\frac{17}{48}$ | $\frac{17}{48}$ | | | |

**Table 5.8-14.. Element variable values and differentials at sub-control volume centers for tetrahedral elements.**

| sub-volume | $\xi$ | $\eta$ | $\zeta$ | $d\xi d\eta d\zeta$ |
|---|---|---|---|---|
| 1 | $\frac{17}{96}$ | $\frac{17}{96}$ | $\frac{17}{96}$ | |
| 2 | $\frac{45}{96}$ | $\frac{17}{96}$ | $\frac{17}{96}$ | |
| 3 | $\frac{17}{96}$ | $\frac{45}{96}$ | $\frac{17}{96}$ | |
| 4 | $\frac{17}{96}$ | $\frac{17}{96}$ | $\frac{45}{96}$ | |

Following the approach in Section 5.8.3.2, a set of subcontrol coordinates is defined to decompose the pyramid element, which are shown in Figure 5.8-9.

Eight subcontrol surfaces for the tetrahedron are the defined using points in counterclockwise ordering as shown in Table 5.8-25. Surface area vectors are calculated using the same approach as in Section 5.8.3.2.

Five subcontrol volumes are defined using the points shown in Table 5.8-26. The first four subcontrol volumes are hexahedrons, so the same volume calculation is used as above. The tip of the pyramid composes an octohedron, but the computation of the volume is only slightly different. The Gauss Divergence Theorem is still used to calculate the volume. However, because the four faces on the pyramid faces must be planar, these faces are decomposed into two triangles–composed of the face midpoint and the pyramid tip vertex–instead of four triangles. The four-point faces interior to the triangle are not planar and are decomposed into four triangles, resulting in 24 triangular facets total. Equation 5.390 is then applied to compute the volume.

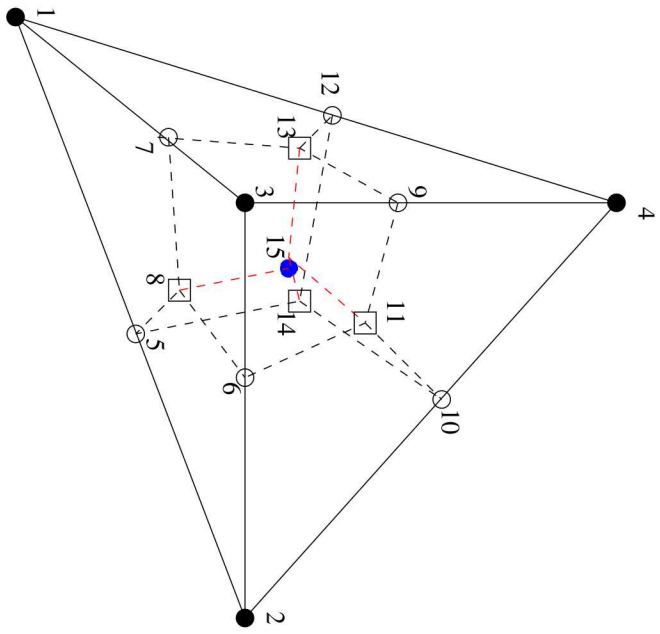


**Figure 5.8-7.. Tetrahedron subcontrol points numbering**

**Table 5.8-15.. Subcontrol face definitions for exact surface area calculation on tetrahedral elements.**

| Face | Point Set | | | |
|---|---|---|---|---|
| 1 | 5 | 8 | 15 | 14 |
| 2 | 8 | 15 | 11 | 6 |
| 3 | 7 | 13 | 15 | 8 |
| 4 | 12 | 14 | 15 | 13 |
| 5 | 14 | 10 | 11 | 15 |
| 6 | 11 | 9 | 13 | 15 |

**Table 5.8-16.. Subcontrol volume definitions for exact volume calculation on tetrahedral elements.**

| Volume | Point Set | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 5 | 8 | 7 | 12 | 14 | 15 | 13 |
| 2 | 2 | 6 | 8 | 5 | 10 | 11 | 15 | 14 |
| 3 | 3 | 7 | 8 | 6 | 9 | 13 | 15 | 11 |
| 4 | 4 | 10 | 14 | 12 | 9 | 11 | 15 | 13 |

**Table 5.8-17.. Nodal shape functions and derivatives for wedge elements. Range is (0,1) and (-1,1).**

| node | $N$ | $\frac{\partial N}{\partial \xi}$ | $\frac{\partial N}{\partial \eta}$ | $\frac{\partial N}{\partial \zeta}$ |
|---|---|---|---|---|
| 1 | $\frac{1}{2}(1-\xi-\eta)(1-\zeta)$ | $-\frac{1}{2}(1-\zeta)$ | $-\frac{1}{2}(1-\zeta)$ | $-\frac{1}{2}(1-\xi-\eta)$ |
| 2 | $\frac{1}{2}\xi(1-\zeta)$ | $\frac{1}{2}(1-\zeta)$ | $0$ | $-\frac{1}{2}\xi$ |
| 3 | $\frac{1}{2}\eta(1-\zeta)$ | $0$ | $\frac{1}{2}(1-\zeta)$ | $-\frac{1}{2}\eta$ |
| 4 | $\frac{1}{2}(1-\xi-\eta)(1+\zeta)$ | $-\frac{1}{2}(1+\zeta)$ | $-\frac{1}{2}(1+\zeta)$ | $\frac{1}{2}(1-\xi-\eta)$ |
| 5 | $\frac{1}{2}\xi(1+\zeta)$ | $\frac{1}{2}(1+\zeta)$ | $0$ | $\frac{1}{2}\xi$ |
| 6 | $\frac{1}{2}\eta(1+\zeta)$ | $0$ | $\frac{1}{2}(1+\zeta)$ | $\frac{1}{2}\eta$ |

**Table 5.8-18.. Element variable values and differentials at control-volume faces for wedge elements. Face-to-edge number mapping.**

| face | Edge ( $\mathrm{Node}_{\mathrm{out}} \rightarrow \mathrm{Node}_{\mathrm{in}}$ ) | $\xi$ | $\eta$ | $\zeta$ | $\mathrm{d}\eta\mathrm{d}\zeta$ | $\mathrm{d}\zeta\mathrm{d}\xi$ | $\mathrm{d}\xi\mathrm{d}\eta$ |
|---|---|---|---|---|---|---|---|
| 1 | $1 \rightarrow 2$ | $\frac{5}{12}$ | $\frac{1}{6}$ | $-\frac{1}{2}$ | | | |
| 2 | $2 \rightarrow 3$ | $\frac{5}{12}$ | $\frac{5}{12}$ | $-\frac{1}{2}$ | | | |
| 3 | $1 \rightarrow 3$ | $\frac{1}{6}$ | $\frac{5}{12}$ | $-\frac{1}{2}$ | | | |
| 4 | $4 \rightarrow 5$ | $\frac{5}{12}$ | $\frac{1}{6}$ | $\frac{1}{2}$ | | | |
| 5 | $5 \rightarrow 6$ | $\frac{5}{12}$ | $\frac{5}{12}$ | $\frac{1}{2}$ | | | |
| 6 | $4 \rightarrow 6$ | $\frac{1}{6}$ | $\frac{5}{12}$ | $\frac{1}{2}$ | | | |
| 7 | $1 \rightarrow 4$ | $\frac{5}{24}$ | $\frac{5}{24}$ | $0$ | | | |
| 8 | $2 \rightarrow 5$ | $\frac{7}{12}$ | $\frac{5}{24}$ | $0$ | | | |
| 9 | $3 \rightarrow 6$ | $\frac{5}{24}$ | $\frac{7}{12}$ | $0$ | | | |

**Table 5.8-19.. Element variable values and differentials at sub-control volume centers for wedge elements.**

| sub-volume | $\xi$ | $\eta$ | $\zeta$ | $d\xi d\eta d\zeta$ |
|---|---|---|---|---|
| 1 | $\frac{5}{24}$ | $\frac{5}{24}$ | $-\frac{1}{2}$ | |
| 2 | $\frac{7}{12}$ | $\frac{5}{24}$ | $-\frac{1}{2}$ | |
| 3 | $\frac{5}{24}$ | $\frac{7}{12}$ | $-\frac{1}{2}$ | |
| 4 | $\frac{5}{24}$ | $\frac{5}{24}$ | $\frac{1}{2}$ | |
| 5 | $\frac{7}{12}$ | $\frac{5}{24}$ | $\frac{1}{2}$ | |
| 6 | $\frac{5}{24}$ | $\frac{7}{12}$ | $\frac{1}{2}$ | |

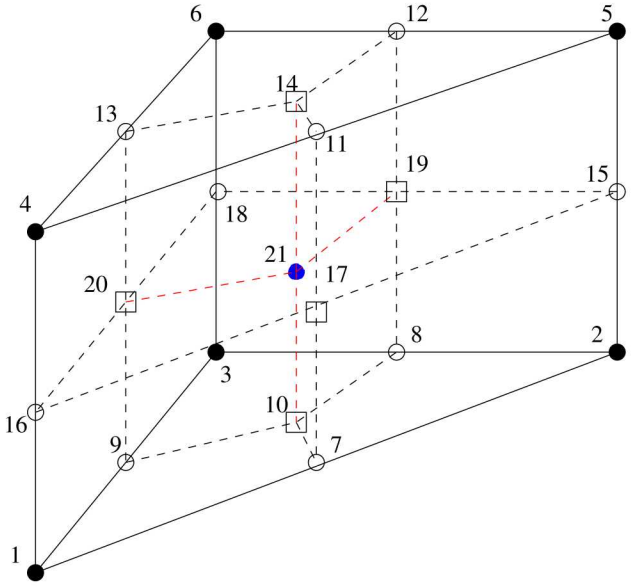


**Figure 5.8-8.. Wedge subcontrol points numbering**

**Table 5.8-20.. Subcontrol face definitions for exact surface area calculation on wedge elements.**

| Face | Point Set | | | |
|---|---|---|---|---|
| 1 | 7 | 10 | 21 | 17 |
| 2 | 8 | 10 | 21 | 19 |
| 3 | 10 | 9 | 20 | 21 |
| 4 | 11 | 17 | 21 | 14 |
| 5 | 14 | 12 | 19 | 21 |
| 6 | 13 | 14 | 21 | 20 |
| 7 | 16 | 17 | 21 | 20 |
| 8 | 17 | 15 | 19 | 21 |
| 9 | 20 | 21 | 19 | 18 |

**Table 5.8-21.. Subcontrol volume definitions for exact volume calculation on wedge elements.**

| Volume | Point Set | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 16 | 17 | 6 | 9 | 20 | 21 | 10 |
| 2 | 10 | 7 | 2 | 7 | 21 | 17 | 15 | 19 |
| 3 | 9 | 10 | 8 | 2 | 20 | 21 | 19 | 18 |
| 4 | 20 | 16 | 17 | 21 | 13 | 4 | 10 | 14 |
| 5 | 21 | 17 | 15 | 19 | 14 | 11 | 4 | 12 |
| 6 | 20 | 21 | 19 | 18 | 13 | 14 | 11 | 6 |

**Table 5.8-22.. Nodal shape functions and derivatives for pyramid elements. Range is (-1,1) and (0,1).**

| node | $N$ | $\frac{\partial N}{\partial \xi}$ | $\frac{\partial N}{\partial \eta}$ | $\frac{\partial N}{\partial \zeta}$ |
|---|---|---|---|---|
| 1 | $\frac{1}{4}(1-\xi)(1-\eta)(1-\zeta)$ | $-\frac{1}{4}(1-\eta)(1-\zeta)$ | $-\frac{1}{4}(1-\xi)(1-\zeta)$ | $-\frac{1}{4}(1-\xi)(1-\eta)$ |
| 2 | $\frac{1}{4}(1+\xi)(1-\eta)(1-\zeta)$ | $\frac{1}{4}(1-\eta)(1-\zeta)$ | $-\frac{1}{4}(1+\xi)(1-\zeta)$ | $-\frac{1}{4}(1+\xi)(1-\eta)$ |
| 3 | $\frac{1}{4}(1+\xi)(1+\eta)(1-\zeta)$ | $\frac{1}{4}(1+\eta)(1-\zeta)$ | $\frac{1}{4}(1+\xi)(1-\zeta)$ | $-\frac{1}{4}(1+\xi)(1+\eta)$ |
| 4 | $\frac{1}{4}(1-\xi)(1+\eta)(1-\zeta)$ | $-\frac{1}{4}(1+\eta)(1-\zeta)$ | $\frac{1}{4}(1-\xi)(1-\zeta)$ | $-\frac{1}{4}(1-\xi)(1+\eta)$ |
| 5 | $\zeta$ | 0 | 0 | 1 |

**Table 5.8-23.. Element variable values and differentials at control-volume faces for pyramid elements. Face-to-edge number mapping.**

| face | Edge ( $\mathrm{Node}_{\mathrm{out}} \rightarrow \mathrm{Node}_{\mathrm{in}}$ ) | $\xi$ | $\eta$ | $\zeta$ | $\mathrm{d}\eta\mathrm{d}\zeta$ | $\mathrm{d}\zeta\mathrm{d}\xi$ | $\mathrm{d}\xi\mathrm{d}\eta$ |
|---|---|---|---|---|---|---|---|
| 1 | $1 \rightarrow 2$ | $0$ | $-\frac{45}{104}$ | $\frac{7}{52}$ | | | |
| 2 | $2 \rightarrow 3$ | $\frac{45}{104}$ | $0$ | $\frac{7}{52}$ | | | |
| 3 | $3 \rightarrow 4$ | $0$ | $\frac{45}{104}$ | $\frac{7}{52}$ | | | |
| 4 | $1 \rightarrow 4$ | $-\frac{45}{104}$ | $0$ | $\frac{7}{52}$ | | | |
| 5 | $1 \rightarrow 5$ | $-\frac{7}{24}$ | $-\frac{7}{24}$ | $\frac{41}{120}$ | | | |
| 6 | $2 \rightarrow 5$ | $\frac{7}{24}$ | $-\frac{7}{24}$ | $\frac{41}{120}$ | | | |
| 7 | $3 \rightarrow 5$ | $\frac{7}{24}$ | $\frac{7}{24}$ | $\frac{41}{120}$ | | | |
| 8 | $4 \rightarrow 5$ | $-\frac{7}{24}$ | $\frac{7}{24}$ | $\frac{41}{120}$ | | | |

**Table 5.8-24.. Element variable values and differentials at subcontrol volume centers for pyramid elements.**

| sub-volume | $\xi$ | $\eta$ | $\zeta$ | $\mathrm{d}\xi\mathrm{d}\eta\mathrm{d}\zeta$ |
|---|---|---|---|---|
| 1 | $-\frac{19}{48}$ | $-\frac{19}{48}$ | $\frac{41}{240}$ | |
| 2 | $\frac{19}{48}$ | $-\frac{19}{48}$ | $\frac{41}{240}$ | |
| 3 | $\frac{19}{48}$ | $\frac{19}{48}$ | $\frac{41}{240}$ | |
| 4 | $-\frac{19}{48}$ | $\frac{19}{48}$ | $\frac{41}{240}$ | |
| 5 | $0$ | $0$ | $\frac{3}{5}$ | |

**Table 5.8-25.. Subcontrol face definitions for exact surface area calculation on pyramid elements.**

| Face | Point Set | | | |
|---|---|---|---|---|
| 1 | 6 | 10 | 19 | 13 |
| 2 | 7 | 10 | 19 | 15 |
| 3 | 8 | 10 | 19 | 17 |
| 4 | 9 | 18 | 19 | 10 |
| 5 | 12 | 13 | 19 | 18 |
| 6 | 11 | 15 | 19 | 13 |
| 7 | 14 | 17 | 19 | 15 |
| 8 | 16 | 18 | 19 | 17 |

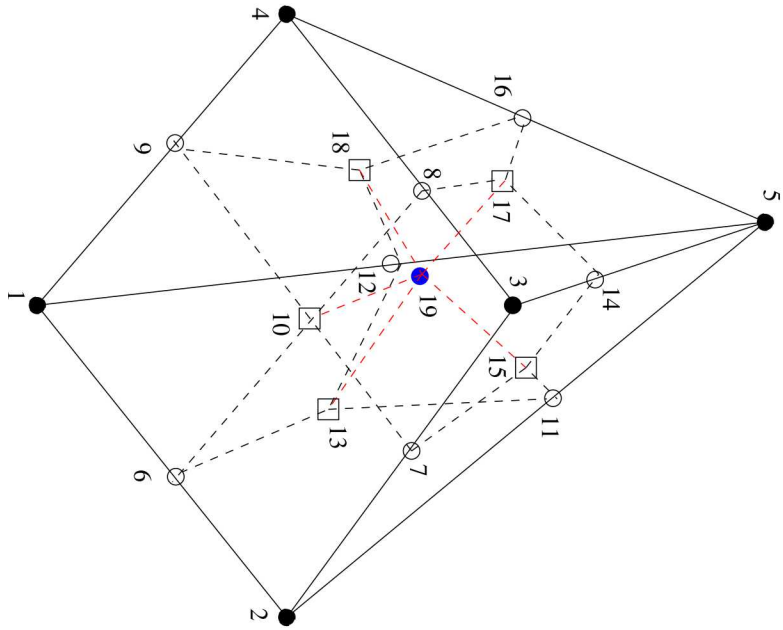


**Figure 5.8-9.. Pyramid subcontrol points numbering**

**Table 5.8-26.. Subcontrol volume definitions for exact volume calculation on pyramid elements.**

| Volume | Point Set | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 6 | 10 | 9 | 12 | 13 | 19 | 18 | | |
| 2 | 6 | 2 | 7 | 10 | 13 | 11 | 15 | 19 | | |
| 3 | 7 | 3 | 8 | 10 | 15 | 14 | 17 | 19 | | |
| 4 | 9 | 10 | 9 | 4 | 18 | 19 | 17 | 16 | | |
| 5 | 5 | 19 | 16 | 18 | 12 | 13 | 11 | 15 | 14 | 17 |

## 5.9. INTERPOLATION FUNCTIONS AND NEGATIVE COEFFICIENTS

A sufficient condition for a monotonic differencing scheme is that all the off-diagonal terms in the stencil be of opposite sign from the diagonal term [155]. Coefficient sets with mixed signs in the off-diagonal entries can potentially admit oscillatory solutions. In this section, the sign convention is that diagonal elements are negative and off-diagonal elements should be greater than or equal to zero. The term “negative coefficients" refers to one or more negative off-diagonal coefficients. Schemes with positive coefficients are usually considered important only when designing upwind convection operators, but they may be just as important for diffusion operators. Monotonic diffusion operators are most useful for artificial viscosity schemes and projection methods in application to the low Mach number Navier-Stokes equations. Positive coefficients are particularly important for the Poisson equation that arises when calculating a velocity correction to the continuity equation. The computed field for the velocity potential should be smooth so that no oscillations are introduced into the pressure field.

Mixed-sign off-diagonal coefficients commonly arise in finite-element-like methods for describing the diffusion operator. Christie and Hall [156] note that applying the Galerkin finite-element method (GFEM) with bilinear quadrilateral elements to harmonic functions sometimes results in negative coefficients. It was later discovered that there is a threshold element aspect ratio for positivity, and negative coefficients are produced on meshes of rectangular elements above that threshold value. Several authors note that the threshold aspect ratio for the quadrilateral element is $\sqrt{2}$ with GFEM and the value is $\sqrt{3}$ with the control volume finite-element method (CVFEM) [157, 158, 154]. The values for the aspect ratio limits only strictly apply to orthogonal structured meshes. Notably, the five-point difference stencil for the 2D finite-difference method never generates negative coefficients. By deduction, the integral formulas that use extra stencil points, introduced by the element-based methods, generate negative coefficients.

A word on oscillations is required before continuing. Smooth solutions are possible with negative coefficients. Finite-element and finite-volume analysis codes for diffusion processes, such as conduction heat transfer, may never experience oscillations. A forcing function is required to induce the oscillations, like a boundary layer with the convection-diffusion equation [159] or an ill-behaved source term in the continuity equation. The mass balance for a control volume is the source term in the projection method. If the mass balance from a time integration step is particularly bad, the projection scheme must smooth large errors. If there are negative coefficients for the velocity potential, then resulting velocity potential field may be non-smooth, which causes the pressure field to be non-smooth. The result is oscillations which grow into the solution and make for a non-robust solution process.

Causes of negative-coefficients are being studied in order to design robust solution algorithms for the Navier-Stokes equations. The numerical method of primary interest is the CVFEM, though results for the GFEM are included for comparison. The GFEM community describes negative-coefficient effects as “hour-glassing". The “hour-glass" oscillations are most common to reduced-integration formulations for the diffusion equation, and stabilization methods [160, 127] have been developed to damp the oscillations. In the CVFEM [143, 161], negative coefficients are prevented by shifting the integration points for the diffusion flux formulation out towards the edges of the control volumes and elements.

The method is termed "integration point shifting" in this section. There is no general way to control the coefficient signs when skewed quadrilateral elements are used with arbitrary connectivities. Coefficient control is not a panacea for negative coefficients since integration point shifting generally reduces the accuracy. Ultimately, the proper mesh will have no negative coefficients at all.

In this section, the numerics behind negative coefficients are discussed for the diffusion operator given by

$$\int \frac{\partial \phi}{\partial x_i} n_i \mathrm{d}S, \tag{5.391}$$

where the surface differential is defined by Eqns. 5.385. Integration point shift functions are derived for the CVFEM diffusion operator. Shift functions are presented for both two and three dimensions which guarantee coefficient positivity for a particular element aspect ratio. Also, the integration point shifting for CVFEM is shown to be similar to hour-glass stabilization for GFEM.

## 5.9.1. Positive Coefficients for Orthogonal Meshes

Negative coefficients arise in the off-diagonal coefficients when the aspect ratio of an element becomes large. Consider the elemental flux contributions to the control volume centered about Node 3, shown in Fig. 5.9-1. The first off-diagonal node to have a negative coefficient is the side node farthest from the control volume center, Node 4. The negative coefficient is associated with the vertical flux over the long horizontal face. At the integration point on the long horizontal face, the flux is approximated by an average of the difference between Nodes 3 and 2 and the difference between Nodes 4 and 1. The weighting between the two differences is determined by the location of the integration point. The negative coefficient is removed by removing the influence from the Node 4–1 difference. The integration point is shifted farther from Nodes 4 and 1, towards Nodes 2 and 3. The integration points are shifted such that the element-level coefficients are positive, a sufficient condition for global positivity.

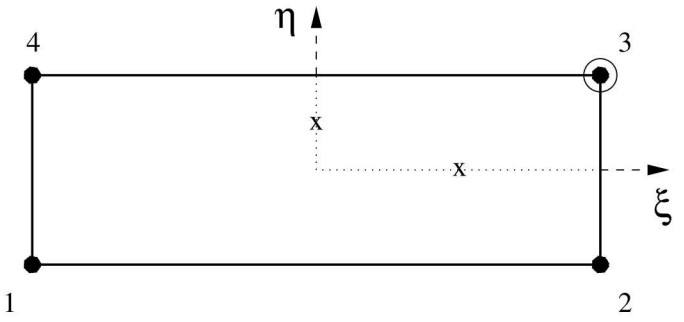


**Figure 5.9-1.. Control volume faces in a single element. Contributions to the the control volume centered about node 3.**

Integration point shift functions and the critical aspect ratio are derived for isoparametric quadrilateral elements with bilinear shape functions, and hexahedral elements with trilinear shape functions. Only the orthogonal form is considered. Linear triangles are discussed since they can also produce negative coefficients. For linear elements, only the element geometry (mesh quality) can be modified to control negative coefficients. With isoparametric bilinear and trilinear elements, both the geometry and the

location of the integration point control negativity. In addition, integration point shifting is compared to finite-element hour-glass control. Positive coefficients are achieved by either shifting the element integration points or applying the hour-glass stabilization matrix, and in some cases the two are identical.

#### 5.9.1.1. Aspect Ratio Definition

In this section, the isoparametric coordinates for an element are oriented such that the aspect ratio is greater than or equal to one. In two dimensions, the aspect ratio for an orthogonal element is the ratio between edge lengths. In three dimensions, each element has two aspect ratios since there are potentially three different edge lengths. The aspect ratios are taken relative to the shortest of the edge lengths which has a reference length of one.

#### 5.9.1.2. Quadrilateral Elements

The coefficients for the diffusion operator result from the combination of two basic second-order accurate diffusion operators: the edge operator and the centroid operator, shown in Fig. 5.9-2. The two operators represent the extremes in evaluating derivatives using the bilinear shape function within the element. The edge scheme always gives positive coefficients while the centroid scheme gives rise to negative coefficients above a certain aspect ratio. The centroid scheme results from evaluating all the derivatives for the four control volume sub-faces at the element centroid, equivalent to reduced integration [160] in GFEM. The edge scheme results from evaluating the derivatives out at the ends of the sub-faces in CVFEM or out at the nodes in GFEM. The edge scheme returns the standard five-point finite-difference operator. The traditional CVFEM [161] uses an equal weighting of the edge and the centroid scheme. The GFEM uses one part edge to two parts centroid. The GFEM is more prone to oscillations with high aspect ratio elements than the CVFEM because it contains a larger weighting of the centroid scheme. The single-point-integrated GFEM element will be the most unstable since it is a pure centroid scheme.

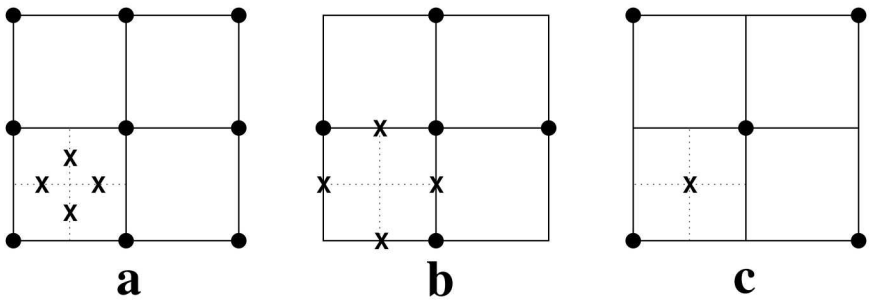


**Figure 5.9-2.. Flux integration points (X) determine nodal (•) contributions to the coefficient stencil: a) mid-face rule of CVFEM, b) edge–operator, c) centroid–operator (one-point integration).**

The coefficient signs for a rectangular element are controlled by moving the integration points away from the centroid of the element. The smallest value of the integration point shift that satisfies

coefficient positivity is found by symbolically integrating the diffusion flux over a control volume. Consider the diffusion operator evaluated over a collection of equal-size rectangular elements. Each element is longer by a factor of $\mathcal{AR}$ in the x-direction than the y-direction, where $\mathcal{AR}$ is the aspect ratio of the elements. The integration points can be shifted in the $\xi$-direction by $s$ and in the $\eta$-direction by $t$. The element coefficients that contribute to the equation centered at Node 3 in Fig. 5.9-1 are:

$$\begin{aligned}
\phi_1 &: \quad \frac{1 + \ \mathcal{AR}^2 - 2s\mathcal{AR}^2 - 2t}{8\mathcal{AR}} \\
\phi_2 &: \quad \frac{-1 + 3\mathcal{AR}^2 + 2s\mathcal{AR}^2 + 2t}{8\mathcal{AR}} \\
\phi_3 &: -\frac{3 + 3\mathcal{AR}^2 + 2s\mathcal{AR}^2 + 2t}{8\mathcal{AR}} \\
\phi_4 &: \quad \frac{3 - \ \mathcal{AR}^2 + 2s\mathcal{AR}^2 + 2t}{8\mathcal{AR}}
\end{aligned} \tag{5.392}$$

The positivity constraint for Node 3 comes from coefficient 4 in the element matrix, where the coefficient becomes negative for large values of aspect ratio. The $s$-shift removes the effect of aspect ratio, while the $t$-shift has no effect on negativity. For CVFEM, the integration points on vertical faces, in the longer x-direction, should be shifted from the mid-faces out towards the element edges by $s$. The y-direction flux is the only flux effected by the shift so the y-direction flux is the flux associated with negative coefficients. The minimal amount of $s$-shift required to maintain positivity depends upon the aspect ratio,

$$s > \frac{1}{2}\frac{\mathcal{AR}^2 - 3}{\mathcal{AR}^2}, \qquad \mathcal{AR} > \sqrt{3}. \tag{5.393}$$

The maximum aspect ratio for which the unshifted CVFEM remains monotone is $\sqrt{3}$. A similar formula exists for GFEM, where $s$ and $t$ are shifted from the Gauss points at $\pm 1/\sqrt{3}$. The element coefficients that contribute to the equation centered at Node 3 in Fig. 5.9-1 are:

$$\begin{aligned}
\phi_1 &: \frac{3 + 3\mathcal{AR}^2 - \left(1 + \sqrt{3}s\right)^2 \mathcal{AR}^2 - \left(1 + \sqrt{3}t\right)^2}{12\mathcal{AR}} \\
\phi_2 &: \frac{-3 + 3\mathcal{AR}^2 + \left(1 + \sqrt{3}s\right)^2 \mathcal{AR}^2 + \left(1 + \sqrt{3}t\right)^2}{12\mathcal{AR}} \\
\phi_3 &: \frac{-3 - 3\mathcal{AR}^2 - \left(1 + \sqrt{3}s\right)^2 \mathcal{AR}^2 - \left(1 + \sqrt{3}t\right)^2}{12\mathcal{AR}} \\
\phi_4 &: \frac{3 - 3\mathcal{AR}^2 + \left(1 + \sqrt{3}s\right)^2 \mathcal{AR}^2 + \left(1 + \sqrt{3}t\right)^2}{12\mathcal{AR}}
\end{aligned} \tag{5.394}$$

Similar to the CVFEM, the $s$-shift is the only shift that affects the aspect ratio term. The positivity constraint for Node 3 comes from coefficient 4 in the element matrix,

$$s > \sqrt{\frac{3\mathcal{AR}^2 - 4}{3\mathcal{AR}^2}} - \frac{1}{\sqrt{3}}, \qquad \mathcal{AR} > \sqrt{2}. \tag{5.395}$$

The maximum aspect ratio for which the unshifted GFEM remains monotone is $\sqrt{2}$.

The shift values are very sensitive to the aspect ratio, out to an aspect ratio of about four. The values of the integration point shift function are plotted as a function of the aspect ratio in Fig. 5.9-3. At that aspect ratio, the shifted integration points are near the edge of the element. Since the requisite integration point shift rapidly reaches the element edge for increasing aspect ratio, it can be argued that the maximum shift should always be taken. It will be shown in the section on accuracy that integration point shifting leads to a general loss in accuracy on non-orthogonal meshes. Therefore, it may be desirable to use Equation 5.393 to compute the minimal shift for each element. The accuracy consideration must be traded against the algorithmic complexity of computing geometry-dependent shape functions for each element.

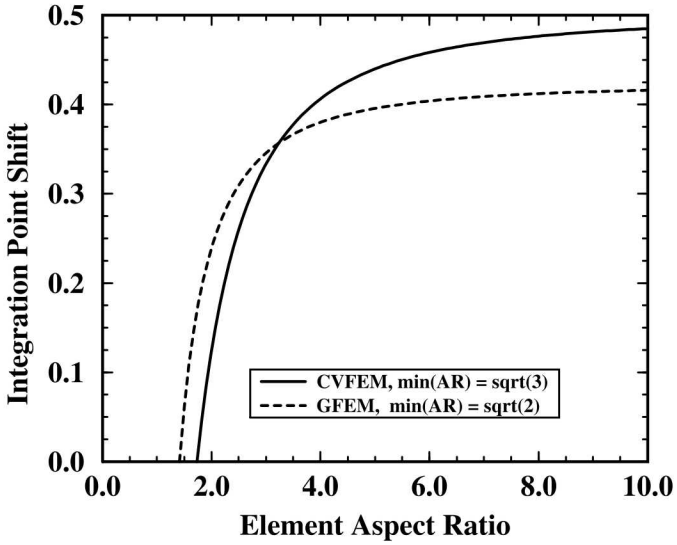


**Figure 5.9-3.. Integration point locations must be shifted out towards the element-edge with increasing element aspect ratio.**

### *5.9.1.3. Reduced Integration*

One-point integration methods for quadrilateral and hexahedral elements are popular because they are computationally efficient. Sometimes, oscillations occur and are called hour-glass modes after the displaced element shapes. Hour-glass stabilization methods prevent the hour-glass oscillations from occurring. The hour-glass terms are derived by examining the eigenmodes of the finite-elements and noting that there are missing mode shapes when the elements are integrated at the center [160]. The stabilization term adds the effects of the missing mode shapes back into the element formulation.

It is shown here that the hour-glass stabilization terms have the same effect on the element coefficients as shifting the integration points in two-dimensional elements. Both the hour-glass stabilization and the integration-point shifting modify the discretization to look more like a five-point scheme; or, more like the edge scheme of the previous section. The H-stabilization method is commonly used [127] in the GFEM with reduced integration. For quadrilateral elements, the element matrix for the hour-glass

stabilization term is

$$C_{\text{hg}} \begin{bmatrix} -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}, \tag{5.396}$$

where the constant $C_{\text{hg}}$ contains scaling information. The first row of the hour-glass stabilization matrix, Equation 5.396, can also be derived by subtracting the matrix coefficients for the one-point integrated GFEM, Equation 5.397,

$$\begin{array}{llll} \phi_1 : & \frac{1+\mathcal{AR}^2}{4\mathcal{AR}} & \phi_3 : & -\frac{1+\mathcal{AR}^2}{4\mathcal{AR}} \\ \phi_2 : & \frac{\mathcal{AR}^2-1}{4\mathcal{AR}} & \phi_4 : & \frac{1-\mathcal{AR}^2}{4\mathcal{AR}} \end{array} \tag{5.397}$$

from the coefficients for the five-point difference scheme, Equation 5.398,

$$\begin{array}{llll} \phi_1 : & 0 & \phi_3 : & -\frac{1+\mathcal{AR}^2}{2\mathcal{AR}} \\ \phi_2 : & \frac{\mathcal{AR}^2}{2\mathcal{AR}} & \phi_4 : & \frac{1}{2\mathcal{AR}} \end{array} \tag{5.398}$$

The resulting coefficient set is identical to the hour-glass stabilization matrix, Equation 5.396, if the multiplier $C_{\text{hg}} = (1 + \mathcal{AR}^2)/4\mathcal{AR}$. The hour-glass stabilization matrix is added to a diffusion operator to make it look more like a five-point finite difference scheme. Note that the multiplier used by GFEM practitioners [127] is $C_{\text{hg}} = 1$. The hour-glass stabilization matrix can also be used with other schemes. The multiplier for standard GFEM is $C_{\text{hg}} = (1 + \mathcal{AR}^2)/6\mathcal{AR}$. The multiplier for standard CVFEM is $C_{\text{hg}} = (1 + \mathcal{AR}^2)/8\mathcal{AR}$, though conservation is only guaranteed on rectangular meshes.

#### 5.9.1.4. Triangular Elements

Linear triangular elements can also produce negative off-diagonal coefficients. There are no shift-functions for triangles since the gradients are constant over the element. Negative coefficients result from the geometry of the element.

Nodal coefficients for a triangular element are computed for the diffusion flux contributions, shown in Fig. 5.9-4. The nodal coefficients for the diffusion flux contribution to the control volume centered about Node 1 are

$$\begin{array}{ll} \phi_1 : & -\frac{1}{2}\frac{\tan\alpha}{\tan\beta}\frac{(1+\tan^2\beta)}{(\tan\alpha+\tan\beta)} \\ \phi_2 : & \frac{1}{2}\frac{(\tan\alpha\tan\beta-1)}{(\tan\alpha+\tan\beta)} \\ \phi_3 : & \frac{1}{2}\frac{1}{\tan\beta} \end{array} \tag{5.399}$$

where the base edge length is $r$ and the two adjacent vertex angles are $\alpha$ and $\beta$. The conditions required to ensure that all the off-diagonal terms remain positive are combined from constraints on all three control volume contributions,

$$\begin{aligned} \tan\alpha &> 0 \\ \tan\beta &> 0 \\ \tan\alpha\tan\beta &> 1. \end{aligned} \tag{5.400}$$

The triangular element yields positive off-diagonal coefficients if $0 < \alpha < \pi/2, 0 < \beta < \pi/2$, and $\pi/2 < \alpha + \beta < \pi$. The triangle must be acute.

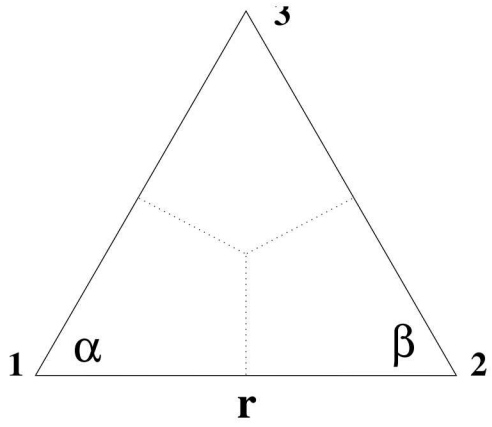


**Figure 5.9-4.. Triangular element geometry in defined by edge length and two vertex angles.**

In a previous work [162], quadrilateral elements were subdivided into triangular elements using edge-swapping, along with a Delaunay algorithm, to minimize the effect of negative coefficients. Given a collection of nodes, a Delaunay triangulation of the nodes will generate triangles where the minimum angle between vertices is maximized, leading to near-equilateral triangles that satisfy Equation 5.400.

#### *5.9.1.5. Hex Elements*

The aspect ratio limit for the CVFEM diffusion operator with orthogonal, hexahedral elements can be as large as $\sqrt{2}$ if the base is square. The three-dimensional element is more difficult to control because there are two aspect ratios. The local element node numbering for a hexahedron is defined in Fig. 5.8-4. Let the edge length between Nodes 1 and 2 be of value $A$, the edge length between Nodes 1 and 4 be of value $B$, and the edge length between Nodes 1 and 5 be of value $C$. These are the $\xi$, $\eta$, and $\zeta$ directions. The parametric representation of the nodal coefficients for the element contribution to the control

volume centered about Node 1 is

$$\begin{aligned}
\phi_1: &\quad -\frac{9\,(\ \ A^2B^2+\ \ B^2C^2+\ \ C^2A^2)}{64ABC}\\
\phi_2: &\quad \frac{3\,(-A^2B^2+3B^2C^2-\ \ C^2A^2)}{64ABC}\\
\phi_3: &\quad \frac{3\,(-A^2B^2+3B^2C^2+3C^2A^2)}{64ABC}\\
\phi_4: &\quad \frac{3\,(-A^2B^2-\ \ B^2C^2+3C^2A^2)}{64ABC}\\
\phi_5: &\quad \frac{(3A^2B^2-\ \ B^2C^2-\ \ C^2A^2)}{64ABC}\\
\phi_6: &\quad \frac{(3A^2B^2+3B^2C^2-\ \ C^2A^2)}{64ABC}\\
\phi_7: &\quad \frac{(\ \ A^2B^2+\ \ B^2C^2+\ \ C^2A^2)}{64ABC}\\
\phi_8: &\quad \frac{(3A^2B^2-\ \ B^2C^2+3C^2A^2)}{64ABC}.
\end{aligned} \tag{5.401}$$

The region for positive coefficients is plotted in Fig. 5.9-5 which comes from examining coefficients for Nodes 2, 4, and 5. The maximum allowable aspect ratios occur for the case of a square base. If the base edges are longer than the vertical edge, then the maximum aspect ratio is $\sqrt{2}$. If the vertical edge is longer than a base edge, the maximum aspect ratio is $\sqrt{3/2}$.

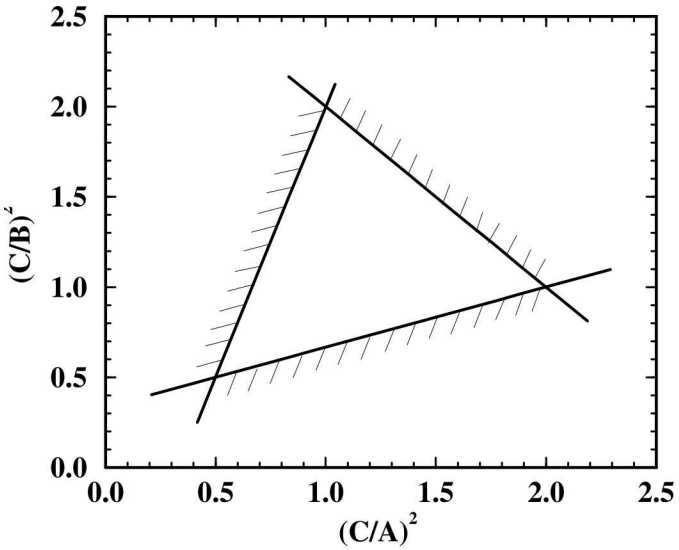


**Figure 5.9-5.. Limits of Edge-Length Ratio for Positive Coefficients in 3D CVFEM.**

The integration points in the CVFEM scheme can be shifted by $s$, $t$, and $u$ in the $\xi$, $\eta$, and $\zeta$ directions.

The conditions for positive coefficients are

$$\begin{array}{rcl} \dfrac{1}{A^2}\dfrac{1}{1-2s} - \dfrac{1}{B^2}\dfrac{1}{3+2t} & > & \dfrac{1}{C^2}\dfrac{1}{3+2u} \\ \dfrac{1}{B^2}\dfrac{1}{1-2t} - \dfrac{1}{C^2}\dfrac{1}{3+2u} & > & \dfrac{1}{A^2}\dfrac{1}{3+2s} \\ \dfrac{1}{C^2}\dfrac{1}{1-2u} - \dfrac{1}{A^2}\dfrac{1}{3+2s} & > & \dfrac{1}{B^2}\dfrac{1}{3+2t} \end{array} \tag{5.402}$$

The relations are nonlinear and require iteration to extract the limiting values of $s$, $t$, and $u$.

The coefficients that are generated by the standard GFEM operator in three dimensions have no allowable maximum aspect ratio. The only element shape that does not have negative coefficients is a cube, and even then the coefficients of the six nearest nodes are zero. The parametric representation of the nodal coefficients for the element contribution to the equation associated with Node 1 are

$$\begin{array}{ll} \phi_1: & -\dfrac{4\,(\ \ A^2B^2+\ \ B^2C^2+\ \ C^2A^2)}{36ABC} \\ \phi_2: & \dfrac{2\,(-A^2B^2+2B^2C^2-\ \ C^2A^2)}{36ABC} \\ \phi_3: & \dfrac{(-A^2B^2+2B^2C^2+2C^2A^2)}{36ABC} \\ \phi_4: & \dfrac{2\,(-A^2B^2-\ \ B^2C^2+2C^2A^2)}{36ABC} \\ \phi_5: & \dfrac{2\,(2A^2B^2-\ \ B^2C^2-\ \ C^2A^2)}{36ABC} \\ \phi_6: & \dfrac{(2A^2B^2+2B^2C^2-\ \ C^2A^2)}{36ABC} \\ \phi_7: & \dfrac{(\ \ A^2B^2+\ \ B^2C^2+\ \ C^2A^2)}{36ABC} \\ \phi_8: & \dfrac{(2A^2B^2-\ \ B^2C^2+2C^2A^2)}{36ABC}. \end{array} \tag{5.403}$$

The contributions from Nodes 2, 4, and 5 are such that there will be negative coefficients for any element shape other than a cube.

Single-point integration is also used for the GFEM hexahedral element. Unfortunately, there is not a clear analogy between the integration point shift and hour-glass stabilization in three dimensions. The element matrix for the hour-glass stabilization term is

$$C_{\text{hg}}\begin{bmatrix} -4 & 2 & 0 & 2 & 2 & 0 & -2 & 0 \\ 2 & -4 & 2 & 0 & 0 & 2 & 0 & -2 \\ 0 & 2 & -4 & 2 & -2 & 0 & 2 & 0 \\ 2 & 0 & 2 & -4 & 0 & -2 & 0 & 2 \\ 2 & 0 & -2 & 0 & -4 & 2 & 0 & 2 \\ 0 & 2 & 0 & -2 & 2 & -4 & 2 & 0 \\ -2 & 0 & 2 & 0 & 0 & 2 & -4 & 2 \\ 0 & -2 & 0 & 2 & 2 & 0 & 2 & -4 \end{bmatrix}, \tag{5.404}$$

where the constant $C_{\text{hg}}$ contains scaling information. The hour-glass stabilization matrix does act to make the diagonal terms more negative, and the problematic off-diagonal terms more positive.

# 5.10. H-ADAPTIVITY MESHING

Note: we currently only allow uniform refinement with no load balancing (12/01). We have not yet decided on a scheme for integrating fluxes over h-refined meshes. We have not yet decided on a prolongation approach for the mass flow rate at faces.

## 5.10.1. H-Adaptivity and Flux Construction

The equation assembly in our control volume method is based on integrating fluxes over control volume sub-faces within an element. A typical h-adapted patch of elements is shown in Figure 5.10-1. The "hanging nodes" do not have control volumes associated with them. Rather, they are constrained to be a linear combination of the two parent edge nodes. There is no element assembly procedure to compute fluxes for the "handing sub-faces" associated with the hanging nodes that occur along the parent-child element boundary.

One possibility is to create a sub-set of element faces that contain hanging-nodes. The fluxes across the hanging sub-faces can then be processed using local nodal information. This precludes computing localized gradients across the face.

The SIERRA h-adaptive scheme is driven at the element level. Refinement occurs within the element and the topology of refined elements is the same as the parent element. If the topology restriction was relaxed, then the following schemes could be used.

Aftosmis [163] describes a vertex-centered finite-volume scheme on unstructured Cartesian meshes. A transitional set of control volumes are formed about the hanging nodes, shown in Figure 5.10-2. on unstructured meshes. (This would require a series of specialized master elements to deal with the different transition possibilities in SIERRA and would be a burden on the application teams.)

Kallinderis [164] describes a vertex-centered finite-volume scheme on unstructured quad meshes. Hanging nodes are treated with a constraint condition. The flux construction for a node on a refinement boundary is based on the unrefined parent elements, leading to a non-conservative scheme.

Kallinderis [165] describes a vertex-centered finite-volume scheme on unstructured tetrahedral meshes. Hanging nodes are removed by splitting the elements on the "unrefined" side of the refinement boundary. Mavriplis [166] uses a similar technique, but extends it to a general set of heterogeneous elements, shown in Figure 5.10-3. (This would require a change to the topology rules in SIERRA as well as splitting elements along the refinement boundary, but there would be little impact on the application codes other than supporting heterogeneous meshes.)

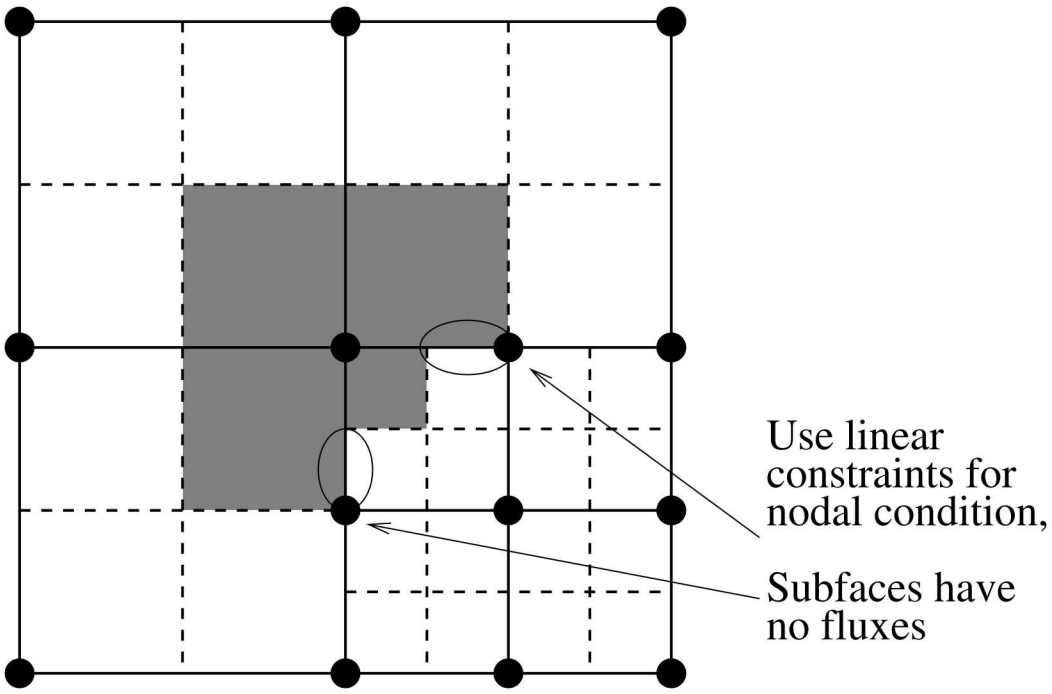


**Figure 5.10-1.. Control volume definition on an h-adapted mesh with hanging nodes. (Four-patch of parent elements with refinement in bottom-right element.)**

#### *5.10.1.1. Prolongation and Restriction*

Nodal variables are interpolated between levels of the h-adapted mesh hierarchy using the traditional prolongation and restriction operators defined over an element. The prolongation operation is used to compute values for new nodes that arise from element sub-division. The parent element shape functions are used to interpolate values from the parent nodes to the sub-divided nodes.

Prolongation and restriction operators for element variables and face variables are required to maintain mass flow rates that satisfy continuity.

#### *5.10.1.2. Mass Continuity*

Care must be taken to ensure continuity of mass between control volumes that contain hanging sub-faces. Especially since control-volume balances at hanging nodes are replaced by constraint conditions.

We need a list of the hanging faces as well as a means of identifying the hanging nodes on each face.

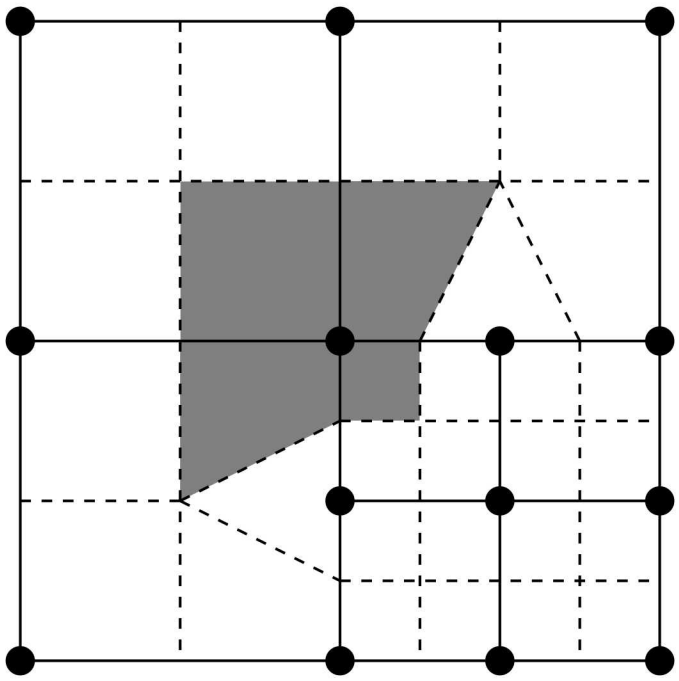

**Figure 5.10-2.. Control volume definition on an h-adapted mesh with transition control volumes about the hanging nodes. (Four-patch of parent elements with refinement in bottom-right element.)**

#### *5.10.1.3. Nodal Gradients*

The nodal gradients are approximated by integrating over the surface of the control volume and applying the discrete form of the Gauss divergence theorem. There are two possible approaches for dealing with the hanging sub-faces. In the first approach, the hanging sub-faces are processed separately. In the second approach, the sub-faces are ignored but the unclosed surface integral is corrected by a reference value, namely the nodal value associated with the control volume centroid,

$$\left.\frac{\partial \phi}{\partial x_i}\right|_p \mathrm{d}V = \int \left(\phi - \phi_p\right) n_i \mathrm{d}S \tag{5.405}$$

### 5.10.2. Dynamic Load-Balancing

Dynamic load-balancing is required as the mesh is adaptively refined across parallel processors. Some processors may end up with more refined elements, so the work load increases. We will use the Zoltan dynamics load-balancing package to drive the load-balancing. We need a good measure of the compute load, most likely a combination of the time to assemble equations and the solve them.

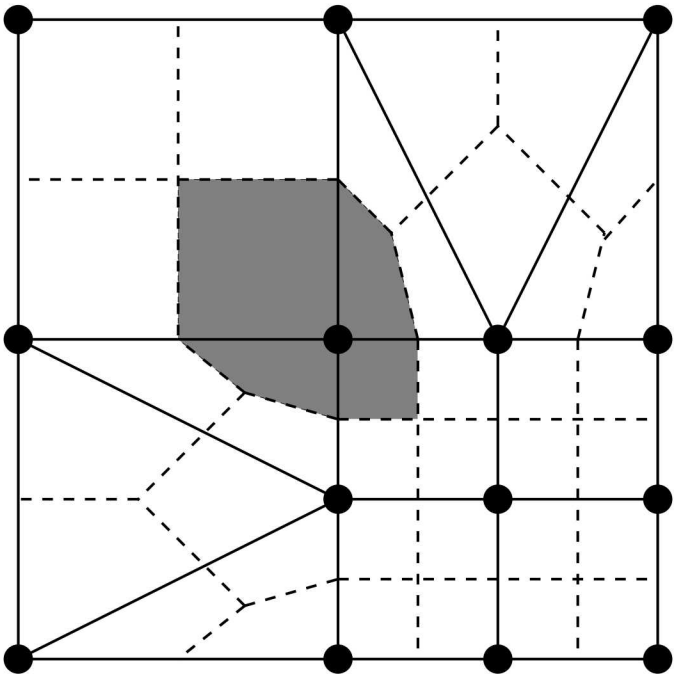

**Figure 5.10-3.. Control volume definition on a heterogeneous h-adapted mesh with no hanging nodes. (Four-patch of parent elements with refinement in bottom-right element and splitting in adjacent parent elements.)**

# 6. IMPLEMENTATION

This is a software implementation description for the ASC application code Fuego. The Fuego code is part of the multi-mechanics suite of codes built upon the SIERRA Frameworks. The SIERRA Frameworks are designed primarily for unstructured, finite-element mechanics codes. The Fuego code is based on a finite-volume method. Finite-volume framework requirements that differ from traditional finite-element frameworks are defined throughout the document. The limitation to vertex-centered finite volume schemes is particular challenging for the implementation of turbulence model wall functions and h-adaptive meshing.

The SIERRA Frameworks [167] provide a hierarchy for describing a mechanics code or a multi-mechanics code. At the top level is the *domain* which contains all the support infrastructure for the code. Within the *domain* is the *procedure* which manages time integration and the exchange of data between any multi-mechanics components. Within the *procedure* can be multiple *regions*. A *region* contains a description of some particular physics. Within a *region* is a collection of *mechanics* which can either be the math models that describe the physics or part of a solution algorithm.

The bulk of the Fuego code exists at the region level and below. The region-level design philosophy for the Fuego code is based on a core continuity/momentum transport capability with a configurable set of transport math models. The *sub-mechanics* within the *region* define the collection of transport equations that describe the physics. In this sense, Fuego itself is capable of supporting multi-mechanics within its own context because it can generate multiple regions, each with a different collection of transport equations.

We use a finite-volume scheme for the discrete form of the Fuego math models, derived from the control-volume, finite-element methods (CVFEM). The most significant difference between our CVFEM implementation in SIERRA and a FEM implementation is in the application of boundary conditions. Most of the boundary conditions for CVFEM are applied as fluxes. The fluxes over an element face are constructed from all the information in the parent element. The fluxes are linearized such that there are both matrix and right-hand-side contributions.

## 6.1. SIERRA FRAMEWORKS

The Fuego code is built upon the SIERRA Frameworks. The SIERRA Frameworks are written in C++ and make extensive use of standard template library (STL) container classes. A good understanding of STL is useful in understanding how to use the SIERRA Frameworks and access data within the Frameworks. Much of the code design documentation is scattered throughout in-source comment-lines and the product design documents (PDD) that accompany each source code check-in to the version control repository. A description of SIERRA Frameworks functionality is contained in the SIERRA

requirements document [168]. The Frameworks theory and design are described in the design document [167]. The following taxonomy describes some of the key mechanisms of the SIERRA Frameworks:

*procedure* The *procedure* support class contains methods for manipulating the *procedure* object. The *procedure* controls the time integration process and the exchange of information between *regions* via a *transfer*. Multi-mechanics code coupling usually occurs within the *procedure* between different *regions*.

*region* The *region* support class contains methods for manipulating the *region* object. The *region* contains the description of the math models and the solution procedures for advancing a time step. Most of the code that makes an application code unique is contained within the *region*. The *regions* are designed to have no direct dependency on *procedure* code or code from other *regions*. All external data is loaded into local *control data* or loaded via *transfers*.

*transfer* The *transfer* object is invoked within a *procedure* in order to move data between *regions*. Each *region* has its own mesh even though they may fill the same physical space. The *transfer* object manages the interpolation of data between the meshes.

*mechanics* A *mechanics* is a generic object within the Frameworks and contains methods for operating on itself or other *mechanics* objects. It may invoke workset algorithms to efficiently process data.

*instance* An *instance* is a member of a list of a *mechanics* object. An instance is typically unique in its association with a *mesh object*.

*context* A *context* is a label that is applied to a collection of objects.

*extent* An *extent* is the collection of objects that have the same *context*.

*iterator* An *iterator* is a method of looping through a list of objects.

*mesh object* A *mesh object* is part of mesh; i.e., an element block, side-set, or node-set.

*workset* The Frameworks uses a caching strategy to process floating-point information. The Frameworks processes the governing equations associated with the math models on an element-by-element basis. A *workset* is a collection of elements that are processed at one time such that all the local data required for the evaluation will fit in cache. Heterogeneous *worksets* are used for boundary condition flux processing. A heterogeneous *workset* is defined by an iterate such as a collection of element faces in a side-set, but processes data based on the parent topology such as the parent elements.

*workset algorithm* The method for processing a *workset* is a *workset algorithm*. Local variables are registered with the algorithm that have associativity with global data. The Frameworks manages the transfer of data to and from the *workset algorithm* via assembles, scatters, and gathers.

*solver* A *solver* object is responsible for assembling a linear system, applying boundary conditions and constraints, and solving the linear system. This object is the interface to the linear solver packages.

*library* A *library* is a method of storing lists of data.

*control data* The *control data* is a dynamic list of integer, real, and string data that is accessed by string labels.

*master element* A *master element* defines the topology of an elements and provides methods for specifying integration locations, performing interpolations, and processing geometry (areas, volumes, and gradients).

*parser* The *parser* system is a method for transferring information from a formatted text input file to the application code. The SIERRA Frameworks supplies a parsed input system. There are three parts to the parsing system: 1) a database of commands using XML, 2) call-backs that are provided by the application code to take action on a line command, and 3) code for linking the XML commands to the call-backs.

The Frameworks objects that form the foundations of an application code are created during the parsing phase.

## 6.2. FUEGO FRAMEWORKS

The SIERRA/Fuego code is a collection of C++ and FORTRAN code. The routines written in C++ contain the frameworks–type operations such as solution algorithms, data management, and variable registration. The routines written in FORTRAN contain the element and boundary condition routines that describe the math models. Some of the Fuego frameworks source code files are listed in Table 6.2-1 with their functionality. These files contain one or more subroutines or functions.

The matrix assembly and linear solve procedures are managed by the SIERRA Frameworks solver objects. The “support" classes listed above for element routines and boundary condition routines register themselves with the solver in the parsing phase. The solver then calls its registered methods to assemble the matrix.

The code is assigned a version number consisting of three digit fields, separated by two periods, X.Y.Z. The first digit field (leftmost) is the major number. The major number will be 0 during development and will increment to 1 upon the official release. The second field is the minor number and represents significant jumps in capability. During development, the minor number will increment with each code stage. The third digit field (rightmost) is the patch level and represents minor modifications and bug-fixes. Changes in input syntax also force a patch level increment. The initial version numbering schedule is shown below.

**0.1.0** - Stage One, PUVW, laminar convection, isothermal, uniform

**0.2.0** - Stage Two, PUVWT, laminar convection, thermal, uniform

**0.3.0** - Stage Three, PUVWKE, turbulent convection, isothermal, uniform

**0.4.0** - Stage Four, PUVWKEHY, turbulent convection, thermal, nonuniform

**0.5.0** - Stage Five, PUVWKEHY, add EDC model for turbulent combustion

**0.6.0** - Stage Six, PUVWKEHYS, soot and fire

Version-number matching is enforced for the input file. The code will abort if the requested version number does not match the current internal code version number.

The default repository configuration of Fuego is to have code-coupling turned on. The code-coupling feature is activated in the repository in order to run the nightly regression tests for coupled-mechanics.

### 6.2.1. Framework Control Data

The control data are dynamic lists of integers, reals, and strings, which are used to store data. In Fuego, the control data is used to store solution parameters and user-defined constants. There is control data that exists at different scopes within the SIERRA Frameworks. There is control data for the procedure, the region, instances of an element mechanics, and instances of a generic mechanics.

A control datum is referenced by a string label. The string-matching is case-sensitive. The Fuego code will use all-capital letters for control data labels. The control data is registered dynamically in the source code, usually during the parsing phase.

The procedure control data contains information about the time integration process, shown in Table 6.2-2. For the purpose of mechanics code-coupling, it should be as mechanics-generic as possible.

The Fuego region control data contains information relevant to the solution procedure for the fire physics math models, shown in Table 6.2-3. The region control data is also a means for passing information back to the calling procedure.

The boundary condition mechanics instance control data, shown in Table 6.2-5, is used to hold information for the boundary conditions. The boundary conditions are implemented as generic mechanics objects. When the boundary condition needs specified values, use either a CONSTANT value, a FUNCTION, or a user-supplied SUBROUTINE. The “function name" must match an entry in the FUNCTION library. The “subroutine name" must match a valid subroutine that has been linked to the code.

### 6.2.2. Framework Procedure

The procedure code manages the time integration and the exchange of data between regions. The procedure object is based on the Afgo_Procedure class (see Figure 6.2-1), which is derived from a Fmwk_Procedure class. The Afgo_Procedure class contains additional timing information. The procedure supports a fixed time step and a variable time step that is set according to a fixed CFL number. The time step control information is defined in the time-control input block. Within the time-control block are multiple time-step blocks. The time step block defines valid time step control parameters for a period of time. The time-step blocks must be contiguous in time. The Fuego code does not use the time-control block TERMINATION time for a stopping criterion. The start and stop times are defined by commands within the procedure.

Two methods are required for the Afgo_Procedure_Support class – initialize() and execute(). The initialize() method registers control data, initializes the time and step size, calls the initialize() method

for each region, initializes the transfer objects if there are coupled mechanics, sets up the region evaluation ordering if there are multiple regions, and initializes all material properties with STATE association. The SIERRA Frameworks will generate a default region iterator where the ordering is alphabetical by name. The regions should be called in a specific order, so the region type list defines the evaluation ordering. There is a region list that defines the processing order of the different regions of the same type. These lists are stored in the procedure control data.

The execute() method performs the time integration. Fuego only supports a transient solution procedure. The transient solver performs nonlinear iteration over regions within a time step until that time step is declared converged, and then advances to the next time step. Within a nonlinear iteration, each region is processed. For each region, there is first an optional pre-nonlinear processing step that usually involves loading region control data. The region execute() method is then called to solve the equations. An optional post-nonlinear processing step is taken for the region, usually consisting of a data transfer. After the nonlinear iteration is finished, the solution variables are advanced and the time step process starts over.

The time-advancement method provided by SIERRA rotates the states of the state variables by swapping pointers. In our solution strategy, the solution in the (N+1) state is always used as an initial guess in the nonlinear solution procedure. When the states are swapped, the initial solution in the (N+1) slot is actually the solution from the (N) slot of the previous time step which is now the (N-1) solution. In order to get the best initial guess, the Fuego code provides additional methods to copy the solution in the new (N) state forward to the new (N+1) state for all nodal and element state variables. The state manipulation is actually called out of the region code since this requirement is particular to the Fuego mechanics. The state copy is only performed for the first subiteration within the time step (remember there is subiteration over regions within a time step and subiteration over equation sets within a region).

### 6.2.3. Framework Region

The region code manages the nonlinear solve of the nonlinear equations describing a sub-mechanics (see Figure 6.2-2). The matrix assembly, linear-solve, and scatter operations are handled entirely by the Fmwk_LinearSolver class and the finite element interface (FEI). Each equation set in the fire mechanics has an associated linear_solver. The linear_solver is told, in the parser registration phase, which workset algorithm to use to build matrix contributions.

A region object is based on the Afgo_Region class, derived from the Fmwk_Region class. The Afgo_Region contains the material property_evaluator object Afgo_Material and references to all the nodal data. The region code requires two methods – initialize() and execute(). The linear solver processing order is defined in initialize(). The linear solvers are labeled according to the equation set they are solving: “solve_p", “solve_u", “solve_v", “solve_w", “solve_k", “solve_e", “solve_t", “solve_h", “solve_y", “solve_n", and “solve_s". The initialize() object then calls the initialize object for all the element mechanics instances. The material property_evaluator objects are then initialized.

The execute() method manages the nonlinear iterative solve. Just as the procedure code will perform iterations over the region within a time step, the region code calls for a specified number of iterations over the equation sets within the region. On the first time through the region code, for each time step at

the procedure level, the state management routines are invoked. Material properties are evaluated by the property_evaluator object at the beginning of each nonlinear iteration within the region. The solver list is iterated upon to loop through the equation set solves. The nodal pressure gradient contribution to the momentum equations is added by the setRHS() method. The interior equations and flux boundary conditions are assembled using the loadBlock() method of the linear_solver. The nodal boundary conditions are then applied through a call to loadBC(). The equations are solved and the solution is scattered back to the global nodal arrays. We use the delta-form of the linear system, so additional state management routines are required to copy the solution "delta" into the solution variable.

When there are multiple species transport equations, an additional temporary array is used to act as a solution array with the linear solver. Special data management routines are used to locally gather and scatter mass fractions from the global "mass_fraction" array to the local "ysolve" array.

When solving the transport equations for turbulent kinetic energy and turbulence dissipation, the updating of the turbulent kinetic energy is lagged until both equations have been solved. This allows for a constant value of $k/\epsilon$ in the source terms of both equations.

There is only one material defined for a region. The material may be a multicomponent gas.

### 6.2.4. Element Mechanics

Element mechanics are a special class of mechanics that know how to loop over elements (see Figure 6.2-3) by association with element topology through a master element. Elements are processed by worksets, so the mechanics is described by a workset algorithm. A workset is a collection of elements that can be processed (assembled) while remaining in cache memory. All element mechanics workset are currently (12/01) hard-wired for hexahedral elements until the convection operator routines can be generalized.

The transport equations can be configured for a particular collection of physics. The collection of transport equations is defined by the sub-mechanics of the problem. The sub-mechanics labels are described in Table 6.2-6.

The SIERRA Frameworks code processes the workset algorithm, loading the workset variables from global variables, scattering workset variables back to global variables, assembling the matrix and right-hand side, and assembling other global variables from workset variables. The data management routines for the workset algorithms are all contained in Afgo_ElemMech_Support.C. The names and descriptions of the workset algorithms are given in Table 6.2-7. The laminar form of the equations are separate from the turbulent form.

### 6.2.5. Boundary Conditions

We primarily use flux boundary conditions, but also support fixed (Dirichlet) boundary conditions. Flux boundary conditions make use of the element information adjacent to the boundary face, generating an element matrix contribution. The flux boundary conditions are implemented as heterogeneous workset algorithms. There are four flux boundary condition classes:

Afgo_InflowBC_Support, Afgo_OutflowBC_Support, Afgo_SymmetryBC_Support, and Afgo_WallBC_Support. Each class contains all the workset algorithms needed to evaluate flux boundary conditions for each of the transport equations. All flux boundary condition routines are currently (12/01) hard-wired for quadrilateral faces with hexahedral parent elements.

The flux boundary condition worksets are not registered until after all the nodal variables have been created. The flux boundary condition classes have mechanics algorithms for the registration phase and workset algorithms for the assembly phase.

The fixed boundary conditions, shown in Table 6.2-8, are derived off of the nodal_contribution() class within the Fmwk_LinearSolver class. There is an Afgo_DirichletBC class to encapsulate the boiler-plate methods.

Each boundary condition mechanics can have several "instances". Each instance shares the same specified data, but is mapped to a unique collection of side sets and node sets. Each instance has its own control data.

The data specific to a boundary condition is obtained through the boundary condition instance control data. Specified values, such as velocity or pressure or temperature, can be either constant, a piecewise linear function of one of the coordinate axes, or derived from a user-supplied subroutine.

### 6.2.6. Material

Material properties in Fuego are evaluated using a property_evaluator software object. All properties are computed as nodal variables. Nodal values are interpolated to sub-faces and sub-volumes. The material properties are evaluated once outside of the equation-set loop in the region code. The material_evaluator evaluates properties by list. A list of property names is defined during the initialization that defines all the properties required to evaluate the equations.

Raw property data is stored in the MATERIAL_PROPERTIES and FUNCTIONS libraries, one entry for each type of material requested. Currently (11/30/99), only two means of specifying material properties is supported: specified functions and Chemkin calls. The specified function properties are defined by the user in the input file. Constant values are defined by a constant function. The second means of defining properties is through Chemkin. A modified form of the Chemkin linking files will be placed in the material library entry (see table).

The material property_evaluator object is located in an Afgo_Region object. There is only one material evaluator per region. Material properties for thermal and/or nonuniform flows must be evaluated using the Chemkin libraries [169, 170]. A modified version of the Chemkin libraries is installed in the SIERRA system. The Chemkin FORTRAN routines have been modified such that there are no common blocks, so the API of some subroutines has changed. Three files need to be present in order to run Fuego – the Chemkin input file, the ASCII Chemkin linking file, and the ASCII transport linking file.

### 6.2.7. Master Elements

The master element classes contain the topological information required to define interpolations, integrations, and geometry processing (areas, volumes, and gradients) within an element. The master elements specific to the element-assembled CVFEM used in the Fuego code are shown in Table 6.2-9. The fluid flow sub-mechanics currently (12/01) only make use of the hexahedral elements. The heat conduction sub-mechanics make use of all CVFEM master elements.

### 6.2.8. User Subroutines

The FUEGO code allows subroutines to be defined and used to set quantities in the code such and boundary conditions (inflow profiles), transport terms, and initial conditions. The only restriction on the subroutine being called is its signature or parameter list. Defined signatures are listed in Table 6.2-10

The subroutine must be dealt with in parsing, initialization, and workset areas of the code. In every case, the relevant parsing callbacks are in the "register_commands" member function associated with the input class.

With respect to initialization, the boundary condition classes are the most involved. Here, each mechanics instance has its own user subroutine and associated user constants. When the parsing triggers a subroutine callback, the appropriate flag is set and the subroutine name is stored. The handler then stores the subroutine name and any constants associated with it in the instance's control data. There is no other initialization to be done for subroutines, although it is important to guard any function (load curve) calls (or function initialization) with a check on the "type" of condition being set (constant, load curve, or subroutine).

In the workset portion of the boundary condition codes, the variable in question (associated with a subroutine) will need to be calculated. At this stage, the variable "type" within the code gets checked; if it's a subroutine type, the code takes its name from the instance control data. At that point, the subroutine pointer itself can be retrieved from the framework registry, its signature checked, and a call made to load the relevant boundary data.

In the transport equations, things are a bit less complicated: the region control data contains the appropriate subroutine associations. However, it could have one for each transport variable. Parsing callbacks are again defined in the "register_commands" member function. In this case, the parsing callback looks at the variable indicated by the parsing (eg. a source term for pressure) and creates an association within the region control data for that particular variable and the given subroutine name. As with the boundary condition implementation, there is precious little to do to initialize a user subroutine. The only task the callback has is to register the subroutine name in the region's control data for later use.

In the workset portion of the code, the source term subroutines are called immediately after the FORTRAN element routine. If the appropriate flags are set, the subroutine name is retrieved from the region's control data, the subroutine pointer is retrieved from the framework registrar, and the signature checked. At this stage, the subroutine can be called. The subroutine takes from the user source terms for

the right hand side and the diagonal entries on the left hand side of the discretized algebraic equations.

Like the source term implementations, initial condition user subroutines are less complicated than the boundary condition implementations. Again, the parsing callbacks are triggered in the "register_commands" member function and again, there is little to do in initialization but store the subroutine name in the instance's control data. In the "workset" part of the code (or that part that corresponds to a workset algorithm), the existence of a subroutine association is checked. From there, the code can retrieve the subroutine pointer from the framework registry, check its signature, and call it. In the initial condition class, the call is made by overloading the set_nodal_variable member function. The subroutine pointer is passed in as a parameter and the appropriate nodal variable is set with the subroutine a single node at a time.

The signatures associated with each use in the FUEGO code are listed in Table 6.2-11.

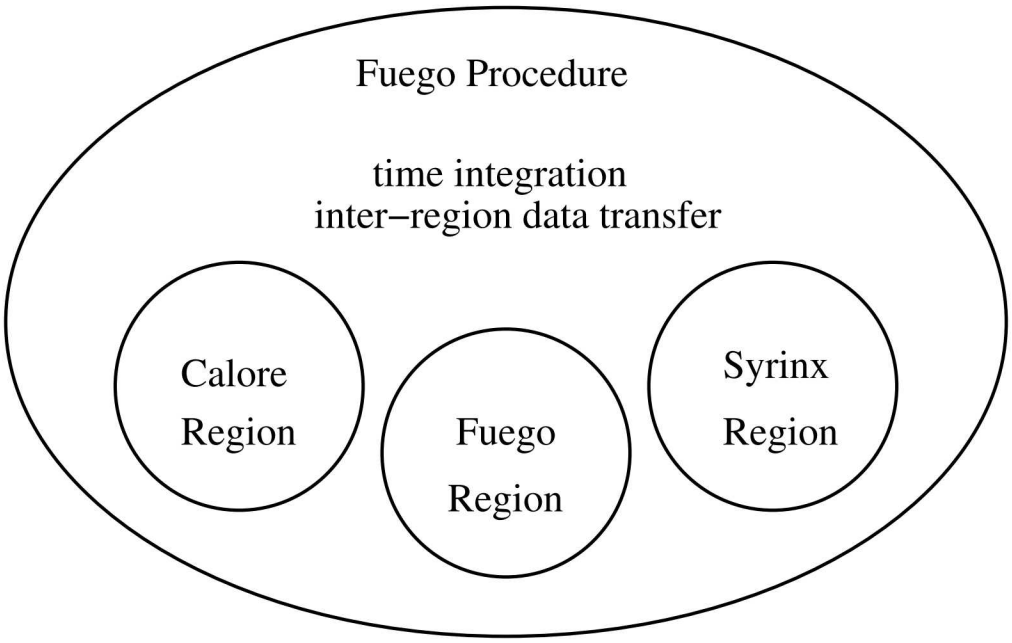


**Figure 6.2-1.. Fuego Procedure Class**

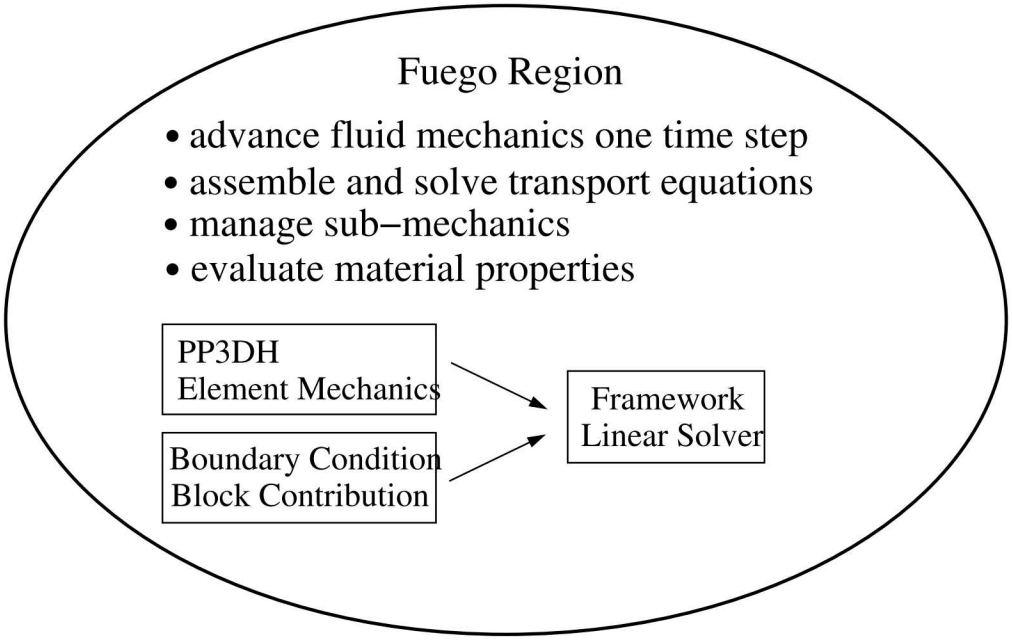


**Figure 6.2-2.. Fuego Region Class**

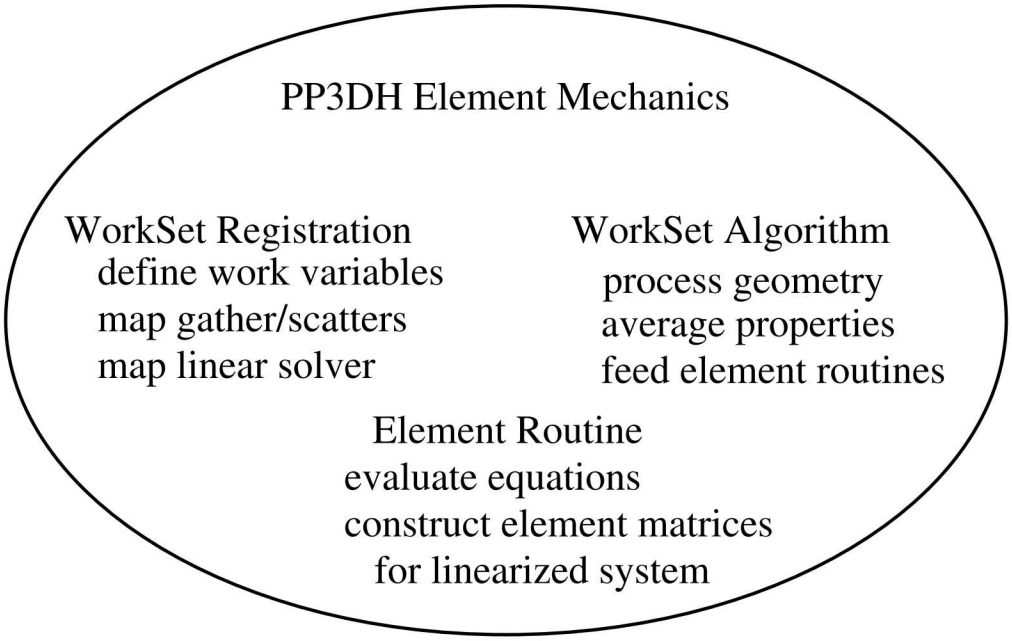


**Figure 6.2-3.. Fuego Element Mechanics Class**

**Table 6.2-1.. Fuego Frameworks Classes**

| | |
|---|---|
| Afgo_Procedure_Support.C | Time integration procedures and state data management. |
| Afgo_Region_Support.C | Nonlinear solution procedure within a time-step and linear solver interactions. |
| Afgo_Region_Parsing.C | Region-specific parsing routines. |
| Afgo_ConstuctElemMech.C | Maybe element mechanics support object to an element block. |
| Afgo_ElemMech_Support.C | Workset algorithms for the transport equations. |
| Afgo_ElemMech_Register.C | Workset registration and solver registration for for the transport equation workset algorithms. |
| Afgo_Dirichlet_*_Support.C | Data management for Dirichlet boundary conditions. |
| Afgo_Input_*BC_Support.C | Input parsing for flux boundary conditions. |
| Afgo_InflowBC_Support.C | Workset algorithms for the flux boundary conditions at an inflow. |
| Afgo_InflowBC_Register.C | Workset registration and solver registration for the flux boundary conditions at an inflow. |
| Afgo_OutflowBC_Support.C | Workset algorithms for the flux boundary conditions at an outflow. |
| Afgo_OutflowBC_Register.C | Workset registration and solver registration for the flux boundary conditions at an outflow. |
| Afgo_SymmetryBC_Support.C | Workset algorithms for the flux boundary conditions at a symmetry plane. |
| Afgo_SymmetryBC_Register.C | Workset registration and solver registration for the flux boundary conditions at a symmetry plane. |
| Afgo_WallBC_Support.C | Workset algorithms for the flux boundary conditions at a wall. |
| Afgo_WallBC_Register.C | Workset registration and solver registration for the flux boundary conditions at a wall. |
| Afgo_Material.C | Material property evaluation methods. |
| Afgo_ConstInitCond_Support.C | Methods to set the initial conditions. |

**Table 6.2-2.. Fuego Procedure Control Data**

| | |
|---|---|
| CODE_VERSION | “0.0.0", fuego version number must match code |
| RESTART_VERSION | “0.0.0", restart file created with a version number must match current |
| PROCEDURE_CONVERGENCE | “TRUE" \| “FALSE", have we converged within this time step |
| PROCEDURE_SUBITERATION | 0, number of subiterations taken within time step |
| MAX_PROCEDURE_SUBITERATION | 1, maximum subiterations within time step |
| DEBUG_LEVEL | 0 \| 1, enable debug messages |
| PMR_SKIP | 1, step interval for evaluating participating media radiation (PMR) |
| NUM_TIME_PERIODS | 0, the number of time step definition blocks |
| GLOBAL_TIMESTEP_COUNTER | 0, the total number of time steps taken |
| TIMEBLOCK_TIMESTEP_COUNTER | 0, the number of time steps taken within the time block |

**Table 6.2-3.. Fuego Region Control Data**

| | |
|---|---|
| DEBUG_LEVEL | 0 \| 1, provide debugging information |
| COORDINATE_SYSTEM | coordinate system |
| REGION_CONVERGENCE | 0 \| 1, has this nonlinear iteration sequence converged |
| REGION_CUTOFF | 0, if all equations meet their nonlinear residual tolerance on the first subiteration, then shut down the code |
| REGION_SUBITERATION | 0, number of nonlinear iterations over equation sets |
| MIN_REGION_SUBITERATIONS | 1, minimum number of nonlinear iterations. |
| MAX_REGION_SUBITERATIONS | 1, maximum number of nonlinear iterations. |
| CONT_NONLIN_TOLERANCE | 1.0e-8, tolerance on continuity equation nonlinear residual for stopping sub-iteration process |
| XMOM_NONLIN_TOLERANCE | 1.0e-8, tolerance on x-momentum equation nonlinear residual for stopping sub-iteration process |
| YMOM_NONLIN_TOLERANCE | 1.0e-8, tolerance on y-momentum equation nonlinear residual for stopping sub-iteration process |
| ZMOM_NONLIN_TOLERANCE | 1.0e-8, tolerance on z-momentum equation nonlinear residual for stopping sub-iteration process |
| TEMP_NONLIN_TOLERANCE | 1.0e-8, tolerance on temperature equation nonlinear residual for stopping sub-iteration process |
| ENTH_NONLIN_TOLERANCE | 1.0e-8, tolerance on enthalpy equation nonlinear residual for stopping sub-iteration process |
| SPEC_NONLIN_TOLERANCE | 1.0e-8, tolerance on species equation nonlinear residual for stopping sub-iteration process |
| TRBK_NONLIN_TOLERANCE | 1.0e-8, tolerance on turbulent kinetic energy equation nonlinear residual for stopping sub-iteration process |
| TRBE_NONLIN_TOLERANCE | 1.0e-8, tolerance on turbulence dissipation equation nonlinear residual for stopping sub-iteration process |
| CONT_URF | 1.0, under-relaxation factor |
| XMOM_URF | 1.0, under-relaxation factor |
| YMOM_URF | 1.0, under-relaxation factor |
| ZMOM_URF | 1.0, under-relaxation factor |
| TEMP_URF | 1.0, under-relaxation factor |
| ENTH_URF | 1.0, under-relaxation factor |
| SPEC_URF | 1.0, under-relaxation factor |
| TRBK_URF | 1.0, under-relaxation factor |
| TRBE_URF | 1.0, under-relaxation factor |
| TVISC_URF | 1.0, under-relaxation factor |
| L2_NORM_SCALING | 1.0, Scale L2 norm by number of nodes |
| TIME_STEP | 0.0, fixed time step, copied from the procedure |
| TIME_STEP_TYPE | 0, for h-adaptive scheme |
| WRITE_STATUS | flag to write status info |
| CURRENT_TIME | 0.0, the current time at the (N+1) time level |
| CURRENT_TIME | 0.0, the current time at the (N+1) time level |
| SUB_MECHANICS | PUVW \| PUVWT \| PUVWKE \| PUVWY, define math models (equation sets) to solve. |
| PERSISTENT_TEMPERATURE | define a temperature for code-coupling, only |

| | |
|---|---|
| STATE_EVALUATION_MODE | USE_REFERENCE_STATE \| USE_REFERENCE_MASS_FRACTIONS \| USE_REFERENCE_TEMPERATURE \| USE_ACTUAL_STATE, defines independent variables for material property evaluation within the element block |
| USE_REFERENCE_TEMPERATURES | "TRUE" \| "FALSE", use library reference value for property eval. |
| NUMBER_MATERIALS | 1, number of materials in the region |
| NUM_SV_PROPS | 0, number of properties to evaluate. |
| SV_PROP_NAMES[] | list of property names, match material library, string |
| SV_PROP_VARS[] | list of workset variable names for property variables, string |
| RESIDUAL_FILENAME | "", write the nonlinear residual history |
| BUOYANCY | "NONE" \| "BUOYANT" \| "DIFFERENTIAL" \| "BOUSSINESQ", activate buoyancy body force terms using one of the listed models |
| GRAVITY[] | gravity vector |
| BUOYANCY_REF_TEMPERATURE | 0.0, buoyancy reference temperature |
| BUOYANCY_REF_MASS_FRACTION_XXX | 0.0, buoyancy reference mass fraction of species XXX |
| BUOYANCY_REF_DENSITY | 0.0, buoyancy reference density |
| BUOYANCY_MASS_REF | 0 \| 1, use mass fractions |
| BUOYANCY_MOLE_REF | 0 \| 1, use mole fractions |
| NUMBER_OF_SPECIES | 0, number of species |
| MULTICOMPONENT | 0 \| 1, species transport equations are active |
| SOLVER_SPECIES_NUMBER | 0, the species equation are we currently solving |
| TURBULENCE_MODEL | "laminar" \| "k_e" \| "v2f" \| "kl", turbulence model definition. |
| NEED_YP | 0 \| 1, compute normal distance from wall |
| NEED_UTAU | 0 \| 1, compute friction velocity |
| BUOYANT_VORTICITY_GEN | 0 \| 1, add BVG model |
| ADD_MOLECULAR_VISC | 0 \| 1, add molecular viscosity to turbulence model diffusion |
| OMIT_WALL_TKE | 0 \| 1, wall bc treatment for turb ke |
| TURBULENCE_MODEL_CMU | 0.0, k-e model parameter from global constant library |
| TURBULENCE_MODEL_SIGMA_K | 0.0, k-e model parameter from global constant library |
| TURBULENCE_MODEL_SIGMA_E | 0.0, k-e model parameter from global constant library |
| TURBULENCE_MODEL_CEPS_1 | 0.0, k-e model parameter from global constant library |
| TURBULENCE_MODEL_CEPS_2 | 0.0, k-e model parameter from global constant library |
| TURBULENCE_MODEL_CEPS_3 | 0.0, buoyant vorticity generation constant from global constant library |
| TURBULENCE_MODEL_CBVG | 0.0, buoyant vorticity generation constant from global constant library |
| TURBULENCE_MODEL_CF_1 | 0.0, v2-f model parameter from global constant library |
| TURBULENCE_MODEL_CF_2 | 0.0, v2-f model parameter from global constant library |
| TURBULENCE_MODEL_ALPHA | 0.0, v2-f model parameter from global constant library |
| TURBULENCE_MODEL_NSEG | 0.0, v2-f model parameter from global constant library |
| TURBULENCE_MODEL_CL | 0.0, v2-f model parameter from global constant library |
| TURBULENCE_MODEL_CETA | 0.0, v2-f model parameter from global constant library |

| | |
|---|---|
| MINIMUM_TIME_STEP | 0.0, minimum allowable time step |
| MAXIMUM_TIME_STEP | 0.0, maximum allowable time step |
| TIME_STEP_CHANGE_FACTOR | 1.0, rate at which time step is allowed to change for step to step |
| CFL_LIMIT | 0.0, criterion for specifying time step |
| MAX_CFL | 0.0, the maximum CFL number over the mesh |
| MIN_CFL | 0.0, the minimum CFL number over the mesh |
| MAX_REY | 0.0, the maximum cell Reynolds number over the mesh |
| MIN_REY | 0.0, the minimum cell Reynolds number over the mesh |
| SIZE_SOLVER_LIST | 1<br>number of solvers (equation sets) |
| SOLVER_LIST[] | “solve_puvw"<br>evaluation ordering for equation sets |
| UPWIND_FACTOR | 0.05, blending factor for pure first-order upwind convection |
| UPWIND_METHOD | upwind convection method |
| UPWIND_LIMITER | slope limiter for MUSCL scheme |
| HYBRID_FACTOR | 1.0, multiplier for cell-Peclet number to control hybrid scheme blending |
| RHIE_CHOW_SCALING | 0 \| 1, activate the scaled Rhie/Chow scheme |
| PRESSURE_SMOOTHING | 0 \| 1, activate the fourth-order pressure smoothing |
| OMIT_DENSITY_DERIVATIVE | 0 \| 1, remove density time derivative from continuity |
| DENSITY_PREDICTOR | 0 \| 1, use a density predictor in time |
| THERMODYNAMIC_PRESSURE_IS_VARIABLE | 0 \| 1, all thermodynamic pressure to vary |
| SCALE_ENTHALPY | 0 \| 1, scale the enthalpy equation |
| ENTHALPY_FORM | 0 \| 1, use the enthalpy form of the energy equation |
| EDC_COMBUSTION | 0 \| 1, activate the combustion model |
| EDC_SOOT | 0 \| 1, activate the soot model |
| EDC_ABSORPTION | 0 \| 1, activate the radiation absorption model |
| EDC_REACTION_TIME_SCALE | 0.0, characteristic time scale for a chemical reaction |
| IGNITION_TIME | 0.0, time at which the flow is ignited |
| PRODUCT_MIN | 0.0, the minimum mass fraction of products required to ignite |
| FUEL_NAME | 0.0, the name of the fuel species |
| SOOT_TEMPERATURE_MIN | 0.0, the lower limit on temperature for producing soot |
| SINTEF_SOOT_MODEL | 0 \| 1, use the SINTEF soot model |
| INDEX_OXY | 0, the index number for oxygen in the species list |
| INDEX_FUEL | 0, the index number for fuel in the species list |
| INDEX_CO | 0, the index number for carbon monoxide in the species list |
| INDEX_$CO_2$ | 0, the index number for carbon dioxide in the species list |
| INDEX_$H_2$ | 0, the index number for hydrogen in the species list |
| INDEX_$H_2O$ | 0, the index number for water in the species list |
| STOICH_$O_2$_FUEL | 0, the stoichiometric ratio of oxygen to fuel |
| STOICH_$O_2$_CO | 0, the stoichiometric ratio of oxygen to carbon monoxide |
| STOICH_$O_2$_$H_2$ | 0, the stoichiometric ratio of oxygen to hydrogen |

**Table 6.2-4.. Fuego Element Mechanics Instance Control Data**

| | |
|---|---|
| formationTime | time required to form matrix, for load-balancing |
| loadMeasure | the number of elements processed, for load-balancing |
| MATERIAL_NAME | name from MATERIAL_LIBRARY entry, define material for this element block |

**Table 6.2-5.. Fuego BC Mechanics Instance Control Data**

| | |
|---|---|
| VARIABLE_TYPE[] | "CONSTANT", "NULL", variable is constant |
| VARIABLE_TYPE[] | "FUNCTION", "X"\|"Y"\|"Z", variable is a function of X\|Y\|Z |
| VARIABLE_TYPE[] | "SUBROUTINE", name, variable comes from a subroutine |
| pressure | 0.0 \| "function name" \|, constant value or function name for subroutine name |
| x-velocity | 0.0 \| "function name" \|, constant value or function name for subroutine name |
| y-velocity | 0.0 \| "function name" \|, constant value or function name for subroutine name |
| z-velocity | 0.0 \| "function name" \|, constant value or function name for subroutine name |
| temperature | 0.0 \| "function name" \|, constant value or function name for subroutine name |
| OMIT_DIFFUSION_TERMS | turn off the diffusion terms for an outflow boundary |
| FLOW_MUST_EXIT_DOMAIN | force flow to leave domain for an outflow boundary |

**Table 6.2-6.. Fuego Sub-Mechanics Definitions**

| | |
|---|---|
| T | heat conduction, 1 equation |
| PUVW | isothermal, uniform, laminar flow, 4 equations |
| PUVWT | thermal, temperature-form, uniform, laminar flow, 5 equations |
| PUVWH | thermal, enthalpy-form, uniform, laminar flow, 5 equations |
| PUVWY | isothermal, nonuniform, laminar flow, (3+NSPEC) equations |
| PUVWHY | thermal, enthalpy-form, nonuniform, laminar flow, (4+NSPEC) equations |
| PUVWKE | isothermal, uniform, turbulent flow, 6 equations |
| PUVWKEH | thermal, uniform, turbulent flow, 7 equations |
| PUVWKEY | isothermal, nonuniform, turbulent flow, (5+NSPEC) equations |
| PUVWKEHY | thermal, nonuniform, turbulent flow, (6+NSPEC) equations |
| PUVWKEHYSN | thermal, nonuniform, soot, turbulent flow, (8+NSPEC) equations |

**Table 6.2-7.. Fuego Element-Mechanics Workset Definitions**

| | |
|---|---|
| assemble_gradient | approximate the gradient of a scalar at nodes by integrating over control-volume faces, assembling the element-level contributions into the global nodal arrays. This routine is used for the MUSCL convection scheme. |
| assemble_pressure_gradient | approximate the gradient of the pressure at nodes by integrating over control-volume faces, assembling the element-level contributions into the global nodal arrays. This routine is used for a Rhie/Chow formulation for the continuity equation. |
| compute_cfl | compute the maximum and minimum cfl number over the elements |
| compute_ap | assemble the diagonal scaling term that is used in the Rhie/Chow interpolation for mass flow rate |
| laminar_p | assemble the continuity equation |
| laminar_p_update | reassemble the continuity equation, but only update the mass flow rate |
| laminar_u | assemble the laminar x-momentum equation |
| laminar_v | assemble the laminar y-momentum equation |
| laminar_w | assemble the laminar z-momentum equation |
| laminar_t | assemble the laminar temperature equation |
| laminar_h | assemble the laminar enthalpy equation |
| laminar_y | assemble a laminar species equation |
| turbulent_u | assemble the turbulent x-momentum equation |
| turbulent_v | assemble the turbulent y-momentum equation |
| turbulent_w | assemble the turbulent z-momentum equation |
| turbulent_h | assemble the turbulent enthalpy equation |
| turbulent_y | assemble a turbulent species equation |
| turbulent_s | assemble a turbulent soot equation |
| turbulent_n | assemble a turbulent soot transport equation |
| turbulent_k | assemble a turbulent kinetic energy transport equation, k-e model |
| turbulent_e | assemble a turbulence dissipation transport equation, k-e model |
| turbulent_k_v2f | assemble a turbulent kinetic energy transport equation, v2-f model |
| turbulent_e_v2f | assemble a turbulence dissipation transport equation, v2-f model |
| turbulent_v2 | assemble a turbulent v2 transport equation, v2-f model |
| turbulent_f | assemble a turbulent Helmholtz equation, v2-f model |

**Table 6.2-8.. Fuego Dirichlet Boundary Condition Definitions**

| | |
|---|---|
| Afgo_Dirichlet_U_Support | fixed nodal x-velocity component |
| Afgo_Dirichlet_V_Support | fixed nodal y-velocity component |
| Afgo_Dirichlet_W_Support | fixed nodal y-velocity component |
| Afgo_Dirichlet_P_Support | fixed nodal pressure |
| Afgo_Dirichlet_T_Support | fixed nodal temperature |
| Afgo_Dirichlet_H_Support | fixed nodal enthalpy |
| Afgo_Dirichlet_Y_Support | fixed nodal mass fraction |
| Afgo_Dirichlet_K_Support | fixed nodal turbulent kinetic energy |
| Afgo_Dirichlet_E_Support | fixed nodal turbulence dissipation |
| Afgo_Dirichlet_Wall_K_Support | fixed nodal turbulent kinetic energy, wall function implementation |
| Afgo_Dirichlet_Wall_E_Support | fixed nodal turbulence dissipation, wall function implementation |
| Afgo_Dirichlet_V2F_E_Support | fixed nodal turbulence dissipation, v2-f model implementation |

**Table 6.2-9.. Fuego Master Element Definitions**

| | |
|---|---|
| Ehex_H8_scs | eight-node hexahedral element for CVFEM, integration locations at sub-faces and element faces |
| Ehex_H8_scv | eight-node hexahedral element for CVFEM, integration locations at sub-volume |
| Etet_Te4_scs | four-node tetrahedron element for CVFEM, integration locations at sub-faces and element faces |
| Etet_Te4_scv | four-node tetrahedron element for CVFEM, integration locations at sub-volume |
| Ewed_W6_scs | six-node wedge element for CVFEM, integration locations at sub-faces and element faces |
| Ewed_W6_scv | six-node wedge element for CVFEM, integration locations at sub-volume |
| Ehex_3DTr3_scs | three-node triangular element for CVFEM, integration locations at sub-volumes |
| Ehex_3DQ4_scs | four-node quadrilateral element for CVFEM, integration locations at sub-volumes |

**Table 6.2-10.. User Subroutine Argument Lists**

| | |
|---|---|
| Apub_ftx3_sub | int*, real*, real*, real*, int*, int*, real*, int* |
| Apub_ftx3spec_sub | int*, real*, real*, int*, real*, int*, int*, real*, int* |
| Afgo_fmmsfgo_sub | int*, int*, int*, real*, real*, real*, real*, real*, real*, real*, real*, real*, real* |

**Table 6.2-11.. User Subroutine Signature Type**

| | |
|---|---|
| Heat BC | Apub_ftx3_sub |
| Convection BC | Apub_ftx3_sub |
| Radiation BC | Apub_ftx3_sub |
| Fixed BC | Apub_ftx3_sub, Apub_ftx3spec_sub |
| Inflow BC | Apub_ftx3_sub, Apub_ftx3spec_sub |
| Outflow BC | Apub_ftx3_sub |
| Wall BC | Apub_ftx3_sub, Apub_ftx3spec_sub |
| Initial Conditions | Apub_ftx3spec_sub |
| Source Terms | Afgo_fmmsfgo_sub |

# BIBLIOGRAPHY


[1] Tieszen, S. R., A. R. Lopez, C. D. Moen, T. Y. Chu, V. F. Nicolette, W. Gill, S. P. Burns, and W. C. Moffatt. "SIERRA/Fuego and SIERRA/Syrinx Verification and Validation Plan, Version 2.0". internal report, Sandia National Laboratories, 2001.

[2] Tieszen, S. R., T. Y. Chu, V. F. Nicolette, K. J. Dowding, B. F. Blackwell, A. R. Lopez, W. Gill, and S. P. Burns. "Fire Environment Simulation: Fuego/Syrinx Strategic Development Plan". unpublished report, Sandia National Laboratories, May 1999.

[3] Rehm, R. G. and H. R. Baum. "The Equations of Motion for Thermally Driven Buoyant Flows". Journal of Research of the National Bureau of Standards, 83:279, 1978.

[4] Paolucci, S. "On the Filtering of Sound Waves from the Navier-Stokes Equations". Technical Report SAND Report 82–8257, Sandia National Laboratories, Livermore, CA, December 1982.

[5] Majda, A. and J. Sethian. "The Derivation and Numerical Solution of the Equations for Zero Mach Number Combustion". Combustion Science and Technology, 42:185–205, 1985.

[6] Merkle, C. L. and Y. H. Choi. "Computation of Compressible Flows at Very Low Mach Numbers". AIAA Paper 86–0351, AIAA 24th Aerospace Sciences Meeting, Reno, NV, January 1986.

[7] Bird, R. B., W. E. Stewart and E. N. Lightfoot. Transport Phenomena. John Wiley and Sons, 1960.

[8] Burns, S. P. "Turbulence Radiation Interaction Modeling in Combustion Simulations". Technical report, Sandia National Laboratories, Albuquerque, NM, 1999.

[9] Modest, M. F. Radiation Heat Transfer. McGraw Hill Book Company, New York, 1993.

[10] Siegel R. and J. R. Howell. Thermal Radiation Heat Transfer, 3rd ed. Hemisphere Publishing, Washington, D.C., 1992.

[11] Tennekes, H. and J. L. Lumley. A First Course in Turbulence. MIT Press, Cambridge, 1972.

[12] Libby, P. A. and F. A. Williams.
Turbulent Reacting Flows, Fundamental Aspects, Topics in Applied Physics, V. 44.
Springer-Verlag., 1980.

[13] Kuo, K. K. Principles of Combustion. John Wiley and Sons, 1986.

[14] Wilcox, D. C. Turbulence Modeling for CFD. DCW Industries, 2nd edition, 1998.

[15] Moin, P., K. Squires, W. Cabot, and S. Lee. A dynamic subgrid-scale model for compressible turbulence and scalar transport. Phys. Fluids A, 3(11):2746–2757, 1991.

[16] Erlebacher, G., M. Y. Hussaini, C. G. Speziale, and T. A. Zang. Toward the large-eddy simulation of compressible turbulent flows. J. Fluid Mech., 238:155–185, 1992.

[17] Gran, I. M. C. Melaaen, and B. F. Magnussen. "Numerical Simulation of Local Extinction Effects in Turbulent Combustor Flows of Methane and Air". In 25th Symposium on Combustion, pages 1283–1291. The Combustion Institute, 1994.

[18] Tieszen, S. R., S. P. Domino, A. R. Black. Validation of a simple turbulence model suitable for closure of temporally-filtered Navier-Stokes equations using a helium plume. Technical Report SAND Report 2005–3210, Sandia National Laboratories, Albuquerque, NM, June 2005.

[19] Launder, G. E. and B. I. Sharma. Application of the energy dissipation model of turbulence to the calculation of flow near a spining disc. Letters in Heat and Mass Transfer, 1(2):131–138, 1974.

[20] Papageorgakis, G. C. and D. N. Assanis. Comparison of linear and nonlinear RNG-based $k$-$\epsilon$ models for incompressible turbulent flows. Numerical Heat Transfer, Part B, 35(1):1–22, 1999.

[21] Durbin, P. A. "Near-Wall Turbulence Closure Modeling Without Damping Functions". Theoretical and Computational Fluid Dynamics, 3:1–13, 1991.

[22] Wilcox, D. C. "Formulation of the k-$\omega$ turbulence model revisisted". 45th AIAA Aerospace Sciences Meeting and Exhibit, 2007.

[23] Menter, F. R., Kuntz, M. and R. Langtry. "Ten years of industrial experience with the SST turbulence model". Turb, Heat and Mass Trans, 2003.

[24] Smagorinsky, J. General circulation experiments with the primitive equations. i. the basic experiment. Monthly Weather Review, 91:99–164, 1963.

[25] Rogallo, R. S. and P. Moin. Numerical simulation of turbulent flows. Annual Review of Fluid Mechanics, 16:99–137, 1984.

[26] Germano, M., U. Piomelli, P. Moin, and W. H. Cabot. A dynamic subgrid-scale eddy viscosity model. Physics of Fluids A, 3(7):1760–1765, July 1991.

[27] Germano, M. Turbulence: The filtering approach. Journal of Fluid Mechanics, 238:325–336, 1992.

[28] Ghosal, S., T. S. Lund, P. Moin, and K. Akselvoll. A dynamic localization model for large-eddy simulation of turbulent flow. Journal of Fluid Mechanics, 286:229–255, 1995.

[29] Lilly, D. K. A proposed modification of the germano subgrid-scale closure model. Physics of Fluids A, 4(3):633–635, March 1992.

[30] Kim, W. W. and S. Menon. "Application of the localized dynamic subgrid-scale model to turbulent wall-bounded flows". 35th AIAA Aerospace Sciences Meeting and Exhibit, 1997.

[31] Nicolette, V. F. and S. R. Tieszen. " Effect of Turbulent Kinetic Energy Source Terms on Pool Fire Simulations with the $k - \epsilon$ Model". Technical report, Internal Memorandum to Distribution, Official Use Only, Sandia National Laboratories, 2000.

[32] Rodi W. Turbulence Models and their Applications in Hydrolics - A State of the Art Review. Publication of teh International Association for Hydrolic Research, Delf, Netherlands, 1984.

[33] de Ris, J. L. Mechanism of buoyant turbulent diffusion flames. Procedia Engineering, 62:13–27, 2013.

[34] Jones, W. P. and B. E. Launder. "The Prediction of Laminarization with a Two-Equation Model of Turbulence". International Journal of Heat and Mass Transfer, 15:301–314, 1972.

[35] White, F. M. Viscous Fluid Flow. McGraw-Hill, Inc., 2nd ed., 1991.

[36] Sondak, D. L. and R. H. Pletcher. " Application of wall functions to generalized nonorthogonal curvilinear coordinate systems ". AIAA Journal, 33(1):33–41, January 1995.

[37] Elkaim, D., M. Reggio, and R. Camarero. "Control Volume Finite-Element Solution of a Confined Turbulent Diffusion Flame". Numerical Heat Transfer, Part A, 23(3):259–279, 1993.

[38] Launder, B. E. and D. B. Spalding. "The Numerical Computation of Turbulent Flows". Computer Methods in Applied Mechanics and Engineering, 3:269–289, 1974.

[39] Versteeg, H. K. and W. Malalasekera. An Introduction to Computational Fluid Dynamincs. Longman Group LTD, 1995.

[40] Jayatilleke, C. L. V. "The Influence of Prandtl Number and Surface Roughness on the Resistance of Laminar Sub-Layer to Momentum and Heat Transfer". Progress in Heat and Mass Transfer, 1, 1969.

[41] Magnussen, B. F., G. H. Hjertager, J. G. Olsen, and D. Bhaduri. "Effect of Turbulent Structure and Local Concentrations on Soot Formation and Combustion in C2H2 Diffusion Flames". In Seventeenth Symposium (International) on Combustion, pages 1383–1393. The Combustion Institute, Pittsburgh, 1979.

[42] Magnussen, B. F. "On the Structure of Turbulence and a Generalised Eddy Dissipation Concept for Chemical Reactions in Turbulent Flow". 9th AIAA Sc. Meeting, St. Louis, 1981.

[43] Byggstyøl, S. and B. F. Magnussen. "A Model for Extinction in Turbulent Flows". In et al. Bradbury, editor, $4^{th}$ Symposium on Turbulent Shear Flow, pages 381–395. Springer-Verlag, Berlin, 1983.

[44] Magnussen, B. F. "Heat Transfer in Gas Turbine Combustors – A Discussion of Combustion, Heat and Mass Transfer in Gas Turbine Combustors". In Conference Proceedings no. 390, Advisory Group for Aerospace Research and Development (AGARD), 1985.

[45] Lilleheie, N. I., I. Ertesvåg, T. Bjorge, S. Byggstyøl, and B. F. Magnussen. "Modeling and Chemical Reactions, Review of Turbulence and Combustion Models". Technical report, The Foundation for Scientific and Industrial Research, Norwegian Institute of Technology, SINTEF Report STF15 A89024, July 1989.

[46] Gran, I. and B. F. Magnussen. "A Numerical Study of a Bluff-body Stabilized Diffusion Flame. Part 2: Influence of Combustion Modeling and Finite Rate Chemistry". Combustion Science and Technology, 119:191–217, 1996.

[47] Tieszen, S. R., V. F. Nicolette, L. A. Gritzo, J. K. Holen, D. Murray, and J. L. Moya. "Vortical Structures In Pool Fires: Observation, Speculation, and Simulation". Technical report, SAND96-2607, Sandia National Laboratories, Albuquerque, NM, November 1996.

[48] Strehlow, R. A. Combustion Fundamentals. McGraw-Hill, New York, 1984.

[49] Ertesvåg, I. S. and B. F. Magnussen. "The Eddy-dissipation turbulence energy cascade model". Technical report, Department of Applied Mechanics, Thermodynamics and Fluid Dynamics, The Norwegian University of Science and Technology, Trondheim, Norway (In Preparation), 1997.

[50] Holen, J., B. Lakså, B. F. Magnussen, and B. E. Vembe. "KAMELEON-II-Fire Theory Manual, A Description of the Mathematical Models, Numerical Methods, and Solution Procedures". Technical report, The Foundation for Scientific and Industrial Research, Norwegian Institute of Technology, Trondheim, Norway, SINTEF Report STF15 F94070, November 1994.

[51] N. Peters. Laminar diffusion flamelet models in non-premixed turbulent combustion. Progress in Energy and Combustion Science, 10:319–339, 1984.

[52] Danny Messig, Franziska Hunger, Jens Keller, and Christian Hasse. Evaluation of radiation modeling approaches for non-premixed flamelets considering a laminar methane air flame. Combustion and Flame, 160(2):251–264, 2013.

[53] T. Poinsot and D. Veynante. Theoretical and Numerical Combustion. R. T. Edwards, Inc., Philadelphia, PA, 2005.

[54] C. D. Pierce and P. Moin. A dynamic model for subgrid-scale variance and dissipation rate of a conserved scalar. Physics of Fluids, 10(12), 3041–3044.

[55] R. W. Bilger. The structure of turbulent nonpremixed flames. In Proceedings of the 22nd Symposium (International) on Combustion, pages 475–488. The Combustion Institute, Pittsburg, PA, 1988.

[56] G. Erlebacher, M. Y. Hussaini, C. G. Speziale, and T. A. Zang. Toward the large eddy simulation of compressible turbulent flows. ICASE Report 87-20, NASA Langley Research Center, Hampton, VA, 1987. Also available as NASA CR 178273.

[57] S. Ghosal, T.S. Lund, P. Moin, and K. Akselvoll. A dynamic localization model for large-eddy simulation of turbulent flow. Journal of Fluid Mechanics, 286:229–255, 1995.

[58] Parente, A. and Malik, M. R. and Contino, F. and Cuoci,A. and Dally, B. B. "Extension of the Eddy Dissipation Concept for turbulence/chemistry interactions to MILD combustion". Fuel, 163:98–111, 2016.

[59] Zukoski, E. E. "Properties of Fire Plumes". In G. Cox, editor, Combustion Fundamentals of Fire. Academic Press, New York, 1995.

[60] Magnussen, B. F. and G. H. Hjertager. "On Mathematical Modeling of Turbulent Combustion with Special Emphasis on Soot Formation and Combustion". In Sixteenth Symposium (International) on Combustion, pages 719–729. The Combustion Institute, Pittsburgh, 1977.

[61] Tesner, P. A., T. D. Snegiriova, and V. G. Knorre. "Kinetics of dispersed carbon formation". Combustion and Flame, 17:253–260, 1971.

[62] Haynes, B. S. and H. G. Wagner. "Soot Formation". Progress in Energy and Combustion Science, 7:229–273, 1981.

[63] Tesner, P. A., E. I. Tsygankova, L. P. Guilazetdinov, V. P. Zuyev, and G. V. Loshakova. "The formation of soot from aromatic hydrocarbons in diffusion flames of hydrocarbon-hydrogen mixtures". Combustion and Flame, 17:279–285, 1971.

[64] Tezduyar, T. E. "Stabilized Finite Element Formulations for Incompressible Flow Computations". In , editor, Advances in Applied Mechanics, volume 28, pages 1–44. Academic Press, Inc., 1992.

[65] Felske, J. D. and C. L. Tien. "Calculation of the Emissivity of Luminous Flames". Combustion Science and Technology, 7:23–31, 1973.

[66] Felske, J. D. and T. T. Charalampopoulos. "Gray Gas Weighting Coefficients for Arbitrary Gas-Soot Mixtures". International Journal of Heat and Mass Transfer, 25(12):1849–1855, 1982.

[67] Abramowitz, M. and I. A. Stegun. Handbook of Mathematical Functions. National Bureau of Standards, 1964.

[68] Leckner, B. "Spectral and Total Emissivity of Water Vapor and Carbon Dioxide". Combustion and Flame, 19:33–48, 1972.

[69] Martinez, M. J. and P. I. Hopkins. "Modeling Subsurface Multiphase Transport of JP8 During a Fuel Spill Fire". Technical report, SAND2000-2464, Sandia National Laboratories, Albuquerque, NM, October 2000.

[70] Saito, K. G. Tashtoush, C. Cremers, and L. A. Gritzo. "Flame Spread over JP8 Aircraft Fuel". to appear in Combustion Science and Technology, 1997.

[71] Gritzo, L. A., E. A. Boucheron, and D. Murray. "Fuel Temperature Distribution and Burning Rate in Large Pool Fires". NIST Annual Conference on Fire Research, Gaithersburg, MD, October 1996.

[72] Gritzo, L. A., V. F. Nicolette, S. R. Tieszen, and J. L. Moya. "Heat Transfer to the Fuel Surface in Large Pool Fires". In S. H. Chan, editor, Transport Phenomenon in Combustion, pages 701–712. Taylor and Francis, 1996.

[73] Blinov, V. I. and G. N. Khudiakov. "Diffusion Burning of Liquids". Technical report, English Translation: U.S. Army Engineering Research and Development Labs, Fort Belvoir, VA, Report AERDL-T-1490-A, 1961.

[74] Mansfield, J. M. and L. J. Linley. "Measurement and Statistical Analysis of Flame Temperatures from Large Fuel Spill Fires". Technical report, NWC TP 7061, Naval Air Warfare Center, China Lake, CA, 93555, 1991.

[75] Cline, D. D. and L. N. Koenig. "The Transient Growth of an Unconfined Pool Fire". Fire Technology, 19(3):149–162, 1983.

[76] Magnoli, D. E. "A Model for Fuel Fire Duration and Application to the B-1B Bomber". Technical report, Lawrence Livermore National Laboratory, Livermore, CA, UCRL-ID-112576, 1992.

[77] Mencinger, J. and Zun, I. Journal of Computational Physics, 230:644–663, 2011.

[78] Martinez, J. and Chesneau, X. and Zeghmati, B. Computational Mechanics, 37:182–193, 2006.

[79] Sun, M. Advances in Pure Mathematics, 3:219–225, 2013.

[80] Huang, M. and Wu, L. and Chen, B. Numerical Heat Transfer, Part B: Fundamentals, 61:412–437, 2012.

[81] Lee, H. and Rhee, S. Journal of Mechanical Science and Technology, 29:1849–1857, 2015.

[82] Raeini, A. and Blunt, M. and Bijeljic, B. Journal of Computational Physics, 231:5653–5668, 2012.

[83] Jofre, L. and Lehmkuhl, O. and Castro, J. and Olivia, A. In Lisbon Portugal, editor, European Conference on Computational Fluid Dynamics, 2010.

[84] Hardt, S. and Wondra, F. Journal of Computational Physics, 227:5781–5895, 2008.

[85] Francois, M. M. and Cummins, S. J. and Dendy, E. D. and Kothe, D. B. and Sicilian, J. M. and Williams, M. W. Journal of Computational Physics, 213:141–173, 2006.

[86] Lin, S. and Yan, J. and Kats, D. and Wagner, G. Journal of Computational Physics, 380:119–142, 2019.

[87] S. P. Domino. Towards verification of sliding mesh algorithms for complex applications using mms. Proceedings of the 2010 Summer Program, Center for Turbulence Research, 2010.

[88] M. Discacciati, A. Quarteroni, and A. Valli. Robin-Robin domain decomposition methods for the Stokes-Darcy coupling. SIAM Journal of Numerical Analysis, 45(3):1246–1268, 2007.

[89] G. S. Beavers and D. D. Joseph. Boundary conditions at a naturally permeable wall. J. Fluid Mech., 30:197–207, 1967.

[90] P. G. Saffman. On the boundary condition at the surface of a porous medium. Studies in Applied Mathematics, 50(2):93–101, 1971.

[91] R. H. Davis and H. A. Stone. Flow through beds of porous particles. Chemical Engineering Science, 48(23):3993–4005, 1993.

[92] Law, C. K. "Recent Advances in Droplet Vaporization and Combustion". Prog. Energy Combust Sci., 8(3):171–201, 1982.

[93] Sirignano, W. A. "Fuel Droplet Vaporization and Spray Combustion Theory". Prog. Energy Combust Sci., 9(4):291–322, 1983.

[94] Faeth, G. M. "Evaporation and combustion of sprays". Prog. Energy Combust Sci., 9:1–76, 1983.

[95] Amsden, A. A., et al. "KIVA: A Computer Program for Two-and Three-Dimensional Fluid Flows with Chemical Reactions and Fuel Sprays", 1985.

[96] Faeth, G.M. "Mixing, Transport and Combustion in Sprays". Prog. Energy Combust Sci., 13:293–345, 1987.

[97] Amsden, A.A., P.O'Rourke, and T.D. Butler. "KIVA-II: A Computer Program for Chemically Reactive Flows with Sprays", 1989.

[98] Crowe, C., M. Sommerfeld, and Y. Tsuji. "Multiphase flows with droplets and particles". CRC Press, New York, NY, 1998.

[99] Sommerfeld, M. and H.H. Qiu. "Experimental studies of spray evaporation in turbulent flow". International Journal of Heat and Fluid Flow, 19(1):10–22, 1998.

[100] Shaw, R.A. " Particle-turbulence interactions in atmospheric clouds". Annual Review of Fluid Mechanics, 35:183–227, 2003.

[101] Holen, J., M. Brostrom, and B.F. Magnussen. "Finite Difference Calculation of Pool Fire". Proc. Combust. Instit., 23:1677–1683, 1990.

[102] Yoon, S.S., et al. "Numerical modeling and experimental measurements of a high speed solid-cone water spray for use in fire suppression applications". International Journal of Multiphase Flow, 30(11):1369–88, 2004.

[103] DesJardin, P.E. and L.A. Gritzo. "A Dilute Spray Model for Fire Simulations: Formulation, Usage and Benchmark Problems". "Sandia Technical Report", 2002.

[104] Maxey, M.R. and J.J. Riley. "Equation of motion for a small rigid sphere in a nonuniform flow". Phys. Fluids, 26(4):883–889, 1983.

[105] Williams, F.A. "Spray Combustion and Atomization". Physics of Fluids, 1(6):541–545, 1958.

[106] O'Rourke, P.J. and A.A. Amsden. "The TAB Method for Numerical Calculation of Spray Droplet Breakup". SAE Technical Paper 872089, 1987.

[107] Patankar, S.V. "Numerical Heat Transfer and Fluid Flow". Taylor and Francis, 1980.

[108] Gosman, A.D. and E. Ioannides. "Aspects of computer simulation of liquid-fueled combustors". AIAA Paper 81-0323, 1981.

[109] Shuen, J.S., L.D. Chen, and G.M. Faeth. "Evaluation of a stochastic model of particle dispersion in a turbulent round jet". AIChE Journal, 29(1):167–70, 1983.

[110] Zhou, Q. and S.C. Yao. "Group modeling of impacting spray dynamics". International Journal of Heat and Mass Transfer, 35(1):121–9, 1992.

[111] Dukowicz, J.K. "A Particle-Fluid Numerical Model for Liquid Sprays". Journal of Computational Physics, 35(2):229–253, 1980.

[112] Jones, W.P. and B.E. Launder. "Prediction of Laminarization with a 2-Equation Model of Turbulence". Int. J. Heat Mass Transfer, 15:301, 1972.

[113] Yuen, M.C. and L.W. Chen. "On Drag of Evaporating Liquid Droplets". Combust. Sci. Technol., 14(4-5-6):147–154, 1976.

[114] Taylor, G.I. "Diffusion by continuous movement". Proc. London Math. Soc., 20:196–211, 1921.

[115] Snyder, W.H. and J.L. Lumley. "Some measurements of particle velocity autocorrelation functions in a turbulent flow". Journal of Fluid Mechanics, 48:41, 1971.

[116] Abramzon, B. and W.A. Sirignano. "Droplet vaporization model for spray combustion calculations". International Journal of Heat and Mass Transfer, 32(9):1605–18, 1989.

[117] Modest, M.F. "Radiative Heat Transfer". Academic Press, second ed. edition, 2003.

[118] Lefebvre, A.H. "Atomization And Sprays". Hemisphere, 1988.

[119] Johns, L.E. and R.B. Beckmann. "Mechanism of dispersed-phase mass transfer in viscous single-drop extraction systems". Amer. Instit. Chem. Eng. J., 12(1):10–16, 1966.

[120] Sirignano, W.A. "Fluid Dynamics and Transport of Droplets and Sprays". Cambridge University Press, New York, NY, 1999.

[121] van de Hulst, H.C. "Light Scattering by Small Particles". Dover, New York, 1981.

[122] Yoon, S.S., et al. "Numerical Modeling and Experimental Measurements of Water Spray Impact and Transport over a Cylinder". Int. J. Multiphase Flow, 2005.

[123] Erikson, W.W. and W. Gill. "Analytic Model for Propellant Fire Heat Transfer with Deposition". JANNAF 40th CS, 28th APS, 22nd PSHS and 4th MSS Joint Meeting, 2005.

[124] O'Rourke, P.J. "Statistical properties and numerical implementation of a model for droplet dispersion in a turbulent gas". Journal of Computational Physics, 83(2):345–60, 1989.

[125] Zienkiewicz, O. C. and R. L. Taylor. The Finite Element Method, 4th ed., Vol. 1. McGraw-Hill, 1989.

[126] Zienkiewicz, O. C. and R. L. Taylor. The Finite Element Method, 4th ed., Vol. 2. McGraw-Hill, 1991.

[127] Gresho, P. M, S. T. Chan, R. L. Lee, and C. D. Upson. "A Modified Finite Element Method for Solving the Time-Dependent, Incompressible Navier-Stokes Equations. Part I: Theory". International Journal for Numerical Methods in Fluids, 4:557–598, 1984.

[128] Patankar, S. V. "Recent Developments in Computational Heat Transfer". Journal of Heat Transfer, 110(4B):1037–1045, 1988.

[129] Shyy, W. "Elements of Pressure-Based Computational Algorithms for Comples Fluid Flow and Heat Transfer". In Hartnett, J. P. and T. F. Irvine, editor, Advances in Heat Transfer, volume 24, pages 191–275. Academic Press, Inc., 1994.

[130] Ferziger, J. H. and M. Perić. Computational Methods for Fluid Dynamics. Springer-Verlag, 1996.

[131] Rhie, C. M. and W. L. Chow. "Numerical Study of the Turbulent Flow Past an Airfoil with Trailing Edge Separation". AIAA Journal, 21(11):1525–1532, November 1983.

[132] Schneider, G. E. and M. J. Raw. "Control Volume Finite-Element Method for Heat Transfer and Fluid Flow Using Colocated Variables–1. Computational Procedure". Numerical Heat Transfer, 11(4):363–390, 1987.

[133] Schneider, G. E. and M. J. Raw. "Control Volume Finite-Element Method for Heat Transfer and Fluid Flow Using Colocated Variables–2. Application and Validation". Numerical Heat Transfer, 11(4):391–400, 1987.

[134] Majumdar, S. "Role of Under-relaxation in Momentum Interpolation for Calculation of Flow with Non-staggered Grids". Numerical Heat Transfer, 13:125–132, 1988.

[135] Perić, M., R. Kessler, and G. Scheuerer. "Comparison of Finite-Volume Numerical Methods with Staggered and Colocated Grids". Computers and Fluids, 16(4):389–403, 1988.

[136] Papageorgakopoulos, J, G. Arampatzis, D. Assimacopoulos, and N. C. Markatos. "Enhancement of the momentum interpolation method on non-structured grids". Int. J. Numer. Meth. Fluids, 33:1–22, 2000.

[137] Codina, R. "Pressure stability in fractional step finite element methods for incompressible flows". J. Comp. Phys., 170:112–140, 2001.

[138] Soto, O. R., F. Löhner, and J. Cebral. "An implicit monolithic time accurate finite element scheme for incompressible flow problems". AIAA-2001-2616, 2001.

[139] Almgren, A. S., J. B. Bell, and W. Y. Crutchfield. "Approximate projection methods: part I. inviscid analysis". SIAM J. Sci. Comp., 22:1139–1159, 2000.

[140] Codina, R. and S. Badia. "On some pressure segregation methods of fractional-step type for the finite element approximation of incompressible flow problems". Comp. Methods. Appl. Mech. Engr., 170:112–140, 2005.

[141] Dukowicz, J. K. and A. S. Dvinsky. "Approximate Factorization as a High Order Splitting for the Implicit Incompressible Flow Equations". Journal of Computational Physics, 102:336–347, 1992.

[142] Strikwerda, J. C. and Y. S. Lee. "The accuracy of the fractional step method". SIAM J. Numer. Anal., 37:37–48, 1999.

[143] Schneider, G. E. Elliptic Systems: Finite-Element Method I. In Minkowycz, W. J., E. M. Sparrow, G. E. Schneider and R. H. Pletcher, editor, Handbook of Numerical Heat Transfer, chapter 10, pages 379–420. John Wiley & Sons, Inc., 1988.

[144] Perot, J. B. "An Analysis of the Fractional Step Method". Journal of Computational Physics, 108:51–58, 1993.

[145] Chorin, A. J. "Numerical Solution of the Navier-Stokes Equations". Mathematics of Computation, 22(104):745–762, 1968.

[146] Kim, D. and H. Choi. "A second-order time-accurate finite volume method for unsteady incompressible flow on hybrid unstructured grids". J. Comp. Phys., 162:411–428, 2000.

[147] Kim, J. and P. Moin. "Application of a Fractional Step Method to Incompressible Navier-Stokes Equations". Journal of Computational Physics, 59(2):308–323, 1985.

[148] Brown, D. L., R. Cortez, and M. Minion. "Accurate projection method for the incompressible Navier-Stokes equations". J. Comp. Phys., 168:464–499, 2001.

[149] Schneider, G. E. "Preliminary Results of a Novel Fluid Flow Prediction Procedure Applied to Axi-Symmetric Problems". AIAA Paper 87-1639, 22nd Thermophysics Conference, Honolulu, HA, June 1987.

[150] Domino et. al. "Fuego Verification Manual". Sandia National Laboratories, http://scico.sandia.gov/fuego, 2007.

[151] Ober, C. C. and J. N. Shadid. "Studies on the accuracy of time-integration methods for the radiation-diffusion equations". J. Comp. Phys., 195:743–772, 2004.

[152] Hirsch, C. Numerical Computation of Internal and External Flows, Volume 2. John Wiley & Sons, 1990.

[153] F. Shakib, T. J. R. Hughes, and J. Zdenek. A new finite element formulation for computational fluids dynamics: The compressible euler and navier stokes equations. Comp. Meth. in App. Mech and Engr., 89:141–219, 1991.

[154] "TASCflow Theory Documentation". Technical report, Advanced Scientific Computing, Ltd., Waterloo, Ontario, 1995.

[155] Jameson, A. "Artificial Diffusion, Upwind Biasing, Limiters and Their Effect on Accuracy and Multigrid Convergence in Transonic and Hypersonic Flows". AIAA Paper 93-3359, 11th AIAA Computational Fluid Dynamics Conference, Orlando, FL, July 1993.

[156] Christie, I. and C. Hall. "The Maximum Principle for Bilinear Elements". International Journal for Numerical Methods in Engineering, 20(3):549–553, 1984.

[157] Yeap, C. F. and J. A. Pearce. "A Unified Subroutine for the Solution of 2D and 3D Axisymmetric Diffusion Equation". Advances in Engineering Software, 11(3):118–127, 1989.

[158] Blackwell, B. F., R. J. Cochran, and R. E. Hogan. "A Formal Method for Computing Thermal Conductors for Arbitrary, Complex Geometries". ASME-HTD Vol. 311, pp. 31–42, 30th National Heat Transfer Conference, Portland, OR, August 1995.

[159] Gresho, P. M, and R. L. Lee. "Don't Suppress the Wiggles—They're Telling You Something". Computers and Fluids, 9(2):223–253, 1981.

[160] Flanagan, D. P. and T. Belytschko. "A Uniform Strain Hexahedron and Quadrilateral with Orthogonal Hourglass Control". International Journal for Numerical Methods in Engineering, 17:679–706, 1981.

[161] Baliga, B. R. and S. V. Patankar. Elliptic Systems: Finite-Element Method II. In Minkowycz, W. J., E. M. Sparrow, G. E. Schneider and R. H. Pletcher, editor, Handbook of Numerical Heat Transfer, chapter 11, pages 421–462. John Wiley & Sons, Inc., 1988.

[162] Forsyth, P. A. "Control Volume Finite Element Approach to NAPL Groundwater Contamination". SIAM Journal on Scientific and Statistical Computing, 12(5):1029–1057, 1991.

[163] Aftosmis, M. "Upwind Method for Simulation of Viscous Flow on Adaptively Refined Meshes". AIAA Journal, 32(2):268–277, 1994.

[164] Kallinderis, Y. G. and J. R. Baron. "Adaptive Methods for a New Navier-Stokes Algorithm". AIAA Journal, 27(1):37–43, 1989.

[165] Kallinderis, Y. and P. Vijayan. "Adaptive Refinement-Coarsening Scheme for Three-Dimensional Unstructured Meshes". AIAA Journal, 31(8):1440–1447, 1993.

[166] Mavriplis, D. J. "Adaptive Meshing Techniques for Viscous Flow Calculations on Mixed Element Unstructured Meshes". International Journal for Numerical Methods in Fluids, 34(2):93–111, 2000.

[167] Edwards, H. C. "SIERRA Framework Core Services: Theory and Design". version 0.04, unpublished, 2001.

[168] Taylor, L. M., H. C. Edwards, and J. R. Stewart. "Functional Requirements for SIERRA, Version 1.0 beta". internal report, Sandia National Laboratories, 1999.

[169] Kee, R. J., F. M. Rupley, and J. A. Miller. "CHEMKIN-II: A Fortran Chemical Kinetics Package for the Analysis of Gas-Phase Chemical Kinetics". Technical Report SAND89-8009B, Sandia National Laboratories, Livermore, 1991.

[170] Kee, R. J., G. Dixon-Lewis, J. Warnatz, M. E. Coltrin, and J. A. Miller. "A Fortran Computer Code Package for the Evaluation of Gas-Phase Multicomponent Transport Properties". Technical Report SAND86-8246, Sandia National Laboratories, Livermore, 1991.

[171] Hirschfelder, J. O., C. F. Curtiss, and R. B. Bird. Molecular Theory of Gases and Liquids. Wiley, 1954.

[172] Kleijn, C. R., Th. H. van der Meer, and C. J. Hoogendoorn. "A Mathematical Model for LPCVD in a Single Wafer Reactor". Journal of the Electrochemical Society, 136(11):3423–4333, 1989.

[173] Kee, R. J., F. M. Rupley, and J. A. Miller. "The Chemkin Thermodynamic Data Base". Technical Report SAND87-8215B, Sandia National Laboratories, Livermore, 1992.

[174] Baliga, B. R. and S. V. Patankar. "A New Finite-Element Formulation for Convection-Diffusion Problems". Numerical Heat Transfer, 3(4):393–409, 1980.

[175] Baliga, B. R.
"A Control Volume Based Finite Element Method for Convective Heat and Mass Transfer". PhD thesis, University of Minnesota, Minneapolis, MN, 1978.

[176] Baliga, B. R. and S. V. Patankar. "A Control Volume Finite-Element Method for Two-Dimensional Fluid Flow and Heat Transfer". Numerical Heat Transfer, 6(3):245–261, 1983.

[177] Schneider, G. E. and M. Zedan. A Control Volume Based Finite Element Formulation of the Heat Conduction Equation. In H. E. Collicott and P. E. Bauer, editor, Spacecraft Thermal Control, Design and Operation, volume 86, pages 305–327. Progress in Astronautics and Aeronautics, 1983.

[178] Schneider, G. E. and M. J. Raw. "A Skewed, Positive Influence Coefficient Upwinding Procedure for Control-Volume-Based Finite-Element Convection-Diffusion Computation". Numerical Heat Transfer, 9(1):1–26, 1986.

[179] Swaminathan, C. R. and V. R. Voller. "Streamline Upwind Scheme for Control-Volume Finite Elements, Part I. Formulations". Numerical Heat Transfer, Part B, 22(1):95–107, 1992.

[180] Patankar, S. V. Numerical Heat Transfer and Fluid Flow. Hemisphere, 1980.

[181] Winslow, A. M. "Numerical Solution of the Quasilinear Poisson Equation in a Nonuniform Triangle Mesh". Journal of Computational Physics, 1(2):149–172, 1966.

[182] Ramadhyani, S. and S. V. Patankar. "Solution of the Poisson Equation: Comparison of the Galerkin and Control-Volume Methods". International Journal for Numerical Methods in Engineering, 15:1395–1418, 1980.

[183] Baliga, B. R., T. T. Pham and S. V. Patankar. "Solution of Some Two-Dimensional Incompressible Fluid Flow and Heat Transfer Problems Using a Control Volume Finite-Element Method". Numerical Heat Transfer, 6(3):263–282, 1983.

[184] Prakash, C. and S. V. Patankar. "A Control Volume-Based Finite-Element Method for Solving the Navier-Stokes Equations Using Equal-Order Velocity-Pressure Interpolation". Numerical Heat Transfer, 8(3):259–280, 1985.

[185] Ramadhyani, S. and S. V. Patankar. "Solution of the Convection-Diffusion Equation by a Finite-Element Method Using Quadrilateral Elements". Numerical Heat Transfer, 8(5):595–612, 1985.

[186] Raithby, G. D. "Skew Upstream Differencing Schemes for Problems Involving Fluid Flow". Computer Methods in Applied Mechanics and Engineering, 9:153–164, 1976.

[187] Raw, M. J.
"A New Control-Volume-Based Finite Element Procedure for the Numerical Solution of the Fluid Flow and Scal
PhD thesis, University of Waterloo, Ontario, Canada, 1985.

[188] LeDain-Muir, B. and B. R. Baliga. "Solution of Three-Dimensional Convection-Diffusion Problems Using Tetrahedral Elements and Flow-Oriented Upwind Interpolation Functions". Numerical Heat Transfer, 9(2):143–162, 1986.

[189] Prakash, C. "An Improved Control Volume Finite-Element Method for Heat and Mass Transfer, and for Fluid Flow Using Equal-Order Velocity-Pressure Interpolation". Numerical Heat Transfer, 9(3):253–276, 1986.

[190] Prakash, C. "Examination of the Upwind (Donor-Cell) Formulation in Control Volume Finite Element Methods for Fluid Flow and Heat Transfer". Numerical Heat Transfer, 11(4):401–416, 1987.

[191] Hookey, N. A., B. R. Baliga and C. Prakash. "Evaluation and Enhancements of Some Control Volume Finite-Element Methods–1. Convection-Diffusion Problems". Numerical Heat Transfer, 14(3):255–272, 1988.

[192] Hookey, N. A. and B. R. Baliga. "Evaluation and Enhancements of Some Control Volume Finite-Element Methods–1. Incompressible Fluid Flow Problems". Numerical Heat Transfer, 14(3):273–293, 1988.

[193] van Doormaal, J. P. and G. D. Raithby. "Enhancements of the SIMPLE Method for Predicting Incompressible Fluid Flows". Numerical Heat Transfer, 7:147–163, 1984.

[194] Swaminathan, C. R. and V. R. Voller. "Streamline Upwind Scheme for Control-Volume Finite Elements, Part II. Implementaion and Comparison with the SUPG Finite-Element Scheme". Numerical Heat Transfer, Part B, 22(1):109–124, 1992.

[195] Brooks, A. N. and T. J. R. Hughes. "Streamline Upwind/Petrov-Galerkin Formulations for Convection Dominated Flows with Particular Emphasis on the Incompressible Navier-Stokes Equations". Computer Methods in Applied Mechanics and Engineering, 32:199–259, 1982.

[196] Baliga, B. R. and H. J. Saabas. "Control-Volume Finite Element Methods for Incompressible Fluid Flow". Invited Keynote Lecture on Fluid Mechanics, III Portuguese Conference on Computational Mechanics, Coimbra, Portugal, September 1992.

[197] Naterer, G. F. and G. E. Schneider. "Physical Influences of Integration Point Equations On a Control-Volume-Based Finite Element Method for Compressible Flows". AIAA Paper 92-0365, 30th Aerospace Sciences Meeting, Reno, NV, January 1992.

[198] Swaminathan, C. R., V. R. Voller and S. V. Patankar. "A Streamline Upwind Control Volume Finite Element Method for Modeling Fluid Flow and Heat Transfer Problems". Finite Elements in Analysis and Design, 13(2-3):169–184, 1993.

[199] Saabas, H. J. and B. R. Baliga. "Co-Located Equal-Order Control-Volume Finite-Element Method for Multidimensional, Incompressible, Fluid Flow–Part I: Formulation". Numerical Heat Transfer, Part B, 26(4):381–407, 1994.

[200] Saabas, H. J. and B. R. Baliga. "Co-Located Equal-Order Control-Volume Finite-Element Method for Multidimensional, Incompressible, Fluid Flow–Part II: Verification". Numerical Heat Transfer, Part B, 26(4):409–424, 1994.

[201] Masson, C., H. J. Saabas and B. R. Baliga. "Co-Located Equal-Order Control-Volume Finite Element Method for Two-Dimensional Axisymmetric Incompressible Fluid Flow". Internation Journal for Numerical Methods in Fluids, 18(1):1–26, 1994.

[202] Masson, C. and B. R. Baliga. "A Control Volume Finite-Element Method for Dilute Gas-Solid Particle Flows". Computers and Fluids, 23(8):1073–1096, 1994.

[203] Karimian, S. M. H. and G. E. Schneider. "Numerical Solution of Two-Dimensional Incompressible Navier-Stokes Equations: Treatment of Velocity-Pressure Coupling". AIAA Paper 94-2359, 25th AIAA Fluid Dynamics Conference, Colorado Springs, CO, June 1994.

[204] Karimian, S. M. H. and G. E. Schneider. "Pressure-Based Computational Method for Compressible and Incompressible Flows". Journal of Thermophysics and Heat Transfer, 8(2):267–274, 1994.

[205] Deng, G. B., J. Piquet, P. Queutey and M. Visonneau. "Incompressible Flow Calculations With a Consistent Physical Interpolation Finite Volume Approach". Computers and Fluids, 23(8):1029–1047, 1994.

[206] Costa, V. A. F., L. A. Oliveira and A. R. Figueiredo. "A Control-Volume Based Finite Element Method for Three-Dimensional Incompressible Turbulent Fluid Flow, Heat Transfer, and Related Phenomena". International Journal for Numerical Methods in Fluids, 21(7):591–613, 1995.

[207] Karimian, S. M. H. and G. E. Schneider. "Application of a Control-Volume-Based Finite-Element Formulation to the Shock Tube Problem (TN)". AIAA Journal, 33(1):165–, 1995.

[208] Karimian, S. M. H. and G. E. Schneider. "Pressure-Based Control-Volume Finite Element Method for Flow at All Speeds". AIAA Journal, 33(9):1611–1618, September 1995.

[209] Padra, C. and A. Larreteguy. "A-Posteriori Error Estimator for the Control-Volume Finite-Element Method as Applied to Convection-Diffusion Problem". Numerical Heat Transfer, Part B, 27(1):63–80, 1995.

[210] Larreteguy, A. E. "An Equal-Order Control-Volume Finite-Element Method for Fluid Flow in Arbitrary Triangulations". Numerical Heat Transfer, Part B, 28:401–413, 1995.

[211] Harms, T. M., T. W. von Backström and J. P. du Plessis. "Simplified Control-Volume Finite-Element Method". Numerical Heat Transfer, Part B, 30(2):179–194, 1996.

[212] Comini, G., S. Del Giudice, and C. Nonino. "Energy Balances in CVFEM and GFEM Formulations of Convection-Type Problems". International Journal for Numerical Methods in Engineering, 39(13):2249–2263, 1996.

[213] Neises, J. and I. Steinbach. "Finite Element Integration for the Control Volume Method". Communications in Numerical Methods in Engineering, 12(9):543–556, 1996.

[214] Völker, S., T. Burton and S. P. Vanka. "Finite-Volume Multigrid Calculation of Natural-Convection Flows on Unstructured Grids". Numerical Heat Transfer, Part B, 30(1):1–22, 1996.

[215] Botta, N. and D. Hempel. "Finite-Volume Projection Method for Incompressible Flows on Triangular Grids". Hamburger Beiträge zur Angewandten Mathematik, Reihe A, Preprint 110, September 1996.

[216] Darbandi, M. and G. E. Schneider. "Momentum Variable Procedure for Solving Compressible and Incompressible Flows". AIAA Journal, 35(12):1801–1805, 1997.

[217] B. R. Baliga. "Control-Volume Finite Element Methods for Fluid Flow and Heat Transfer". In Minkowycz, W. J. and E. M. Sparrow, editor, Advances in Numerical Heat Transfer, Volume 1, pages 97–135. Taylor and Francis, 1997.

[218] O'Rourke, P. J., and M. S. Sahota. "A Variable Explicit/Implicit Numerical Method for Calculating Advection on Unstructured Meshes". Journal of Computational Physics, 143:312–345, 1998.

[219] Gresho, P. M. and R. L. Sani. Incompressible Flow and the Finite Element Method. John Wiley and Sons, 1998.

[220] Venditti, D. A. and B. R. Baliga. "An h-Adaptive Strategy for CVFEM Simulations of Viscous Incompressible Flow". Proc. 6th Annual Conference of the Computational Fluid Dynamics Society of Canada (CFD 98), pp. VIII-65–VIII-70, Quebec City, Canada, June 7-9, 1998.

[221] Reyes, M., J. Rincon, J. and J. Damia. "Simulation of Turbulent Flow in Irregular Geometries Using a Control-Volume Finite-Element Method". Numerical Heat Transfer, Part B, 39(1):79–90, 2001.

[222] Campos Silva, J. B. and L. F. M. de Moura. "A Control-Volume Finite-Element Method (CVFEM) for Unsteady, Incompressible, Viscous Fluid Flows". Numerical Heat Transfer, Part B, 40(1):61–82, 2001.

[223] Zhao, Y., J. Tai, and F. Ahmed . "Simulation of Micro Flows with Moving Boundaries Using High-Order Upwind FV Method on Unstructured Grids". Computational Mechanics, 28:66–75, 2002.

[224] Kettleborough, C. F., S. R. Husain, and C. Prakash. "Solution of Fluid Flow Problems with the Vorticity-Streamfunction Formulation and the Control Volume Based Finite-Element Method". Numerical Heat Transfer, Part B, 16(1):31–58, 1989.

[225] Choudhury, S. and R. A. Nicolaides. "Discretization of Incompressible Vorticity-Velocity Equations on Triangular Meshes". Internation Journal for Numerical Methods in Fluids, 11(6):823–, 1990.

[226] Krakov, M. S. "Control Volume Finite-Element Method for Navier-Stokes Equations in Vortex-Streamfunction Formulation". Numerical Heat Transfer, Part B, 21(2):125–145, 1992.

[227] Elkaim, D., M. Reggio, and R. Camarero. "Simulating Two-Dimensional Turbulent Flow by Using the k-$\epsilon$ Model and the Vorticity-Streamfunction Formulation". International Journal for Numerical Methods in Fluids, 14(8):961–980, 1992.

[228] Elkaim, D., F. McKenty, M. Reggio, and R. Camarero. "Control Volume Finite Element Solution of Confined Turbulent Swirling Flows". International Journal for Numerical Methods in Fluids, 19(2):135–152, 1994.

[229] Banaszek, J. A. "A Conservative Finite Element Method for Heat Conduction Problems". International Journal for Numerical Methods in Engineering, 20:2033–2050, 1984.

[230] Blackwell, B. F. "Numerical Prediction of One-Dimensional Ablation Using a Finite Control Volume Procedure with Exponential Differencing". Numerical Heat Transfer, 14:17–34, 1988.

[231] Abboud, J. B. and H. Hardisty. "Control-Volume Energy-Balance FE Formulations of the 8-Node Hexahedron Element". Communications in Applied Numerical Methods, 7:141–153, 1991.

[232] Blackwell, B. F. and R. E. Hogan. "Numerical Solution of Axisymmetric Heat Conduction Problems Using Finite Control Volume Technique". Journal of Thermophysics and Heat Transfer, 7:462–471, 1993.

[233] Blackwell, B. F. and R. E. Hogan. "One-Dimensional Ablation Using Landau Transformation and Finite Control Volume Procedure". Journal of Thermophysics and Heat Transfer, 8(2):282–287, 1994.

[234] Ferguson, W. J. and I. W. Turner. "Control Volume Finite Element Model of Mechano-Sorptive Creep in Timber". Numerical Heat Transfer, Part A, 29(2):147–164, 1996.

[235] Ferguson, W. J. and I. W. Turner. "A Control Volume Finite Element Numerical Simulation of the Drying of Spruce". Journal of Computational Physics, 125(1):59–70, 1996.

[236] Ferguson, W. J. "A Control Volume Finite Element Numerical Solution of Creep Problems". International Journal for Numerical Methods in Engineering, 40(18):3463–, 1997.

[237] Letniowski, F. W. and P. A. Forsyth. "A Control Volume Finite Element Method for Three-Dimensional NAPL Groundwater Contamination". International Journal for Numerical Methods in Fluids, 13(8):955–970, 1991.

[238] Fung, L. S. K., A. D. Hiebert, and L. X. Nghiem. "Reservoir Simulation With a Control-Volume Finite Element Method". SPE Reservoir Engineering, 7(3):349–, 1992.

[239] Durlofsky, L. J. "Accuracy of Mixed and Control Volume Finite Element Approximations to Darcy Velocity and Related Quantities". Water Resources Research, 30(4):965–973, 1994.

[240] Eymard, R. and F. Sonier. "Mathematical and Numerical Properties of Control-Volume, Finite-Element Scheme for Reservoir Simulation". SPE Reservoir Engineering, 9(4):283–, 1994.

[241] Fung, L. S. K., L. Buchanan, and R. Sharma. "Hybrid-CVFE Method for Flexible-Grid Reservoir Simulation". SPE Reservoir Engineering, 9(3):188–194, 1994.

[242] Jones, J. E., Z. Cai, S. F. McCormick, and T. F. Russell. "Control-Volume Mixed Finite Element Methods". Technical Report TR-97-16, ICASE, Langley, VA, February 1997.

[243] Gottardi, G. and M. Venuttelli. "A Control-Volume Finite-Element Model for Two-Dimensional Overland Flow". Advances in Water Resources, 16(5):277–, 1993.

[244] Di Giammarco, P., E. Todini and P. Lamberti. "A Conservative Finite Elements Approach to Overland Flow: The Control Volume Finite Element Formulation". Journal of Hydrology, 175(1-4):267–291, 1996.

[245] Oñate, E., M. Cervera, and O. C. Zienkiewicz. "A Finite Volume Method for Structural Mechanics". Internation Journal for Numerical Methods in Engineering, 37:181–201, 1994.

[246] Bailey, C. and M. Cross. "A Finite Volume Procedure to Solve Elastic Solid Mechanics Problems in Three Dimensions on an Unstructured Mesh". Internation Journal for Numerical Methods in Engineering, 38:1757–1776, 1995.

[247] Evans, G. H. " Turbulence Modeling with v2-f Transport Equations". Technical report, Sandia National Laboratories, Livermore, CA, 1999.

[248] Evans, G. H. " Turbulent Convective Heat Transfer with Applications to Sandia Fire Modeling". Technical report, Sandia National Laboratories, Livermore, CA, 2000.

[249] Winters, W. S., G. H. Evans, and C. D. Moen. "CURRENT - A Computer Code for Modelling Two-Dimensional, Chemically Reacting, Low Mach Number Flows". Technical Report SAND Report 97–8202, Sandia National Laboratories, Livermore, CA, October 1996.

[250] Durbin, P. A. e-mail communication, October 8, 1999.

[251] Patel, V. C., W. Rodi, and G. Scheuerer. "Turbulence Models for Near-Wall and Low Reynolds Number Flows: A Review". AIAA Journal, 23:1308–1319, 1985.

[252] Durbin, P. A. "Application of a Near-Wall Turbulence Model to Boundary Layers and Heat Transfer". International Journal of Heat and Fluid Flow, 14:316–323, 1993.

[253] Durbin, P. A. "Separated Flow Computations with the $k - \epsilon - v^2$ Model". AIAA Journal, 33:659–664, 1995.

[254] Parneix, S., M. Behnia, and P. A. Durbin. "Predictions of Turbulent Heat Transfer in an Axisymmetric Jet Impinging on a Heated Pedestal". Journal of Heat Transfer, 121:43–49, 1999.

[255] Behnia, M., S. Parneix, and P. A. Durbin. "Prediction of Heat Transfer in an Axisymmetric Turbulent Jet Impinging on a Flat Plate". International Journal of Heat and Mass Transfer, 41:1845–1855, 1998.

[256] Hussain, A. K. M. F. and W. C. Reynolds. "Measurements in Fully Developed Turbulent Channel Flow". Trans. ASME J. Fluids Engineering, pages 568–580, December, 1975.

[257] Moin, P. and J. Kim. "Numerical investigation of turbulent channel flow". J. Fluid Mech., 118:341–377, 1982.

[258] Cox, G. "Turbulent Closure and the Modeling of Fire by Using Computational Fluid Dynamics". Philosophical Transactions of the Royal Society of London, A, 356:2835–2854, 1998.

[259] Chomiak, J. and J. R. Nisbet. Combustion and Flame, 102:371–386, 1995.

# 7. TRANSPORT PROCESSES

We provide detailed derivations of the approximate form of the transport equations.

## APPENDIX A. MULTICOMPONENT TRANSPORT

Gas-phase mass transport and chemical reactions are modeled with the multicomponent transport equations. The gas-phase species transport equations are:

$$\frac{\partial \rho Y_g}{\partial t} + \frac{\partial \rho u_j Y_g}{\partial x_j} = -\frac{\partial \rho \hat{u}_{j,g} Y_g}{\partial x_j} + \dot{\omega}_g, \tag{7.1}$$

where the summation rule has been suspended for the species index, $g$. The mass fractions of the chemical species are $Y_g$, the chemical source terms are $\dot{\omega}_g$, and the diffusion velocities are $\hat{u}_{i,g}$. The diffusion velocities are functions of both mass diffusion and thermal diffusion, and are defined by the multicomponent diffusion equation [171, 13]. Diffusion due to pressure gradients or body forces is neglected. The diffusion equation can be manipulated into a form that is more readily applied algorithmically [172]. The mass diffusion flux is defined as $\mathrm{j}_{i,g} = -\rho \hat{u}_{i,g} Y_g$.

$$\begin{aligned} \mathrm{j}_{i,g} = -\rho \overline{D}_g \frac{\partial Y_g}{\partial x_i} \quad &- \left( \frac{\rho Y_g \overline{D}_g}{W} \frac{\partial W}{\partial x_i} + D_g^T \frac{1}{T} \frac{\partial T}{\partial x_i} \right. \\ &- \left. Y_g \overline{D}_g W \sum_{n=1; n \neq g}^{NS} \left[ \frac{\mathrm{j}_{i,n}}{W_n \mathcal{D}_{ng}} + \frac{{D_n}^T}{W_n \mathcal{D}_{ng}} \frac{1}{T} \frac{\partial T}{\partial x_i} \right] \right) \end{aligned} \tag{7.2}$$

The multicomponent diffusion coefficients are $\overline{D}_g$, the binary diffusion coefficients are $\mathcal{D}_{ij}$, the thermal diffusion coefficients are ${D_i}^T$, and the molecular weight is $W$. The multicomponent diffusion coefficients are defined to be

$$\overline{D}_i = \left[ \sum_{j=1; j \neq i}^{NS} \frac{x_j}{\mathcal{D}_{ij}} \right]^{-1} \tag{7.3}$$

The modified form of the equations helps decouple the equations for a segregated solution approach. Equation 7.2 must first be solved for the mass diffusion fluxes as a closure equation. The equations are not linearly independent over all the species, so one equation must be replaced with the constraint that $\sum \mathrm{j}_{i,g} = 0$. The chemical properties and rate terms are computed using CHEMKIN [169, 170, 173].

For coupled heat and mass transfer, the heat flux term in the energy equation is modified:

$$q_i = -k \frac{\partial T}{\partial x_i} - \sum_g \mathrm{j}_{i,g} h_g \tag{7.4}$$

The term involving the Dufour effect from thermal diffusion is neglected.

# APPENDIX B. TIME-AVERAGING AND FAVRE-AVERAGING

The time-averaged and Favre-averaged transport equations are given in the following section.

## B.1. Conservation of Mass

The continuity equation:

(a) time averaged:

$$\int \frac{\partial \overline{\rho}}{\partial t} \mathrm{d}V + \int \overline{\rho u}_j n_j \mathrm{d}S + \int \overline{\rho' u'_j} n_j \mathrm{d}S = 0 \tag{7.5}$$

(b) Favre averaged:

$$\int \frac{\partial \overline{\rho}}{\partial t} \mathrm{d}V + \int \overline{\rho} \tilde{u}_j n_j \mathrm{d}S = 0 \tag{7.6}$$

## B.2. Conservation of Momentum

The momentum transport equations:

(a) time averaged:

$$\begin{aligned} \int \frac{\partial \overline{\rho u}_i}{\partial t} \mathrm{d}V + \int \frac{\partial \overline{\rho' u'_i}}{\partial t} \mathrm{d}V \quad &+ \quad \int \overline{\rho}\, \overline{u}_i \overline{u}_j n_j \mathrm{d}S + \int \overline{\rho} \overline{u'_i u'_j} n_j \mathrm{d}S + \int \overline{u_j} \overline{\rho' u'_i} n_j \mathrm{d}S \\ &+ \quad \int \overline{u_i} \overline{\rho' u'_j} n_j \mathrm{d}S + \int \overline{\rho' u'_i u'_j} n_j \mathrm{d}S + \int \overline{p} n_i \mathrm{d}S \\ &= \quad \int \overline{\tau}_{ij} n_j \mathrm{d}S + \int \overline{\rho} g_i \mathrm{d}V \end{aligned} \tag{7.7}$$

(b) Favre averaged:

$$\int \frac{\partial \overline{\rho} \tilde{u}_i}{\partial t} \mathrm{d}V + \int \overline{\rho} \tilde{u}_i \tilde{u}_j n_j \mathrm{d}S + \int \overline{p} n_i \mathrm{d}S = \int \overline{\tau}_{ij} n_j \mathrm{d}S - \int \overline{\rho u''_i u''_j} n_j \mathrm{d}S + \int \overline{\rho} g_i \mathrm{d}V \tag{7.8}$$

## B.3. Conservation of Energy

The energy transport equation, assume Lewis number is one (see Section 2.5.3):

(a) time averaged:

(b) Favre averaged:

$$\int \frac{\partial \overline{\rho}\tilde{h}}{\partial t}\mathrm{d}V + \int \overline{\rho}\tilde{h}\tilde{u}_j n_j \mathrm{d}S = \int \overline{\frac{\kappa}{C_p}\frac{\partial h}{\partial x_j}} n_j \mathrm{d}S - \int \overline{\rho h'' u_j''} n_j \mathrm{d}S - \int \frac{\partial \overline{q_i^r}}{\partial x_i}\mathrm{d}V \tag{7.9}$$

## B.4. Conservation of Species

The species transport equation:

(a) time averaged:

$$\begin{aligned}\int \frac{\partial \overline{\rho}\overline{Y}_k}{\partial t}\mathrm{d}V \quad &+ \quad \int \frac{\partial \overline{\rho' Y_k'}}{\partial t}\mathrm{d}V + \int \overline{\rho}\overline{Y}_k \overline{u}_j n_j \mathrm{d}S + \int \overline{\rho}\overline{Y_k' u_j'} n_j \mathrm{d}S + \int \overline{u_j}\overline{\rho' Y_k'} n_j \mathrm{d}S \\ &+ \quad \int \overline{Y_k}\overline{\rho' u_j'} n_j \mathrm{d}S + \int \overline{\rho' Y_k' u_j'} n_j \mathrm{d}S \\ &= \quad \int \overline{\rho Y_k \hat{u}_{j,k}} n_j \mathrm{d}S + \int \overline{\dot{\omega}_k}\mathrm{d}V \end{aligned} \tag{7.10}$$

(b) Favre averaged:

$$\int \frac{\partial \overline{\rho}\tilde{Y}_k}{\partial t}\mathrm{d}V + \int \overline{\rho}\tilde{Y}_k \tilde{u}_j n_j \mathrm{d}S = -\int \overline{\rho Y_k'' u_j''} n_j \mathrm{d}S + \int \overline{\rho Y_k \hat{u}_{j,k}} n_j \mathrm{d}S + \int \overline{\dot{\omega}_k}\mathrm{d}V \tag{7.11}$$

# APPENDIX C. DISCRETE 2D/AXISYMMETRIC TRANSPORT EQUATIONS

The transport equations for two-dimensional, axisymmetric flow are given in section 2.2.5. These equation descriptions are a work–in–progress; caveat emptor.

## C.1. X-Momentum (axial), 2D Laminar Transport

The time term is lumped. The time-term contribution is evaluated for each sub-volume.

$$A^t_{I,I} \mathrel{+}= \rho^*_I \frac{\Delta V_I}{\Delta t} \tag{7.12}$$

$$b^t_I \mathrel{-}= \left(\rho^*_I U^*_I - \rho^n_I U^n_I\right) \frac{\Delta V_I}{\Delta t} \tag{7.13}$$

The convection term is computed at each face $k$ and assembled to the left (IL) and right (IR) control volumes.

$$A^c_{IL,J} \mathrel{+}= C^*_{k,J} \tag{7.14}$$
$$A^c_{IR,J} \mathrel{-}= C^*_{k,J} \tag{7.15}$$

$$b^c_{IL} \mathrel{-}= \sum_J C^*_{k,J} U^*_J \tag{7.16}$$
$$b^c_{IR} \mathrel{+}= \sum_J C^*_{k,J} U^*_J \tag{7.17}$$

The viscous stress term is computed at each face $k$ and assembled to the left (IL) and right (IR) control volumes. Only the solenoidal part of the stress term is used for the matrix. The stress term may or may not include the molecular viscosity, depending on the user specified model.

$$F_{k,J} = -\mu_k \left( \left.\frac{\partial N_J}{\partial x}\right|_k A_x + \left.\frac{\partial N_J}{\partial y}\right|_k A_y \right) \tag{7.18}$$
$$A^d_{IL,J} \mathrel{+}= F_{k,J} \tag{7.19}$$
$$A^d_{IR,J} \mathrel{-}= F_{k,J} \tag{7.20}$$

$$\tau_{xx} = \mu_k \left(u^*_x + u^*_x\right) \tag{7.21}$$
$$\tau_{xy} = \mu_k \left(u^*_y + v^*_x\right) \tag{7.22}$$
$$f_k = -\left(\tau_{xx} A_x + \tau_{xy} A_y\right) \tag{7.23}$$
$$b^d_{IL} \mathrel{-}= f_k \tag{7.24}$$
$$b^d_{IR} \mathrel{+}= f_k \tag{7.25}$$

The pressure is assembled in the form of a volume integral. The pressure gradients have been pre-computed at nodes use a surface-integral approximation.

$$b_I^s \; -= \; \left.\frac{\partial P}{\partial x}\right|_I^* \Delta V_I \tag{7.26}$$

## C.2. Y-Momentum (radial), 2D Laminar Transport

The time term is lumped. The time-term contribution is evaluated for each sub-volume.

$$A_{I,I}^t \; += \; \rho_I^* \frac{\Delta V_I}{\Delta t} \tag{7.27}$$

$$b_I^t \; -= \; \left(\rho_I^* V_I^* - \rho_I^n V_I^n\right) \frac{\Delta V_I}{\Delta t} \tag{7.28}$$

The convection term is computed at each face $k$ and assembled to the left (IL) and right (IR) control volumes.

$$A_{IL,J}^c \; += \; C_{k,J}^* \tag{7.29}$$
$$A_{IR,J}^c \; -= \; C_{k,J}^* \tag{7.30}$$

$$b_{IL}^c \; -= \; \sum_J C_{k,J}^* V_J^* \tag{7.31}$$
$$b_{IR}^c \; += \; \sum_J C_{k,J}^* V_J^* \tag{7.32}$$

The viscous stress term is computed at each face $k$ and assembled to the left (IL) and right (IR) control volumes. Only the solenoidal part of the stress term is used for the matrix.

$$F_{k,J} \; = \; -\mu_k \left( \left.\frac{\partial N_J}{\partial x}\right|_k A_x + \left.\frac{\partial N_J}{\partial y}\right|_k A_y \right) \tag{7.33}$$
$$A_{IL,J}^d \; += \; F_{k,J} \tag{7.34}$$
$$A_{IR,J}^d \; -= \; F_{k,J} \tag{7.35}$$

$$\tau_{yx} = \mu_k \left( v_x^* + u_y^* \right) \tag{7.36}$$
$$\tau_{yy} = \mu_k \left( v_y^* + v_y^* \right) \tag{7.37}$$
$$f_k = -\left( \tau_{yx} A_x + \tau_{yy} A_y \right) \tag{7.38}$$
$$b_{IL}^d -= f_k \tag{7.39}$$
$$b_{IR}^d += f_k \tag{7.40}$$

There is a radial force contribution from the azimuthal stresses. These are evaluated for sub-volumes.

$$F_{k,J} = 2\mu_k \frac{A_c}{y} \tag{7.41}$$
$$A_{IC,J}^d += F_{k,J} \tag{7.42}$$
$$\tag{7.43}$$

$$\tau_{\theta\theta} = 2\mu_k \frac{v}{y} \tag{7.44}$$
$$f_k = \tau_{\theta\theta} A_c \tag{7.45}$$
$$b_{IC}^d -= f_k \tag{7.46}$$

There is an acceleration force from swirl. These are evaluated for sub-volumes.

$$b_{IC}^s += \rho W^2 A_c \tag{7.47}$$

The pressure is assembled in the form of a volume integral. The pressure gradients have been pre-computed at nodes use a surface-integral approximation.

$$b_I^s -= \left. \frac{\partial P}{\partial y} \right|_I^* \Delta V_I \tag{7.48}$$

## C.3. $\theta$-Momentum (swirl), 2D Laminar Transport

The time term is lumped. The time-term contribution is evaluated for each sub-volume. We solve for the angular velocity, $\Omega$, instead of the azimuthal velocity, $w$.

$$A^t_{I,I} \mathrel{+}= \rho^*_I \frac{\Delta V_I}{\Delta t} \tag{7.49}$$

$$b^t_I \mathrel{-}= \left(\rho^*_I \Omega^*_I - \rho^n_I \Omega^n_I\right) \frac{\Delta V_I}{\Delta t} \tag{7.50}$$

The convection term is computed at each face $k$ and assembled to the left (IL) and right (IR) control volumes.

$$A^c_{IL,J} \mathrel{+}= C^*_{k,J} \tag{7.51}$$
$$A^c_{IR,J} \mathrel{-}= C^*_{k,J} \tag{7.52}$$

$$b^c_{IL} \mathrel{-}= \sum_J C^*_{k,J} \Omega^*_J \tag{7.53}$$
$$b^c_{IR} \mathrel{+}= \sum_J C^*_{k,J} \Omega^*_J \tag{7.54}$$

The viscous stress term is computed at each face $k$ and assembled to the left (IL) and right (IR) control volumes. Only the solenoidal part of the stress term is used for the matrix.

$$F_{k,J} = -\mu_k \left( \left.\frac{\partial N_J}{\partial x}\right|_k A_x + \left.\frac{\partial N_J}{\partial y}\right|_k A_y \right) \tag{7.55}$$
$$A^d_{IL,J} \mathrel{+}= F_{k,J} \tag{7.56}$$
$$A^d_{IR,J} \mathrel{-}= F_{k,J} \tag{7.57}$$

$$\tau_{zx} = \mu_k \left(\omega^*_x\right) \tag{7.58}$$
$$\tau_{zy} = \mu_k \left(\omega^*_y\right) \tag{7.59}$$
$$f_k = -\left(\tau_{zx} A_x + \tau_{zy} A_y\right) \tag{7.60}$$
$$b^d_{IL} \mathrel{-}= f_k \tag{7.61}$$
$$b^d_{IR} \mathrel{+}= f_k \tag{7.62}$$

$$b_I^d += 2\mu \frac{1}{r} \frac{\partial \Omega}{\partial r} \Delta V_I \tag{7.63}$$

There is a Coriolis force from swirl. These are evaluated for sub-volumes.

$$b_{IC}^s -= 2\rho V W A_c \tag{7.64}$$

## C.4. Mass Transport – 2D Continuity

There is no net flow through the azimuthal face if an axisymmetric coordinate system is used.

The time term is lumped. The time-term contribution is evaluated for each sub-volume.

$$b_I^t -= (\rho_I^* - \rho_I^n) \frac{\Delta V_I}{\Delta t} \tag{7.65}$$

The convection term is computed at each face $k$ and assembled to the left (IL) and right (IR) control volumes using the Rhie/Chow scheme from Section 5.1.

$$F_{k,J} = -f \Delta t \left( \left. \frac{\partial N_J}{\partial x} \right|_k A_x + \left. \frac{\partial N_J}{\partial y} \right|_k A_y \right) \tag{7.66}$$

$$A_{IL,J}^d += F_{k,J} \tag{7.67}$$

$$A_{IR,J}^d -= F_{k,J} \tag{7.68}$$

$$u_k^* = \sum_J N_J|_k U_J^* + f \frac{\Delta t}{\rho} \left( \sum_J \left. \frac{\partial P}{\partial x} \right|_J^* - p_x^* \right) + f \left( u_k^n - \sum_J N_J|_J U_J^n \right) \tag{7.69}$$

$$v_k^* = \sum_J N_J|_k V_J^* + f \frac{\Delta t}{\rho} \left( \sum_J \left. \frac{\partial P}{\partial y} \right|_J^* - p_y^* \right) + f \left( v_k^n - \sum_J N_J|_J V_J^n \right) \tag{7.70}$$

$$\dot{m}_k = \rho (u_k^* A_x + v_k^* A_y) \tag{7.71}$$

$$b_{IL}^c -= \dot{m}_k \tag{7.72}$$

$$b_{IR}^c += \dot{m}_k \tag{7.73}$$

Velocity correction and new mass flow rate.....

## C.5. Energy, 2D Laminar Transport

The laminar energy equation is linearized with respect to the temperature. The time term is lumped. The time-term contribution is evaluated for each sub-volume. The density must also be linearized for stability.

$$A^t_{I,I} \mathrel{+}= \left( \rho^*_I C^*_{p,I} - \rho^*_I \frac{H^*_I}{T^*_I} \right) \frac{\Delta V_I}{\Delta t} \tag{7.74}$$

$$b^t_I \mathrel{-}= \left( \rho^*_I H^*_I - \rho^n_I H^n_I \right) \frac{\Delta V_I}{\Delta t} \tag{7.75}$$

The convection term is computed at each face $k$ and assembled to the left (IL) and right (IR) control volumes.

$$A^c_{IL,J} \mathrel{+}= C^{n+1}_{k,J} C^*_{p,J} \tag{7.76}$$

$$A^c_{IR,J} \mathrel{-}= C^{n+1}_{k,J} C^*_{p,J} \tag{7.77}$$

$$b^c_{IL} \mathrel{-}= \sum_J C^{n+1}_{k,J} H^*_J \tag{7.78}$$

$$b^c_{IR} \mathrel{+}= \sum_J C^{n+1}_{k,J} H^*_J \tag{7.79}$$

The heat conduction term is computed at each face $k$ and assembled to the left (IL) and right (IR) control volumes.

$$F_{k,J} = -\kappa_k \left( \left. \frac{\partial N_J}{\partial x} \right|_k A_x + \left. \frac{\partial N_J}{\partial y} \right|_k A_y \right) \tag{7.80}$$

$$A^d_{IL,J} \mathrel{+}= F_{k,J} \tag{7.81}$$

$$A^d_{IR,J} \mathrel{-}= F_{k,J} \tag{7.82}$$

$$q_k = -\kappa_k \left( t^*_x A_x + t^*_y A_y \right) \tag{7.83}$$

$$b^d_{IL} \mathrel{-}= q_k \tag{7.84}$$

$$b^d_{IR} \mathrel{+}= q_k \tag{7.85}$$

## C.6. Temperature, 2D Laminar Transport

The laminar temperature equation is linearized with respect to the temperature. The time term is lumped. The time-term contribution is evaluated for each sub-volume.

$$A^t_{I,I} \mathrel{+}= \rho^*_I \frac{\Delta V_I}{\Delta t} \tag{7.86}$$

$$b^t_I \mathrel{-}= \left(\rho^*_I T^*_I - \rho^n_I T^n_I\right) \frac{\Delta V_I}{\Delta t} \tag{7.87}$$

The convection term is computed at each face $k$ and assembled to the left (IL) and right (IR) control volumes.

$$A^c_{IL,J} \mathrel{+}= C^{n+1}_{k,J} \tag{7.88}$$

$$A^c_{IR,J} \mathrel{-}= C^{n+1}_{k,J} \tag{7.89}$$

$$b^c_{IL} \mathrel{-}= \sum_J C^{n+1}_{k,J} T^*_J \tag{7.90}$$

$$b^c_{IR} \mathrel{+}= \sum_J C^{n+1}_{k,J} T^*_J \tag{7.91}$$

The heat conduction term is computed at each face $k$ and assembled to the left (IL) and right (IR) control volumes.

$$F_{k,J} = -\frac{\kappa_k}{C_{p,k}} \left( \left.\frac{\partial N_J}{\partial x}\right|_k A_x + \left.\frac{\partial N_J}{\partial y}\right|_k A_y \right) \tag{7.92}$$

$$A^d_{IL,J} \mathrel{+}= F_{k,J} \tag{7.93}$$

$$A^d_{IR,J} \mathrel{-}= F_{k,J} \tag{7.94}$$

$$q_k = -\frac{\kappa_k}{C_{p,k}} \left(t^*_x A_x + t^*_y A_y\right) \tag{7.95}$$

$$b^d_{IL} \mathrel{-}= q_k \tag{7.96}$$

$$b^d_{IR} \mathrel{+}= q_k \tag{7.97}$$

A correction for variable specific heat is applied as a volume term. The correction is computed at the centroid of the sub-volume, $k$, for control volume $I$.

$$b^d_I \mathrel{+}= \frac{\kappa}{C^2_p} \left(t_x C_{p,x} + t_y C_{p,y}\right) \Delta V_I \tag{7.98}$$

## C.7. Species, 2D Laminar Transport

There is a species equations for each species. The mass fraction is $Y_s$, where $s$ is the species number. The time term is lumped. The time-term contribution is evaluated for each sub-volume.

$$A^t_{I,I} \mathrel{+}= \rho^*_I \frac{\Delta V_I}{\Delta t} \tag{7.99}$$

$$b^t_I \mathrel{-}= \left(\rho^*_I Y^*_{s,I} - \rho^n_I Y^n_{s,I}\right) \frac{\Delta V_I}{\Delta t} \tag{7.100}$$

The convection term is computed at each face $k$ and assembled to the left (IL) and right (IR) control volumes.

$$A^c_{IL,J} \mathrel{+}= C^{n+1}_{k,J} \tag{7.101}$$
$$A^c_{IR,J} \mathrel{-}= C^{n+1}_{k,J} \tag{7.102}$$

$$b^c_{IL} \mathrel{-}= \sum_J C^{n+1}_{k,J} Y^*_{s,J} \tag{7.103}$$
$$b^c_{IR} \mathrel{+}= \sum_J C^{n+1}_{k,J} Y^*_{s,J} \tag{7.104}$$

The mass diffusion term is computed at each face $k$ and assembled to the left (IL) and right (IR) control volumes.

$$F_{k,J} = -\rho_k D_{s,k} \left( \left.\frac{\partial N_J}{\partial x}\right|_k A_x + \left.\frac{\partial N_J}{\partial y}\right|_k A_y \right) \tag{7.105}$$
$$A^d_{IL,J} \mathrel{+}= F_{k,J} \tag{7.106}$$
$$A^d_{IR,J} \mathrel{-}= F_{k,J} \tag{7.107}$$

$$f_k = -\rho_k D_{s,k} \left( y s^*_x A_x + y s^*_y A_y \right) \tag{7.108}$$
$$b^d_{IL} \mathrel{-}= f_k \tag{7.109}$$
$$b^d_{IR} \mathrel{+}= f_k \tag{7.110}$$

# 8. REVIEW OF CONTROL VOLUME FINITE ELEMENT METHODS

The earliest reference to control-volume finite-element methods is the 1980 work by Baliga and Patankar [174] for the convection-diffusion equation, a refinement of Baliga's 1978 dissertation [175]. Baliga and Patankar [176] first apply their approach to the Navier-Stokes equations of fluid mechanics in 1983. At the same time, Schneider and Zedan [177] develop a control-volume finite-element for heat conduction. Schneider and Raw [178] then develop a control-volume finite-element method for fluid flow in 1986. The work of Baliga/Patankar and Raw/Schneider are the foundations for two of the main control-volume finite-element methods that are used today for fluid mechanics. A third control-volume finite-element method is adapted from Galerkin Least Squares (GLS) finite-element methods by Swaminathan and Voller [179] in 1992, but there is no evidence of widespread use.

There are three difficult issues that must be addressed in all numerical methods for the Navier-Stokes equations: 1) stability at high Reynolds number and Peclet numbers, where pure centered differencing for the convection terms, or the analogs in FEM and FVM, can lead to numerical oscillations, 2) coupling of the pressure and velocity field, where "checker-boarding" can occur when the variables are co-located and use similar interpolations, and 3) updating of the pressure field. There are three main schools of thought in the CVFEM community for addressing the three issues above. With the Baliga/Patankar approach, upwinding is achieved with exponential shape functions on linear triangular and tetrahedral elements. Originally, pressure-velocity coupling was attained using mixed-order elements. Later, an equal-order scheme was developed that involved pressure terms in the interpolation functions. Convecting and convected velocities were maintained for pressure-velocity coupling. The pressure is solved using a projection method similar to the SIMPLER [180] algorithm. The method is practically limited to triangles and tetrahedra because of the form of the interpolation functions. With the Raw/Schneider approach, upwinding is achieved using the skewed upwinding or positive influence coefficient approaches. The pressure and velocity are solved fully coupled using an approximation of the transport equations as an interpolation function. Two velocity fields are maintained, a convecting and a convected field. The method is applicable to all element forms and has been successfully implemented in a commercial computational fluid dynamics code, TASCflow [154]. With the Swaminathan/Voller approach, the methods of streamline upwinding and pressure stabilization are adapted from finite-element methods. There is only a small amount of literature on this particular CVFEM.

The following historical synopsis of CVFEM's addresses research for solving the pressure-based incompressible Navier-Stokes equations and the more elementary convection-diffusion equations.

**1966** Winslow [181] presents a control volume formulation for a Poisson equation based on linear triangular elements. This work is important because it is one of the first applications of the finite volume method on unstructured meshes.

**1980** Ramadhyani and Patankar [182] compare the accuracy of the Galerkin finite element method with a control volume method for the Laplacian operator. They use bilinear shape functions and rectangular elements, where the control volume method uses the bilinear shape functions as interpolation functions. The numerical errors of the control volume method are half those of the finite element method.

Baliga and Patankar [174] introduce a flow-oriented upwind interpolation for convection-diffusion problems on triangular elements, a refinement of 1978 dissertation work [175]. The upwinding is introduced through an interpolation function based on a locally analytic solution to the velocity-aligned transport equation, resulting in exponential shape functions. They solve both radial heat conduction in a rotating hollow cylinder for Peclet numbers up to 100, and the transport of a step scalar field, all with specified velocity fields. The directional upwinding provides better solutions than uniform first-order upwinding.

**1983** Baliga and Patankar [176] develop a mixed-interpolation scheme for solving the Navier-Stokes equations with heat transfer on triangular elements. The mixed interpolation keeps the pressure from decoupling from velocity. The pressure is solved by applying the continuity equation over macro-triangles. The interpolation function for the convecting velocity contains the pressure gradient. Each macro-triangle is subdivided into four sub-triangles for the momentum and energy equations. The flow-oriented upwind scheme is used to interpolate velocity and temperature for their respective transport equations. The velocity is assumed to vary linearly over the element for computing mass flow rates. The equations are solved in a segregated manner using an approach similar to the SIMPLER method [180]. This work is the first application of the CVFEM for the Navier-Stokes equations.

Baliga, Pham, and Patankar [183] apply the mixed-interpolation scheme [176] to fluid flow and heat transfer. They solve flow between rotating cylinders for Reynolds numbers up to 1000, fully developed flow in a square duct with a laterally imposed velocity for Reynolds numbers up to 100, natural convection in rectangular enclosures for Rayleigh numbers up to $10^5$, and natural convection in a trapezoidal enclosure for Rayleigh numbers up to $10^6$.

**1985** Prakash and Patankar [184] solve the Navier-Stokes equations with an equal-order interpolation for velocity and pressure on triangular elements. The mass flow velocity, used for continuity, is different from that derived from momentum, thus avoiding staggering or mixed-interpolation. The flow-oriented upwind scheme is used to interpolate the convected velocity for the momentum equation while the pressure gradient is treated as an element-constant source term. The coefficient matrices for momentum are used to define the velocities for the continuity equation which include the now-unknown pressure gradient across control volume faces. They use a pressure correction approach similar to the SIMPLER algorithm to update velocity and accelerate convergence. The continuity and momentum equations are segregated in the solution process. They solve flow between rotating cylinders for Reynolds numbers up to 1000, and natural convection in a closed cavity with a Boussinesq-type buoyancy term for Grashof numbers up to $10^5$. The solutions are more accurate than with the mixed interpolation scheme of Baliga [176]. They note problems with negative coefficients during the first iterations of a solution.

Ramadhyani and Patankar [185] extend the flow-oriented upwind interpolation scheme from linear triangles to bilinear quadrilateral elements for convection-diffusion problems. Three-point quadratures (Simpsons Rule) are used to evaluate flux integrals, as in all the previously mentioned work. They argue that one-point quadratures are less accurate because of the nonlinear nature of the interpolation functions, but only at intermediate values of cell-Reynolds number. They present solutions for five different test cases, including the convection of scalar profiles and diffusion in rotating systems. After this article, there are no further publications for quadrilateral or hexahedral elements using methods developed by Baliga, Patankar, and Prakash.

**1986** Schneider and Raw [178] develop a positive influence-coefficient extension to skewed upwind interpolation [186] for convection terms, based on 1985 dissertation work [187]. They apply the scheme to the convection-diffusion equation on quadrilateral elements. Diffusion terms are calculated by integrating the gradients of the isoparametric, bilinear interpolation functions. They solve several convected-scalar cases and claim smooth solutions where the methods of Baliga and Patankar exhibit oscillations. The skewed upwind method has less dependence on the element orientation than flow-oriented streamline upwinding.

LeDain-Muir and Baliga [188] extend the flow-oriented upwind interpolation scheme to linear tetrahedral elements in three dimensions for the convection-diffusion problem. Each tetrahedron contains six control volume faces. A single unit normal is calculated for each control volume face. For integration, each face is subdivided into two triangles. A three-point quadrature is used on each triangular subface where the sample points are taken along the midpoints of the triangle edges. They solve radial heat conduction in a rotating hollow sphere, scalar transport of a step profile, and transport with radial convection between concentric spheres.

Prakash [189] modifies the flow-oriented upwind interpolation to include source terms from the transport equations on triangular elements, with applications to the incompressible Navier-Stokes equations. The pressure gradient in the momentum equations is treated as a source term in the interpolation function for velocity, directly coupling the pressure to the velocity. The source term has a streamwise-linear influence on the interpolation function. The mass flux is calculated using the new interpolation function instead of assuming a linear variation. The pressure is then calculated through the continuity equation by directly applying the new velocity interpolation function, replacing the SIMPLER scheme but keeping the segregated approach. A pressure correction step is included to make sure the velocity interpolation function satisfies continuity. He solves flow between rotating cylinders up to a Reynolds number of 1000, the lid-driven cavity for Reynolds numbers up to 400, and natural convection in a square cavity for Grashof numbers up to $10^5$. The solutions are more accurate than with the original collocated scheme of Praskash and Patankar [184].

**1987** Schneider and Raw [132, 133] extend the positive-coefficient, skewed upwind interpolation [178] to the incompressible Navier-Stokes equations on quadrilateral elements. They use a element-local discretization of the transport equations to derive interpolation functions at control volume faces that couple the velocity and pressure. The convection terms are constructed with the positive-coefficient, skewed upwinding. The skewed upwinding couples all the control volume face values together within an element, so an internal matrix inversion must be applied to calculate individual face values. The momentum and continuity equations are solved all at once as a coupled system. They solve convection of a scalar field with a step profile, the lid-driven

cavity for Reynolds numbers up to 1000, the inviscid forward-facing step to test the conservation of total pressure, and flow between rotating cylinders. Grid convergence studies suggest spatial accuracy near second order. They call their method Finite Element Difference Scheme (FIELDS).

Schneider [149] extends their algorithm [132] to cylindrical, axisymmetric coordinates and presents solutions for the cylindrical driven cavity.

Prakash [190] examines a donor-cell method for replacing flow-oriented upwind interpolation on triangular elements. The donor cell method provides positive coefficients, where the flow-oriented upwinding can yield negative coefficients, leading to oscillations. The donor-cell scheme is applied to several of the previous convected scalar problems and the thermally driven cavity. The scheme exhibits excessive diffusion and is not generally recommended.

**1988** Hookey, Baliga, and Prakash [191] modify the treatment of the source term in the flow-oriented upwind interpolation for triangular elements relative to the previous source term modifications of Prakash [189]. A crossflow-quadratic multiplier is added for the source term in the the interpolation function. They apply the scheme to the convection-diffusion equation for radial heat conduction between rotating cylinders and radial heat conduction in radial flow between cylinders. The new source treatment proves better than the previous scheme of Prakash only when the flow has multidimensional features.

**1988** Hookey and Baliga [192] apply the flow-oriented upwind interpolation with the modified source treatment [191] to the incompressible Navier-Stokes equations on triangles. Instead of calculating pressure by applying the interpolation functions directly to the continuity equation as was done by Prakash [189], a method similar to SIMPLEC [193] is used. The previous approach converged poorly at higher Reynolds numbers. A pressure correction approach is still employed to force the interpolation function for velocity to satisfy continuity, but with the penalty of an enlarged stencil for the pressure-correction equation. The continuity and momentum equations are solved simultaneously. They solve a polar lid-driven cavity for Reynolds numbers up to 380 and the natural convection for Rayleigh numbers up $10^6$. Solutions are compared against results from the older methods of Prakash [189] and Baliga [176], and prove to be more accurate.

Reviews of control volume finite element methods for fluid flow and heat transfer are given in the Handbook of Numerical Heat Transfer by both Baliga [143] and Schneider [161]. They provide implementation details for many of the methods published to date.

**1992** Swaminathan and Voller [179, 194] extend of the ideas of the Streamline-Upwind Petrov-Galerkin (SUPG) method [195] to solving the convection-diffusion equation with quadrilateral elements. They solve several convected-scalar problems and compare the results to a FEM implementation of the SUPG scheme. The CVFEM analog of SUPG performs just as well, except for time accurate solutions where the phase error is larger. The SUPG method provides better solutions than the skew upwinding or flow-oriented upwinding for heat conduction between rotating cylinders, but worse for the scalar transport of a step profile.

Baliga and Saabas [196] provide a critical review of control volume finite element methods. They criticize the schemes of Hookey [192] and Raw [132] for being too expensive, computationally. They introduce the mass advection weighted scheme of Saabas where he adapts the concept of positive influence-coefficients from Schneider and Raw to the formulation of Baliga and Prakash.

They call the original flow-oriented upwind scheme of Baliga and Patankar FLO, the source-term modified scheme of Prakash FLOS, and the mass advection-weighted scheme of Saabas MAW. The FLO(S) schemes result in mixed-sign off-diagonal coefficients if the triangular elements are obtuse, potentially admitting oscillations. Additionally, many of the schemes developed to date, for CVFEM, over-specify the pressure boundary conditions, leading to poor convergence.

Naterer and Schneider [197] extend the approach of Schneider and Raw [132] to compressible flow. An explicit predictor-corrector time integration is used for transient solutions. The influence-coefficient matrices are used to interpolate density, velocity, and internal energy at control volume faces at an intermediate time level. These values are then used to correct the state variables using a forward Euler integration. They solve a transient shock tube problem for an initial pressure ratio of 10, flow through a converging-diverging nozzle with an area ratio of 2, and Mach 3 supersonic flow over a forward-facing step.

**1993** Swaminathan, Voller, and Patankar [198] extend the streamline-upwind Petrov-Galerkin method and the pressure-stabilized Petrov-Galerkin [195] method to a conservative form for the control volume finite element method. The streamline-upwind control-volume, pressure-stabilized control-volume (SUCV/PSCV) method is applied to the incompressible Navier-Stokes equations with quadrilateral elements. They evaluate the integrals using mid-point quadrature and solve the segregated equations using a SIMPLER approach. They solve the lid-driven cavity at a Reynolds number of 400, natural convection in a square enclosure for a Rayleigh number of $10^5$, and natural convection in a cylindrical annulus at a Rayleigh number of $10^4$.

**1994** Saabas and Baliga [199, 200] adapt the positive influence-coefficient scheme of Schneider and Raw [132] to triangular and tetrahedral elements and call the method mass advection weighting (MAW). They introduce a new control volume construction for tetrahedral elements. Their tetrahedral element contains one four-point planar face and two three-point planar faces, whereas the control volume construction of LeDain-Muir [188] contained six four-point surfaces. The reduced number of control volume faces makes the MAW scheme less expensive to apply, but the element shape functions become dependent on the shape of each element. They solve for pressure using the original approach of Prakash [184] with a SIMPLER method. The solution technique is segregated. For solving practical problems, they recommend using the FLO scheme for the convection terms and switching to the MAW scheme only if there are problems with negative coefficients. They advise against using the FLOS schemes of Prakash [189] and Hookey [192] because they typically do not provide enough improvement in accuracy to justify their slower convergence properties. Additionally, they claim that carrying pressure gradient terms in the velocity interpolation function requires the boundary conditions for pressure to be over-specified for inflow/outflow problems. They solve the 2D lid-driven cavity for Reynolds numbers up to 1000, 2D turbulent flow over a backward-facing step using a $k-\epsilon$ turbulence model for a Reynolds number of about $10^6$, 3D natural convection in a cavity for Rayleigh numbers up to $10^6$, and a turbulent jet injection into crossflow for jet Reynolds numbers up to 53600. The MAW scheme is required for the jet problem because of negative coefficient problems with the FLO scheme.

Masson, Saabas, and Baliga [201] extend the MAW scheme of Saabas [199] to axisymmetric flows with triangular elements. They solve developing pipe flow for a Reynolds number of 40, pipe flow with a step constriction up to a Reynolds number of 1000, natural convection in a

cylindrical enclosure for a Grashof number of 2 and Prandtl number of 2500, and flow through and arterial section for Reynolds numbers up to 350.

Masson and Baliga [202] apply the MAW scheme of Saabas [199] to dilute-particle flows with triangular elements. They solve equations for the gas phase and the dispersed phase. They solve for flow through a constricted channel for a Reynolds number of 100 and Stokes numbers between $10^{-3}$ and $10^{-1}$, and for flow in a split inertial separator for a Reynolds number of 200.

Karimian and Schneider [203] improve the velocity-pressure coupling of the original Schneider-Raw scheme [132]. The original scheme, referred to as FIELDS, has poor performance for inviscid flow. They improve the coupling by adding a discrete continuity relation to the interpolation functions for the convecting velocity. The additional terms help smooth oscillations that occur for a mass sink test problem. They verify the new interpolation function on the lid-driven cavity for Reynolds numbers up to 3200, and the backward-facing step for Reynolds numbers up to 230.

Karimian and Schneider [204] extend the method of Schneider and Raw [132] to both compressible and incompressible flow for the quasi-one-dimensional Euler equations. They solve for flow through a converging-diverging nozzle with an area ratio of 2.035 with and without a shock.

Deng *et al.* [205] present a new flux reconstruction scheme to replace the FIELDS scheme of Schneider and Raw [133]. They note that the FIELDS scheme is similar to the original work of Rhie and Chow [131] who where some of the first researchers to solve incompressible flow on collocated grids. Deng takes features of both schemes to create a compact reconstruction that does not require matrix inversions to calculate the integration point values in terms of nodal values. Since they question the consistency of the FIELDS scheme, they call their new scheme consistent physical interpolation (CPI). They apply the scheme to two and three-dimensional Navier-Stokes calculations on structured Cartesian meshes. They solve the lid-driven cavity for Reynolds numbers up to 1000, a 3D lid-driven cavity for Reynolds numbers up to 1000, and turbulent vortex shedding over a square cylinder for a Reynolds number of 22000.

**1995** Costa *et al.* [206] apply the MAW scheme of Saabas [199] to three-dimensional turbulent flows with tetrahedral elements. They solve a turbulent jet injected into a crossflow for jet Reynolds number up to 53600, and flow through a T-junction in ducts at Reynolds numbers near 90000.

Karimian and Schneider [207] apply control-volume finite-element methods to a shock-tube problem.

Karimian and Schneider [208] extend the FIELDS scheme and the convecting velocity corrections to compressible flow. They solve the lid-driven cavity for Reynolds numbers up to 3200, flow over a shallow bump in a channel with Mach numbers from 0.5 to 1.65, and flow through a ramped inlet for a Mach number of 2.5.

Padra and Larreteguy [209] develop an error estimator with mesh refinement for the convection-diffusion equation. They use the formulation of Baliga and Patankar [174] with triangles. Larreteguy [210] then extends the scheme to fluid flow with triangles.

**1996** Harms *et al.* [211] introduce a simplified interpolation function for the control volume finite element method. They develop a method for applying analytic shape functions on nonorthogonal meshes. They apply the scheme to flow between rotating cylinders for Reynolds numbers up to 1000 and the scalar transport of a step profile.

Comini et al. [212] compare CVFEM and GFEM formulations for the convection-diffusion equations.

Neises and Steinbach [213] develop a control volume finite element method based on a mixed interpolation approach to facilitate pressure-velocity coupling and artificial dissipation for convective stability. They begin with a Galerkin finite element method and then manipulate off-diagonal terms to force conservation. They solve a laminar, 3D obstructed channel flow for Reynolds number from 0.1 to 50000.

Völker, Burton, and Vanka [214] apply a multigrid solution technique to a control volume finite element method on triangular elements. Linear interpolation is used throughout the triangles with a three-point quadrature to integrate fluxes. The pressure is solved using the SIMPLE method. They solve natural convection problems in square, triangular, and semicircular cavities for Rayleigh numbers up to $10^6$.

Botta and Hempel [215] describe a finite-volume projection method for unstructured, triangular meshes with element-centered variables.

**1997** Darbandi and Schneider [216] develop a scheme for both compressible and incompressible flow using a momentum variable formulation of the Schneider/Raw scheme [132, 133]. The interpolation formula for the convecting velocities is derived from an approximation of the momentum equation with an additional velocity-weighted continuity equation term. Solutions are demonstrated for velocities up to Mach 0.9.

Baliga [217] gives an overview of the control volume finite element method as applied to fluid flow.

**1998** O'Rourke and Sahota [218] develop an edge-based scheme in 3D for the convection operator. The convection operator is constructed from a multidimensional upwind scheme. Within each element, the quadrature points are associated with edge mid-points instead of sub-face mid-points, so the amount of work is reduced over the traditional CVFEM.

Gresho and Sani [219] compare CVFEM methods to GFEM methods.

Venditti and Baliga [220] describe an error estimation strategy for incompressible flow with CVFEM.

**2001** Reyes, Rincon, and Damia [221] present a CVFEM approach for turbulent flow with wall functions.

Campos Silva and de Moura [222] present a method for 9-noded quad elements with the mass advection weighted scheme.

**2002** Zhao, Tai and Ahmed [223] implement a 2D CVFEM on triangles for micro flows. They use an upwind scheme where nodal gradient are used to reconstruct the high-order fluxes at the control volume faces.

With respect to fluid flow, the CVFEM methods have been developed primarily for triangular and tetrahedral elements [174, 176, 184, 188, 189, 191, 192, 199, 201]. Development focused on triangular and tetrahedral elements because the shape functions are linear and gradient terms become constant over the element. Constant first derivatives simplify the formulation of many of the schemes. Fewer articles have been published on the use of quadrilateral elements [185, 178, 132, 179, 198] in two dimensions and no articles have been published for CVFEM with hexahedral elements in three dimensions. In addition, there have been CVFEM formulations for the streamline-vorticity equations [224, 225, 226, 227, 37, 228], for the heat equation [177, 229, 230, 231, 232, 233, 234, 235, 236], for flow in porous media [162, 237, 238, 239, 240, 241, 242], for overland flows [243, 244], and for linear elasticity [245, 246].

# 9. TURBULENCE MODELING WITH V2-F TRANSPORT EQUATIONS

The level 1 turbulence model in FUEGO is the standard $k-\epsilon$ turbulence model with constants and wall functions established for forced convection flows. The v2-f turbulence model is a modified $k-\epsilon$ turbulence model [21] that has been implemented and evaluated recently for several flows without [247] and with [248] heat transfer in the Sandia research code CURRENT [249]. The v2-f model has been implemented in FUEGO recently as an unsupported feature. As an initial test of the FUEGO implementation, model results for a fully developed, isothermal, turbulent flow in a channel are compared with results from a 1D code [250]. Using 60 non-uniformly spaced grid points across the half-height of the channel, good agreement is obtained for a channel flow at $\mathrm{Re}_h = u_{\mathrm{cl}} h/\nu = 13,800$ where $h$ is the half-height of the channel and $u_{\mathrm{cl}}$ is the centerline velocity. This grid results in $y^+ = y u_\tau/\nu \approx 0.5$ at the center of the subcontrol volume that is adjacent to the channel wall.

The next steps will be to compare the model for heat transfer with other model results and experiments in flow regimes of forced and mixed convection for several flow geometries (channel or tube flow, stagnation flow and separated flow). While the forced flow results can be compared with other numerical and experimental work, there is much less information available for the mixed convection regime.

# APPENDIX A. INTRODUCTION

Although radiation is the dominant heat transfer mechanism in a pool fire, convection can be significant for some conditions. The convective heat transfer regime most likely to exist in a pool fire is turbulent mixed convection where both buoyancy and forced flow effects (due to external wind or air flow induced by the large density changes associated with the fire) can be important. The flow regimes and geometries encountered range from flow over a flat surface (e.g., the ground) to impinging and separated flow (e.g., objects lying on the ground either in or adjacent to the fire). Turbulent transport processes are typically modeled using the Boussinesq hypothesis to relate the turbulent transport terms (stresses or fluxes that result from averaging the dependent variables in the conservation equations) to the mean rate of strain. For example, the Reynolds stresses are often modeled as:

$$-\overline{\rho u_i' u_j'} = \mu_t \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \rho \delta_{ij} k \tag{9.1}$$

where $k$ is the kinetic energy of turbulence, $\frac{1}{2}\overline{u_i' u_i'}$, and $\mu_t$ is the turbulent viscosity, which in the widely used $k-\epsilon$ model of turbulence is given by:

$$\mu_t = C_\mu \frac{\rho k^2}{\epsilon}; \tag{9.2}$$

$\epsilon$ is the mean viscous dissipation, defined by:

$$\epsilon = \nu \overline{\left(\frac{\partial u_i'}{\partial x_j} + \frac{\partial u_j'}{\partial x_i}\right) \frac{\partial u_i'}{\partial x_j}} \tag{9.3}$$

In the standard form of the $k-\epsilon$ model (Launder and Spalding [38]), which we are using in the level 1 fire code modeling, the transport equations for $k$ and $\epsilon$ are given by:

$$\frac{\partial \rho k}{\partial t} + \frac{\partial \rho u_j k}{\partial x_j} = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] + 2\mu_t S^2 - \rho\epsilon \tag{9.4}$$

$$\frac{\partial \rho \epsilon}{\partial t} + \frac{\partial \rho u_j \epsilon}{\partial x_j} = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_\epsilon}\right)\frac{\partial \epsilon}{\partial x_j}\right] + 2C_{\epsilon_1}\mu_t S^2 \frac{\epsilon}{k} - C_{\epsilon_2}\rho\frac{\epsilon^2}{k} \tag{9.5}$$

where

$$S^2 = S_{ij}S_{ij} = \frac{1}{4}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right), \tag{9.6}$$

and $C_\mu = 0.09$, $C_{\epsilon_1} = 1.45$, $C_{\epsilon_2} = 1.92$, $\sigma_k = 1.0$, $\sigma_\epsilon = 1.3$.

The above constants have been determined through extensive numerical studies on primarily isothermal, high Reynolds number, turbulent shear flows. The boundary conditions for the transport equations in turbulent flow have traditionally involved the use of wall functions to avoid the computational cost of resolving the very steep gradients of the variables near the wall. These wall functions assume knowledge of the profiles of the variables (e.g., velocity and temperature) near the wall, and in the case of the turbulence parameters assumptions are made about the transport processes in the wall region (e.g., production and dissipation of turbulent kinetic energy are in balance).

Alternatively, if computational costs are not a concern, the wall function approach is abandoned; instead, a fine grid is used near the wall and boundary conditions are applied directly at the wall (e.g., zero values of the velocity components, specified temperature, $k = 0$ and $\epsilon = \infty$). In this case (referred to as the low Reynolds number modification to the standard $k-\epsilon$ model of turbulence, Jones and Launder [34]), however, the coefficients in the above equations are no longer constant but become dependent on the distance from the wall (modeled using damping functions). A review of the low Reynolds number turbulence models is given in Patel et al. [251]. Both wall functions and damping functions require empirical information or assumptions about the gradients of the variables near a wall; neither case is desirable, since a primary reason for solving transport equations is to predict the spatial variation of the dependent variables. Recently, Durbin [21] has presented the v2-f modification of the $k-\epsilon$ turbulence model that avoids both wall functions and damping functions by solving two additional transport equations. The model has been tested for several forced convection flows, with [252] and without [253] heat transfer; more recently it has been applied successfully to convective heat transfer problems in stagnation flow with (Parneix et al. [254]) and without (Behnia et al. [255]) separated flow regions.

# APPENDIX B. THE V2-F MODEL

Durbin [21] introduced a method for handling the wall region without using either wall functions or damping functions. In his method a fine grid is required near the wall (e.g., the first grid point is typically within one dimensionless unit of distance from the wall where the coordinate normal to the wall is nondimensionalized with the inner scale for a turbulent boundary layer, $y^+ = yu_\tau/\nu < 1$ at the first grid point, where $u_\tau$ is the friction velocity, $\sqrt{\tau_w/\rho}$). The model employs two transport equations in addition to slightly modified $k-\epsilon$ equations to account for the nonhomogeneous region near the wall. The eddy viscosity is formulated using the component of turbulent kinetic energy normal to the wall for velocity scaling (instead of using $\sqrt{k}$ as done in the standard $k-\epsilon$ model):

$$\mu_t = C_\mu \rho \overline{v^2} T. \tag{9.7}$$

The time scale, $T$, is the usual time scale, $k/\epsilon$, away from the wall region; however, near the wall, if $k/\epsilon$ becomes smaller than the Kolmogorov time scale, $\sqrt{\nu/\epsilon}$, then the latter is used for $T$. The model includes a transport equation for $\overline{v^2}$:

$$\frac{\partial \rho \overline{v^2}}{\partial t} + \frac{\partial \rho u_j \overline{v^2}}{\partial x_j} = \frac{\partial}{\partial x_j}\left[(\mu + \mu_t)\frac{\partial \overline{v^2}}{\partial x_j}\right] + \rho k f - \frac{\rho N \overline{v^2}}{T_1}. \tag{9.8}$$

An elliptic relaxation model equation is formulated to solve for the variable $f$ in the above equation. The purpose of the elliptic relaxation model is to account for nonlocal effects such as wall blocking; the equation is given by:

$$f - L^2 \frac{\partial}{\partial x_j}\left(\frac{\partial f}{\partial x_j}\right) = C_1 \frac{\left(2/3 - \overline{v^2}/k\right)}{T_1} + C_2 2\nu_t \frac{S^2}{k} + (N-1)\frac{\overline{v^2}/k}{T_1}. \tag{9.9}$$

The turbulent kinetic energy equation 9.4 remains the same in the v2-f model; however, the dissipation equation 9.5 is modified as follows:

$$\frac{\partial \rho \epsilon}{\partial t} + \frac{\partial \rho u_j \epsilon}{\partial x_j} = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_\epsilon}\right)\frac{\partial \epsilon}{\partial x_j}\right] + 2C'_{\epsilon_1}\mu_t \frac{S^2}{T} - C_{\epsilon_2}\rho\frac{\epsilon}{T}. \tag{9.10}$$

The time and length scales in the above equations are given by:

$$T = \min\left[T_1, \frac{\alpha}{2\sqrt{3}}\frac{k}{\overline{v^2}C_\mu\sqrt{S^2}}\right] \tag{9.11}$$

$$T_1 = \max\left[\frac{k}{\epsilon}, 6\sqrt{\frac{\nu}{\epsilon}}\right] \tag{9.12}$$

$$L = C_L \max\left[L', C_\eta\left(\frac{\nu^3}{\epsilon}\right)^{\frac{1}{4}}\right] \tag{9.13}$$

$$L' = \min\left[\frac{k^{\frac{3}{2}}}{\epsilon}, \frac{1}{\sqrt{3}}\frac{k^{\frac{3}{2}}}{\overline{v^2}C_\mu\sqrt{S^2}}\right] \tag{9.14}$$

and the constants are given by: $C'_{\epsilon_1} = C_{\epsilon_1}\left(1 + 0.045\sqrt{k/\overline{v^2}}\right)$, $C_{\epsilon_1} = 1.4$, $C_{\epsilon_2} = 1.9$, $C_\mu = 0.22$, $C_1 = 0.4$, $C_2 = 0.3$, $\alpha = 0.6$, $N = 6$, $C_L = 0.23$, $C_\eta = 70$, $\sigma_\epsilon = 1.0$.

Boundary conditions at a no-slip, solid wall are given by:

$$k = \overline{v^2} = f = 0 \tag{9.15}$$

$$\epsilon = 2\nu k(1)/y(1)^2 \tag{9.16}$$

where $k(1)$ and $y(1)$ are the turbulent kinetic energy and the normal distance from the wall at the center of the subcontrol volume that is adjacent to the wall. The $\epsilon$ condition at the wall node is determined by weighting the above expression for each subcontrol volume associated with the wall node by the subcontrol volume wall surface area, accumulating the values for all the subcontrol volumes that make up the boundary control volume associated with the wall node, and dividing by the total wall surface area for the boundary control volume.

# APPENDIX C. TEST PROBLEM

The equations and boundary conditions for the v2-f model have been implemented in FUEGO. A 1-D code for solving the equations of fully developed, isothermal, turbulent flow in a channel was obtained from Durbin [250] and is used here to verify the model implementation. The Reynolds number chosen for the verification test is $\mathrm{Re}_h = u_{\mathrm{cl}}h/\nu = 13,800$ where $h$ is the half-height of the channel and $u_{\mathrm{cl}}$ is the centerline velocity. This condition was chosen for verification and validation purposes because it has been studied thoroughly both experimentally [256] and numerically [257]. From [256], $u_\tau/u_{\mathrm{cl}} = 0.0464$ at $\mathrm{Re}_h = 13,800$, and $h = 3.175$cm. Properties of nitrogen at 300K, 1atm are used in the FUEGO calculations: $\rho = 1.138\mathrm{x}10^{-3}\mathrm{g/cm}^3$ and $\mu = 1.813\mathrm{x}10^{-4}\mathrm{g/cm-s}$. This gives a Reynolds number based on the friction velocity $u_\tau$ and the channel height $2h$ of 1280.6 to use in Durbin's 1D channel code. The profiles of velocity, turbulent kinetic energy $k$, turbulent dissipation rate $\epsilon$, and $v2$ computed from the 1D simulation are used as inlet profiles for the FUEGO (version 0.5.2) calculation. The 3D FUEGO simulation included one element in the lateral ($z$) direction; symmetry conditions were imposed on the minimum and maximum $z$ planes; symmetry was also imposed at the channel centerline ($y = 0$). Outflow boundary conditions with $p = 0$ were imposed at the outflow boundary ($x = 8$cm). Three different meshes were used to discretize the half-width of the channel: a fine mesh with 26 nodes where the smallest to largest mesh spacing ratio was 0.375; a finer mesh with 40

nodes where the smallest to largest mesh spacing ratio was 0.1; and a finest mesh with 60 nodes where the smallest to largest mesh spacing ratio was 0.075. For these meshes the values of $y^+ (\equiv (h-y)u_\tau/\nu)$ at the center of the subcontrol volume that is adjacent to the channel wall are $\approx$ 3, 1, and 0.5, respectively. The finest mesh FUEGO velocity profile at the outlet of the channel ($x = 8$cm) is compared with the 1D profile and with the experimental data of Hussain and Reynolds in the near wall region in Figure C-1. The velocity in Figure C-1 is normalized with the centerline velocity; the distance from the wall, $h - y$, is normalized with the half-height of the channel $h$.

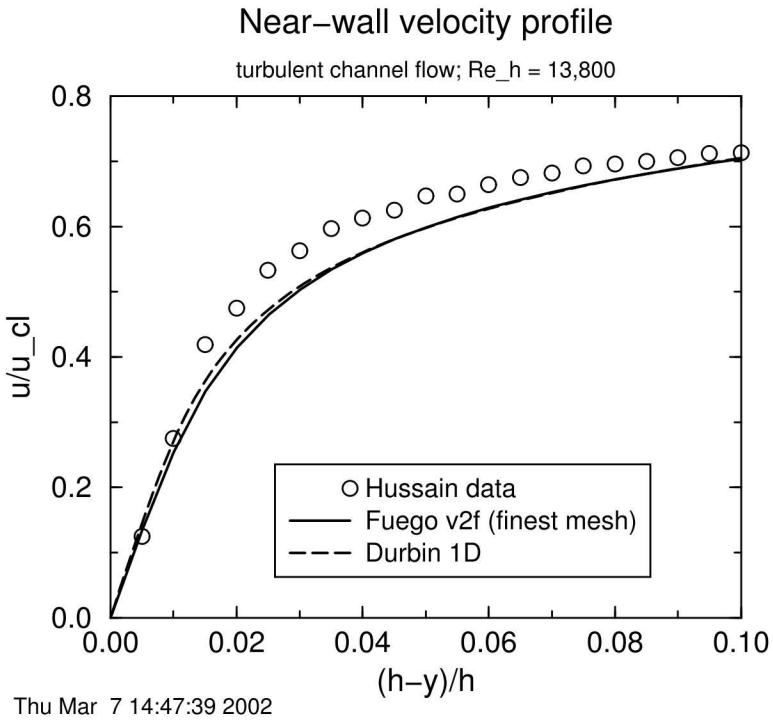


**Figure C-1.. Near wall profiles of velocity, Re=13,800 (fuego-0.5.2).**

Velocity profiles across the half-width of the channel are shown in dimensional form in Figure C-2. Included in the Figure is the velocity profile from a FUEGO $k-\epsilon$ calculation which solved the $k$ transport equation for the control volume adjacent to the wall and used the code option *use equilibrium production model*. The $k-\epsilon$ calculation used 10 equally spaced elements across the channel half-height and is described in detail in the verification chapter of this document dealing with wall functions in turbulent flow.

The finest mesh FUEGO turbulent kinetic energy profile at the outlet of the channel ($x = 8$cm) is compared with the 1D profile and with a $k$ profile formed from a combination of the experimental data of Hussain and Reynolds for $\overline{u'^2}$ and the LES calculation of Moin and Kim for $\overline{v'^2}$ and $\overline{w'^2}$ in the near wall region in Figure C-3. The $k$ values are normalized with ${u_\tau}^2$.

The variation of turbulent kinetic energy across the half-height of the channel is compared with Durbin's 1D profile in Figure C-4. A profile of $k$ computed using the $k-\epsilon$ model is included for reference.

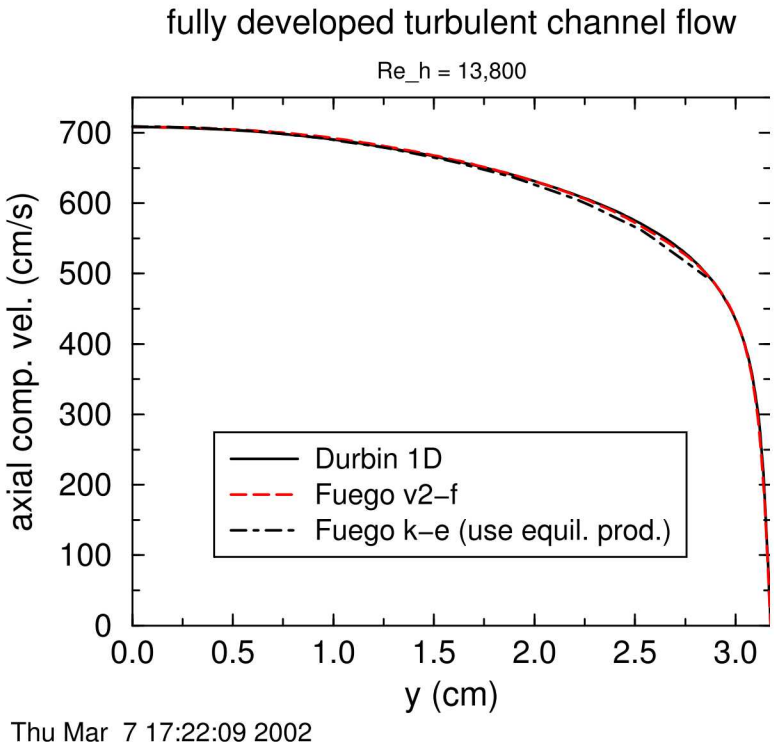


**Figure C-2.. Profiles of velocity, Re=13,800 (fuego-0.5.2).**

The variation of turbulent dissipation across the half-height of the channel is compared with Durbin's 1D profile shown in Figure C-5. A profile of $\epsilon$ computed using the $k - \epsilon$ model is included for reference.

Profiles of $\epsilon$ normalized with $h/{u_\tau}^3$ in the near wall region are shown in Figure C-6 as a function of $y^+$ for the three meshes and compared with Durbin's 1D profile. Good agreement is obtained for the finest mesh.

Profiles of $v2$ across the half-height of the channel at the channel exit ($x = 8$ cm) for the three meshes are compared in Figure C-7 with the 1D profile of Durbin. Good agreement with the 1D profile is obtained for the finest mesh FUEGO calculation; note that for the finest mesh $y^+$ at the center of the subcontrol volume that is adjacent to the channel wall is $\approx 0.5$.

# APPENDIX D. NUMERICAL IMPLEMENTATION ISSUES AND DETAILS

In the course of verification and validation of the v2-f model in Fuego, several convergence related issues have emerged. Poor and/or lack of convergence that did not respond to modifications in the CFL criterion or the underrelaxation factors or the projection scheme became severe issues for the turbulent mixed convection flat plate validation study. As a result, some rather drastic steps were taken to obtain convergence.

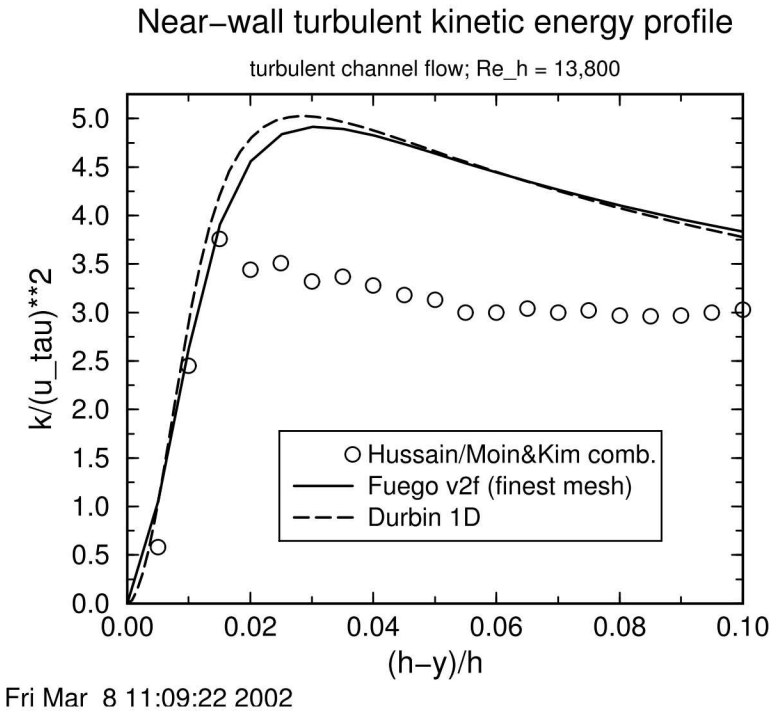


**Figure C-3.. Near wall profiles of turbulent kinetic energy, Re=13,800 (fuego-0.5.2).**

First, the order of solution and update of the turbulence variables was modified. The resulting order of solution and update became the following: (1) solve the f equation and update f; (2) solve the k equation, then the epsilon equation, and finally the v2 equation; (3) update k, epsilon, and v2.

Second, as noted in the model formulation above, the time scale used in the f and v2 equations differs from the time scale used in the epsilon equation and the turbulent viscosity formula; $T_1$ (no realizability constraint) is used in the f and v2 equations, whereas T (including the realizability constraint) is used in the epsilon equation and turbulent viscosity. Also, Durbin's original model has epsilon/k in the sink term in the v2 equation; this has been replaced by $1/T_1$ as noted in the above model formulation. This modification was determined by Svengingsson to have a large stabilizing effect on the v2-f model in solutions of gas turbine flows. Usage of the realizability constraint in the time scale in some of the turbulence equations and not others has appeared on and off in publications of the v2-f model over the years.

Third, apply limiters to the time and length scales and to the source terms in the turbulence equations and the coefficient of the production term in the epsilon equation. These limiters have been hardwired in the code (must be changed in source code and then the code is recompiled); they serve to allow the v2-f model to adjust to the initial condition. Once the adjustment is made, then the limiters can be removed and the simulation remains stable. Without the use of limiters, the model can be unstable.

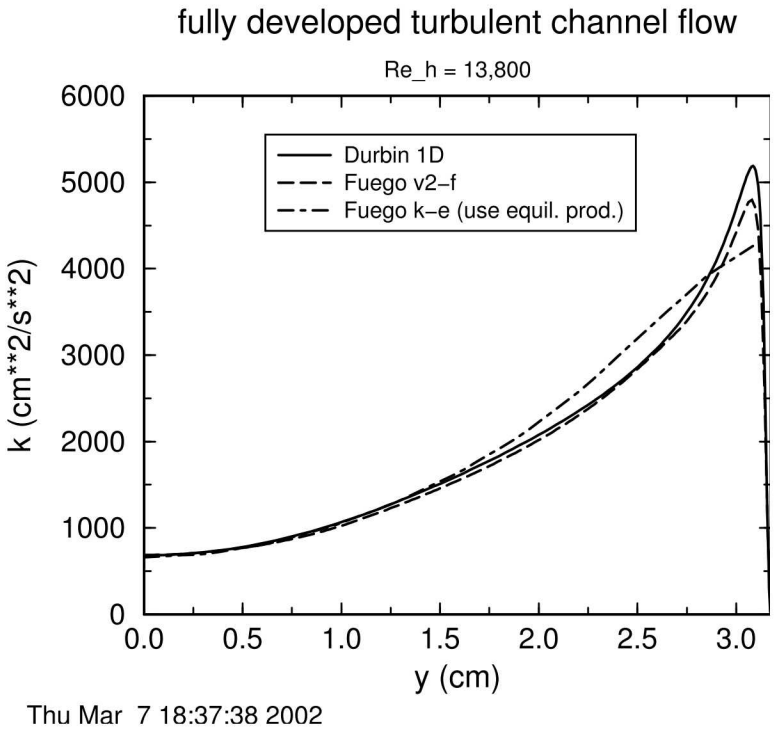


**Figure C-4.. Profiles of turbulent kinetic energy, Re=13,800 (fuego-0.5.2).**

# APPENDIX E. PLAN

The goal is to have a model for turbulent mixed convection heat transfer in FUEGO that provides a more accurate prediction of the convective heat transfer to surfaces in or near fires than the standard $k - \epsilon$ model provides without increasing the cost dramatically. The v2-f model seems to be a good starting point for such a model. The next steps will involve applying the model to solve for the convection heat transfer in flow regimes of forced and mixed convection for channel flow, boundary layer flow, and separated flow. Comparisons will be made with published numerical solutions and experimental data. While the forced flow results can be compared with other numerical and experimental work, there is much less information available for the turbulent mixed convection regime.

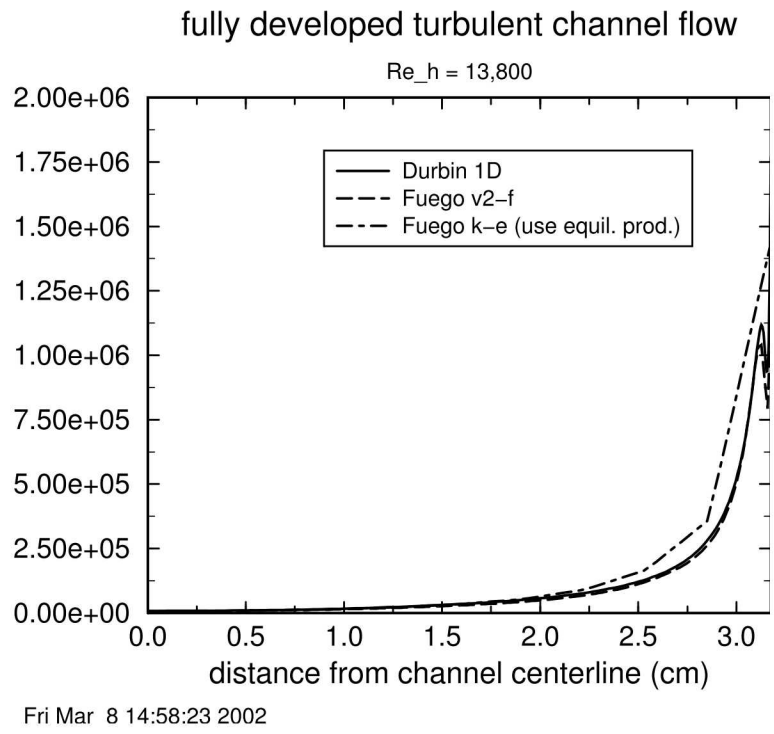


**Figure C-5.. Profiles of turbulent dissipation, Re=13,800 (fuego-0.5.2).**

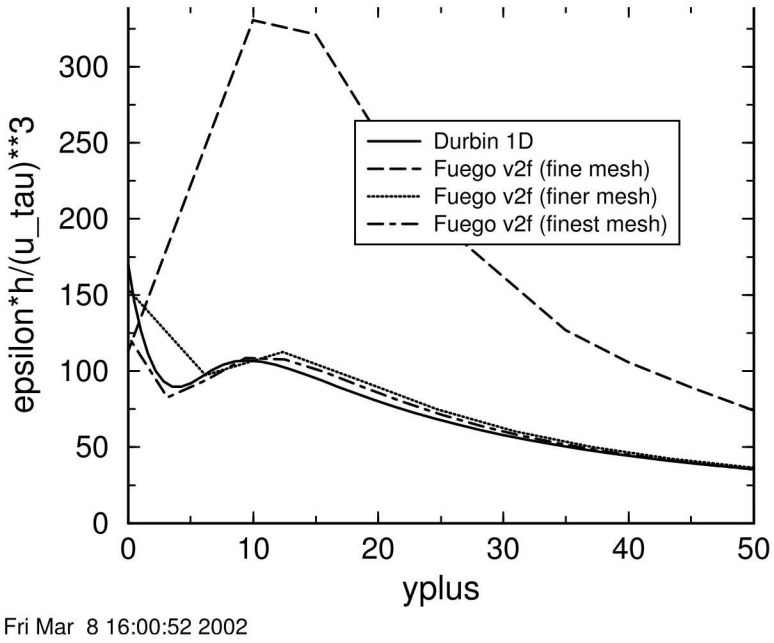


**Figure C-6.. Near wall profiles of turbulent dissipation, Re=13,800 (fuego-0.5.2).**

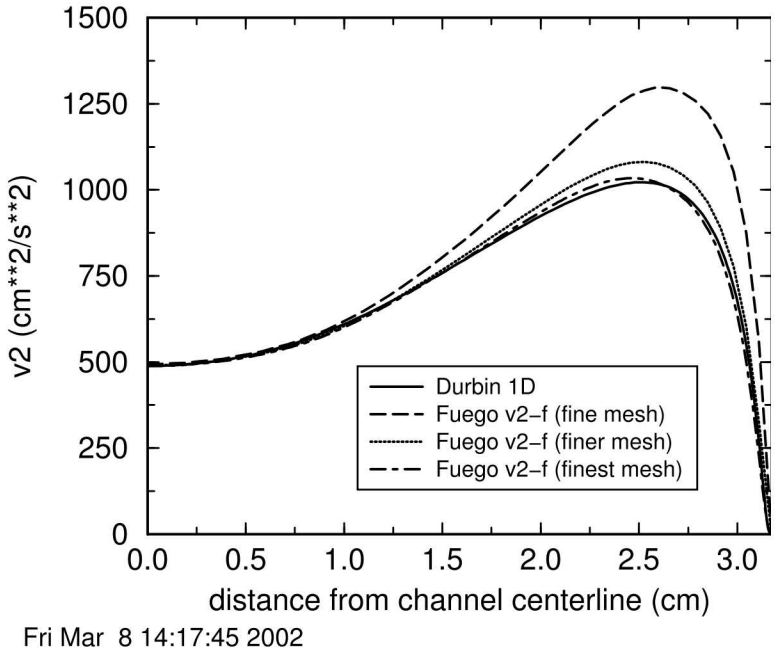


**Figure C-7.. Profiles of $v2$ using three meshes, Re=13,800 (fuego-0.5.2).**

# 10. BUOYANT VORTICITY GENERATION MODEL

The turbulence models most commonly employed in commercial fire CFD tools (or production codes, i.e., codes that are not research codes) are based upon the $k-\epsilon$ model of turbulence [38, 34]. Such models have well-known strengths and weaknesses, and are used primarily because they are robust, i.e., they yield reasonable results for many different scenarios of interest. The use of the $k-\epsilon$ turbulence model for fire simulation is somewhat surprising, as the model was derived for flows with primarily shear-generated turbulence, whereas fires are flows with primarily buoyancy-generated turbulence. Most CFD fire simulation tools employ a standard $k-\epsilon$ turbulence model (some with low Reynolds number modifications). When fires in enclosures are simulated, a correction term is often included to account for thermal stratification effects that tend to dampen turbulent kinetic energy in the hot gas layer near the ceiling of the enclosure ($cf.$ [258]). This correction was first suggested by Rodi [32], and is referred to herein as 'Rodi's term.' In some of the results that follow, reference is made to the 'standard $k-\epsilon$ model.' This is assumed to include Rodi's term for buoyant turbulence, as his model is 'standard' for most of the literature regarding conventional CFD fire simulations.

## .1. The Present Work

The goal of the present work is to develop a model of buoyancy-induced turbulence for pool fires using a buoyant vorticity-based generation mechanism. The models developed previously ($cf.$ [31], while promising, had several shortcomings that needed to be overcome. First, the authors believed that the model needed to be put on a more solid theoretical foundation. Chomiak and Nisbet [259] had relied upon similarities to flows involving bubble dynamics in developing key parts of their formulation. Second, the previous model relied upon an upper limit on the buoyant production term. This limit was felt to be a severe hindrance in applying the model over a broad range of fire environments, and therefore a significant shortcoming. Third, previous models yielded a build up of eddy viscosity not only in the plume, but also in regions far removed from it. In view of these shortcomings, it was deemed necessary to develop another model of buoyancy-generated turbulence for pool fires.

## .2. Model Development

The standard equations relevant to momentum transport and turbulence are presented first to establish a background. This is followed by a derivation of the buoyancy-generated turbulence modifications for the modeling of pool fires.

The momentum equation (written in integral form) can be written as:

$$\int \frac{\partial \overline{\rho}\tilde{u}_i}{\partial t}\mathrm{d}V + \int \overline{\rho}\tilde{u}_i\tilde{u}_j n_j \mathrm{d}S + \int \overline{p} n_i \mathrm{d}S = \int \overline{\tau}_{ij} n_j \mathrm{d}S - \int \overline{\rho u_i'' u_j''} n_j \mathrm{d}S + \int \left(\overline{\rho} - \rho_\circ\right) g_i \mathrm{d}V \quad (10.1)$$

where variables with an overbar are Reynolds averaged, variables with a tilde are Favre-averaged (density weighted), and the double prime (") indicates a fluctuation. The second to the last term on the right hand side (RHS) involving the velocity fluctuations is commonly referred to as the Reynolds stress term. It is this term that requires modeling in order to close the set of equations (which also includes conservation of mass, species, and energy (or enthalpy)).

Invoking the Boussinesq eddy viscosity assumption, the Reynolds stress term can be written as:

$$\begin{aligned} -\overline{\rho u_i'' u_j''} &= \mu_t \left( \frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} \right) - \frac{2}{3}\left( \overline{\rho}\tilde{k} + \mu_t \frac{\partial \tilde{u}_k}{\partial x_k} \right) \delta_{ij} \\ &= \overline{\tau^t}_{ij} - \frac{2}{3}\overline{\rho}\tilde{k}\delta_{ij}, \end{aligned} \quad (10.2)$$

where $mu_t$ is the turbulent eddy viscosity, snd is given by the Prandtl-Kolmogorov relationship,

$$\mu_t = \mathrm{C}_\mu \overline{\rho} \frac{k^2}{\epsilon}. \quad (10.3)$$

:

When expressions for k and $/epsilon$ are put forth, then the Reynolds stress term can be evaluated, a closed set of equations is obtained, and a solution to the suite of momentum, mass, species, and enthalpy equations can (in theory) be obtained.

The standard form of the $k-\epsilon$ equations for buoyant flow is modified as follows. The equation for turbulent kinetic energy (see Equation 2.102 for the original form) can be written as

$$\int \frac{\partial \overline{\rho}\tilde{k}}{\partial t}\mathrm{d}V + \int \overline{\rho}\tilde{k}\tilde{u}_j n_j \mathrm{d}S = \int \frac{\mu_t}{\sigma_k}\frac{\partial \tilde{k}}{\partial x_j} n_j \mathrm{d}S + \int \left(P_k - \overline{\rho}\tilde{\epsilon} + G_B\right) \mathrm{d}V. \quad (10.4)$$

The term $G_B$ represents a source term due to buoyancy, and needs to be modeled (the term $P_k$ is the standard source term due to shear). The equation for the dissipation of turbulent kinetic energy (see Equation 2.103 for the original form) can be written as

$$\int \frac{\partial \overline{\rho}\tilde{\epsilon}}{\partial t}\mathrm{d}V + \int \overline{\rho}\tilde{\epsilon}\tilde{u}_j n_j \mathrm{d}S = \int \frac{\mu_t}{\sigma_\epsilon}\frac{\partial \tilde{\epsilon}}{\partial x_j} n_j \mathrm{d}S + \int \frac{\tilde{\epsilon}}{\tilde{k}} \left(C_{\epsilon 1} P_k - C_{\epsilon 2}\overline{\rho}\tilde{\epsilon} + C_{\epsilon 3} G_B\right) \mathrm{d}V. \quad (10.5)$$

The term $G_B$ appears in this equation also.

## .3. A new model for buoyancy generated turbulence

In view of the limitations and weaknesses of previous models, the development of a new model was undertaken. From Equation 2.102 we note that:

$$\int \frac{\partial \overline{\rho}\tilde{k}}{\partial t} \mathrm{d}V = \int G_B \mathrm{d}V. \tag{10.6}$$

Since we are using a $k - \epsilon$ based turbulence model, shear-generated turbulence influences the momentum equations through a turbulent eddy viscosity (or diffusivity). Therefore, we want the influence of buoyancy-generated turbulence to manifest itself also as an eddy viscosity. From Equation 2.106, we can see that, for the shear-generated turbulence case:

$$\mu_t = \mathrm{C}_\mu \overline{\rho} \frac{k^2}{\epsilon} = \mathrm{C}_\mu \overline{\rho} \frac{k}{\epsilon} k \, k\tau. \tag{10.7}$$

Since we want the same effect (i.e., the same eddy viscosity) when the turbulence is buoyancy-generated, we can write

$$\mu_t = \mu_B \tag{10.8}$$

(8) From Equation 10.7 and Equation 10.8, we can derive a relationship between the buoyancy-generated turbulence quantities and the shear-generated turbulence quantities that will ensure proper representation of the eddy viscosity for the buoyancy-generated turbulence case,

$$k\tau_s = k_B \tau_B \tag{10.9}$$

where the subscript $s$ indicates a shear-related quantity, and the subscript $B$ represents a buoyancy-related quantity. Note that we could also write as $k = k_s$, but have chosen to not include the subscript $s$ on k (and below on $epsilon$) in order to be consistent with the previous equations and naming convention. Rearranging equation Equation 10.9,

$$k_B = k \frac{\tau_s}{\tau_B} \tag{10.10}$$

The appropriate time scale for shear-generated turbulence is given by:

$$\frac{\tau_s = k}{\epsilon} \tag{10.11}$$

With proper representation of both $k_B$ and $\tau_B$ for buoyancy-generated turbulence in Equation 10.10, then the proper impact of buoyancy-generated turbulence on the momentum equations (via Equation 2.106) will be obtained.

An appropriate time scale for buoyancy-generated turbulence can be deduced by noting that the turbulence in pool fires is generated primarily as a result of buoyant vorticity generation (for a more detailed discussion, see [47] ). From the vorticity equation, we note that

$$\frac{\partial \omega}{\partial t} \frac{1}{\tau^2} \frac{1}{\tau^2} GradRxGradP \tag{10.12}$$

where the double vertical bars indicate that a magnitude must be taken (since the resultant of the cross product is itself a vector). This time scale is based on the mechanism for buoyant vorticity generation (BVG), and is inversely proportional to the square root of the cross product of the local density gradient and the pressure gradient. Making use of equations and 10.12 and 10.11 we can re-write Equation 10.10 as:

Or, making use of equation (3):

(14)

Examining equation (6), and noting that GB can also be related to the rate of change with time of the buoyancy-generated turbulent kinetic energy, , we can re-write (6) as:

(15)

Adding in a constant of proportionality, CBVG , the source term to the k-equation due to buoyancy-generated turbulence becomes:

(16)

Note that CBVG is not the only constant that must be determined for the model. The equation for the dissipation of turbulent kinetic energy, equation (5), also contains a constant (C??) that must be determined. The determination of these two constants is done by comparing the results of the model to experimental data. Calibration of these constants is presently underway.

## .4. Implementation Issues

If the present model is implemented into a code which uses an essentially incompressible scheme, it has been observed that there can be problems with the model during the first several time steps. The large pressure pulse that occurs upon startup results in very high values of the pressure gradient. Although this pressure pulse generally only lasts for the first several time steps, it can wreak havoc with the solution by generating significant values of $G_b$ in regions far removed from the plume itself.

## .5. Summary

This new model is hereafter referred to as the BVG model (for Buoyant Vorticity Generation model), and has also been implemented into the Sandia VULCAN fire simulation. The model appears to work for non-reacting as well as reacting buoyant flows. Work is underway to calibrate the constants against experimental data.

# 11. PROPOSED RESTART FIX FOR DT SCALING ALGORITHM

A solution to the fuego restart problem, within the context of the "dt" scaling algorithm, is provided herein. The development of the algorithm, which centers about the construction of the appropriate mass conserving integration point mass flow rates, provides a method for calculation of the appropriate mass conserving integration point flow rates that is strictly a function of the previous projected pressure and velocity field. Moreover, the interpolation provides for a more satisfying method for cases of variable density. Therefore, this method requires no framework necessity of integration point data structure saving. The equivalence of this new interpolation formula with the slightly corrected current formulation will be demonstrated.

It is important to note that the traditional Rhie-Chow scaling, which requires the "old" mass flow rates at integration points still requires the framework capability of data structure restart support at integration points.

The derivation begins with the development of the "dt" scaling algorithm adopted by Jones. For completeness, the following derivation is again repeated.

The form of the convecting velocity as derived from a semi-discrete form of the momentum equations is similar, except that the pressure gradient scaling term is the limiting value for small time step. With this form, the transient correction term is not required.

The derivation begins with the semi-discrete formulation of the momentum equations,

$$\rho_i{}^{n+1} U_i{}^{n+1} = \rho_i{}^{n} U_i{}^{n} + \Delta t \left( F_i{}^{n+1} + b_i \right) - \Delta t \nabla p_i^* \tag{11.1}$$

where $F_i^{n+1}$ represents the convection and diffusion terms and $b_i$ contains the non-solenoidal stress and any potential buoyancy terms. An analogous form of a semi-discrete integration point velocity can be written as,

$$\rho_{ip}^{n+1} u_{ip}^{n+1} = \rho_{ip}^{n} u_{ip}^{n} + \Delta t \left( F_{ip}^{n+1} + b_{ip} \right) - \Delta t \nabla p_{ip}^{n+1} \tag{11.2}$$

Equation 11.1 is rearranged to provide the following term in Equation 11.2,

$$\rho_{ip}^{n} u_{ip}^{n} + \Delta t \left( F_{ip}^{n+1} + b_{ip} \right) = \overline{\rho_i{}^{n} U_i{}^{n} + \Delta t \left( F_i{}^{n+1} + b_i \right)} = \overline{\rho_i{}^{n+1} U_i{}^{n+1} + \Delta t \nabla p_i^*} \tag{11.3}$$

The results from Equation 11.3 are substituted within Equation 11.2 to yield the final form of the face mass flow rate,

$$\rho_{ip}^{*} u_{ip}^{n+1} = \sum_i N_i \rho_i^{*} U_i^{*} + \Delta t \left[ \sum_i N_i \sum_j G_{ij} p_j^{*} - \nabla p_{ip}^{n+1} \right] \tag{11.4}$$

The above equation is approximated and implemented within Fuego as,

$$\rho_{ip}^{*} u_{ip}^{n+1} = \rho_{ip}^{*} \sum_i N_i U_i^{*} + \Delta t \left[ \sum_i N_i \sum_j G_{ij} p_j^{*} - \nabla p_{ip}^{n+1} \right] \tag{11.5}$$

where the $^{*}$ represents the provisional scalar value; the most current density is discussed in Section 5.3.2.

In the proposed method, however, let us not make the assumption of somewhat constant density thereby retaining Equation 11.4 as the form of the convecting velocity.

It is now important to note that the use of the convecting velocity formula based on interpolated values from the momentum field includes a error term that is due to the fact that the discrete momentum equation was solved to a user specified tolerance. In reality, this is not such a great issue when compared to the staggered grid community as the convecting velocities used for the continuity equation are the velocities that result from the momentum solve.

Equation 11.4 is substituted within the discrete continuity equation to form the pressure equation. Once the continuity equation is solved, the new pressure field is first applied within Equation 11.4, to obtain the conserved mass flow rates and then within the nodal velocity correction to obtain the appropriate nodal velocity field, Equation 11.6

$$\rho_i^{*} U_i^{n+1} = \rho_i^{*} U_i^{*} - \Delta t \sum_j G_{ij} \left( p_j^{n+1} - p_j^{*} \right) \tag{11.6}$$

It is now proposed that the equivalent form of Equation 11.4 can be written as,

$$\rho_{ip}^{*} u_{ip}^{n+1} = \sum_i N_i \rho_i^{*} U_i^{n+1} + \Delta t \left[ \sum_i N_i \sum_j G_{ij} p_j^{n+1} - \nabla p_{ip}^{n+1} \right] \tag{11.7}$$

where it is noted that provisional values are substituted by the projected variables.

Upon adoption of the above equation, the mass conserving flow rates can be determined by the latest projected velocity and pressure field. Therefore, without approximation, the appropriate mass flow rates can be computed upon restart. This statement, the equivalence of Equation 11.7 and Equation 11.4, is easily verified by substitution of the rearranged nodal correction equation, Equation 11.6, within Equation 11.7,

$$\rho^*_{ip} u^{n+1}_{ip} = \sum_i N_i \rho^*_i U^*_i - \Delta t \sum_i N_i \sum_j G_{ij} p^{n+1}_j + \Delta t \sum_i N_i \sum_j G_{ij} p^*_j + \Delta t \sum_i N_i \sum_j G_{ij} p^{n+1}_j - \Delta t \nabla p^{n+1} \tag{11.8}$$

$$= \sum_i N_i \rho^*_i U^*_i + \Delta t \left[ \sum_i N_i \sum_j G_{ij} p^*_j - \nabla p^{n+1}_{ip} \right] \tag{11.9}$$

Therefore, having shown the equivalence of these two equations it seems that the ability to use restart for the dt scaling algorithm is complete.

Note that use of Equation 11.7 would require a different placing of the nodal pressure gradient evaluation and a slightly different weighting of the nodal mass flow rate within the velocity interpolation routine.

At this point, I do not recommend that we change the form of the integration point velocity formula other than to include the proper presumption that density is not constant, Equation 11.4. In fact, due to inconsistencies between the presumption of constant density, e.g., Equation 11.5, the equivalence between Equation 11.5 and Equation 11.7 can not be demonstrated. However, the justification for this shortcut approach has never been justified by a sensitivity of this interpolation in variable density flows.

Therefore, Equation 11.7 will only be used upon restart.

# 12. VIRTUAL THERMOCOUPLE MODEL

The purpose of the virtual thermocouple model as implemented in Fuego is to approximate the temperature that would be obtained from a thermocouple, given the results of a CFD simulation.

# APPENDIX A. THEORETICAL DESCRIPTION OF THE MODEL

Neglecting conduction through the thermocouple, the governing equations describing heat transfer to the thermocouple are written as

$$\rho c_p V_{tc} \frac{\partial T}{\partial t} = \nabla \cdot \mathbf{q}_r - h A_{tc} \left( T - T_\infty \right), \tag{12.1}$$

or alternatively,

$$\int \rho c_p \frac{\partial T}{\partial t} \, dV = \oint \mathbf{q}_r \cdot \mathbf{n} \, dA - \oint h \left( T - T_\infty \right) dA, \tag{12.2}$$

where $\mathbf{q}_r$ is the radiative heat flux vector, $h$ is the turbulent heat transfer coefficient, and $T_\infty$ is the surrounding gas temperature. In equation (12.2), the integrals are evaluated over the surface of the thermocouple.

## A.1. Convective Heat Flux

The heat transfer coefficient is given in terms of the Nusselt number, the gas phase thermal conductivity, and the pertinent thermocouple length scale, $\ell$, as

$$Nu = \frac{h\ell}{\lambda} \tag{12.3}$$

### *Correlations for Nusselt Number*

The Nusselt number is given as a function of the Reynolds and Prandtl numbers by the following correlation, obtained from Incropera & Dewitt (1996) for a cylinder in cross-flow:

$$Nu = 0.3 + \frac{0.62 Re^{1/2} Pr^{1/3}}{\left[1 + (0.4/Pr)^{2/3}\right]^{1/4}} \left[ 1 + \left( \frac{Re}{282,000} \right)^{5/8} \right]^{4/5}, \tag{12.4}$$

with the Reynolds number and Prandtl number given by

$$Re = \frac{u_g \ell \rho_g}{\mu_g}, \qquad Pr = \frac{c_{p_g} \mu_g}{\lambda_g}. \tag{12.5}$$

The subscript $g$ emphasizes that these properties are evaluated in the gas phase. The length scale, $\ell$, is the thermocouple diameter. The velocity, $u_g$, is the component of the gas velocity perpendicular to the thermocouple.

## A.2. Radiative Heat Flux

Given the incident spectral radiation intensity field, $I = \int_0^\infty I_\lambda \, d\lambda$, the total radiative heat flux in direction $\mathbf{s}$ may be obtained as

$$\mathbf{q}_r = \int_0^{2\pi} \int_0^{\pi} I(\theta, \phi) \, \mathbf{s} \cos\theta \sin\theta \; d\theta \; d\phi, \tag{12.6}$$

where $\mathbf{s}$ is the directional vector,

$$\mathbf{s} = \sin\theta \sin\phi \mathbf{i} + \cos\theta \mathbf{j} + \sin\theta \cos\phi \mathbf{k}, \tag{12.7}$$

as depicted in figure A-1. However, we are interested in only the heat flux incident on the faces of the

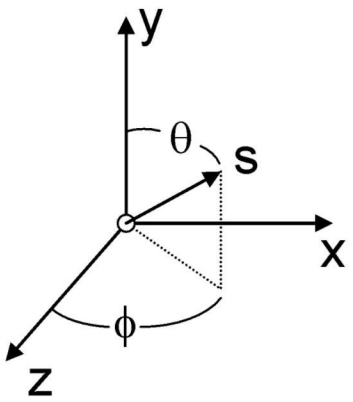


**Figure A-1.. Coordinate system, showing the vector $\mathrm{s}$**

control volume which contain the thermocouple. The incident heat flux on any surface, $q_r^I$, may be obtained as

$$q_r^I = \int_0^{2\pi} \int_0^{\pi} I(\theta, \phi) \, F(\theta, \phi) \cos\theta \sin\theta \; d\theta \; d\phi, \tag{12.8}$$

where

$$F(\theta, \phi) \equiv \max\left(0, -\mathbf{n} \cdot \mathbf{s}\right), \tag{12.9}$$

and $\mathbf{n}$ is the outward-pointing unit-vector normal for the given surface.

Assuming that the thermocouple emits radiation according to the Plank distribution, the net radiative heat flux at any point on the thermocouple may be written as

$$q_r = \alpha_c \, q_r^I - \varepsilon_c \sigma T^4, \tag{12.10}$$

where $\alpha_c$ and $\varepsilon_c$ are, respectively, the absorptivity and emissivity of the thermocouple, and $q_r^I$ is given by (12.8).

## A.3. Working Assumptions

There are several working assumptions:

1. The thermocouple does not affect the flow field in any way (i.e. one-way coupling to the model). This implies the next assumption:
2. The thermocouple exists entirely within a single computational cell. Ideally, the thermocouple dimensions should be small relative to the computational mesh. Otherwise, the assumption that the thermocouple does not affect the flow field is invalid.
3. Negligible attenuation of the irradiation between the edge of the computational cell and the thermocouple surface,
4. Negligible conduction along the thermocouple.
5. Spectral emission from the thermocouple is assumed to follow the Plank distribution, i.e. the total emissive power is proportional to $\sigma T^4$.
6. All thermocouple properties are homogeneous and constant (do not vary with space, or time).
7. The emissivity and absorptivity of the thermocouple are equal, $\varepsilon_c = \alpha_c$.
8. The heat transfer coefficient and convective temperature, $T_\infty$ are homogeneous over the thermocouple.
9. The thermal conductivity of the thermocouple is sufficiently large that conduction through the thermocouple is fast relative to the convective and radiative time scales. This implies that the thermocouple is at a single, uniform temperature.

Several of these assumptions stem from the assumption that the thermocouple exists within a single computational cell. For a more detailed treatment, details of the thermocouple geometry must be specified to a level where meshing the thermocouple itself may be required.

# APPENDIX B. MODEL IMPLEMENTATION

This section describes the implementation of the model, including the user interface, numerical discretization, and solution strategy.

## B.1. User Interface

The following quantities must be supplied by the user:

- Orientation. The three components of the thermocouple orientation vector must be specified. The orientation vector need not be a unit-normal; it will be normalized internally.
- Diameter and length of the thermocouple.
- Heat capacity, density, and emissivity of the thermocouple.
- Initial temperature of the thermocouple.
- Optionally, the user may request that the steady solution for the thermocouple temperature be found, rather than the transient solution. In this case, the initial temperature of the thermocouple is not required.

## B.2. Discrete Equations

### *Fully Discretized Governing Equation*

Given the assumptions listed in §A.3, the fully discrete equation for the thermocouple temperature may be obtained using a backward-Euler difference in time as

$$\rho c_p V_{tc} \frac{T^{n+1} - T^n}{\Delta t} = \sum_i A^p_{tc,i} \left( \alpha_c \, q^I_{r,i} - \varepsilon_c \sigma (T^{n+1})^4 \right) - h A_{tc} \left( T^{n+1} - T_\infty \right), \tag{12.11}$$

where $V_{tc} = \pi D^2 L/4$ is the thermocouple volume and $A_{tc} = \pi D(D/2 + L)$ is the thermocouple surface area.

Equation (12.11) may be rewritten as a fourth-order polynomial in $T^{n+1}$:

$$a(T^{n+1})^4 + bT^{n+1} + c = 0, \tag{12.12}$$

with coefficients given by

$$a = \varepsilon_c \sigma \sum_i A^p_{tc,i}, \tag{12.13}$$

$$b = \frac{\rho c_p V_{tc}}{\Delta t} + h A_{tc}, \tag{12.14}$$

$$c = -\frac{\rho c_p V_{tc}}{\Delta t} T^n - h A_{tc} T_\infty - \sum_i A^p_{tc,i} \alpha_c \, q^I_{r,i}. \tag{12.15}$$

If requested, the model will compute the steady-state solution to (12.11) by replacing (12.14) and (12.15) with

$$b = h A_{tc}, \tag{12.16}$$

$$c = -h A_{tc} T_\infty - \sum_i A^p_{tc,i} \alpha_c \, q^I_{r,i}. \tag{12.17}$$

## B.3. Solution Procedure

The solution procedure may be outlined as follows

1. Obtain $t_{end}$ - the time increment over which we wish to update the thermocouple temperature. This is nominally the timestep determined from the procedure.
2. Select $\Delta t$. This should be chosen such that the time-integration of the thermocouple temperature is sufficiently accurate. Details are discussed in §B.5.
3. Compute $a$ from (12.13).
4. Compute the projected thermocouple area, $A^p_{tc,i}$. This is discussed in greater detail in §B.4.
5. Compute the convective heat transfer coefficient, $h$.
6. Compute the contribution to $c$ from the incident radiative flux: $\sum_i A^p_{tc,i} \alpha_c \, q^I_{r,i}$.
7. Compute $b$ from (12.14).
8. Set $t = 0$. Set $T^0$ via the supplied initial condition or using the value from the previous solution.
9. `while` $t < t_{end}$
   - Compute $c$ from (12.15).
   - Solve (12.12) for $T^{n+1}$. This is obtained using Newton's method.
   - Advance time: $t = t + \Delta t; \quad T^n = T^{n+1}$.

## B.4. Determining the Projected Thermocouple Area

We must compute the area of the thermocouple projected on each CV face for use in equation (12.15). The total thermocouple area must be projected to each CV face to determine $A^p_{tc,i}$.

Given the surface unit normal, $\mathbf{n}_i$ and the thermocouple orientation unit-vector $\mathbf{p}$, we may write the projected thermocouple area on face $i$ as

$$A^p_{tc,i} = [A_{end}(\mathbf{p} \cdot \mathbf{n}_i)] + [A_{side}(\mathbf{m}_i \cdot \mathbf{n}_i)] \tag{12.18}$$

where

$$A_{end} = \frac{\pi}{4} D^2, \tag{12.19}$$

$$A_{side} = DL, \tag{12.20}$$

$\mathbf{m}_i$ is a unit-vector perpendicular to $\mathbf{p}$ (i.e. $\mathbf{m}_i \cdot \mathbf{p} = 0$), $L$ is the thermocouple length, and $D$ is the diameter, as given by the user. The vector $\mathbf{m}_i$ represents the area unit-vector for the side of the thermocouple and is given as

$$\mathbf{m}_i = \frac{\mathbf{n}_i - (\mathbf{p} \cdot \mathbf{n}_i)\, \mathbf{p}}{|\mathbf{n}_i - (\mathbf{p} \cdot \mathbf{n}_i)\, \mathbf{p}|}. \tag{12.21}$$

The projected area, $A^p_{tc,i}$, is constrained by

$$0 < \frac{A^p_{tc,i}}{a_i} \leq 1, \tag{12.22}$$

where $a_i$ is the full area of CV face $i$. This implies constraints on both the thermocouple length and diameter. Specifically, the thermocouple must fit within a CV. The length and diameter may be no greater than what will fit in the CV given the thermocouple orientation.

The constraint mentioned above must be enforced within each CV. Note that for the model assumptions listed in §A.3 to be valid, we really require that $A^p_{tc,i}/a_i \ll 1$. This is not currently enforced. The only constraint currently imposed is that $A^p_{tc,i}/a_i \leq 1$. Currently, the code will issue warnings if $A^p_{tc,i}/a_i > 0.1$.

## B.5. Selection of Timestep

The timestep is selected based on the minimum of three criteria:

$$\Delta t = \min(\tau_c, \tau_r, \tau_t) \tag{12.23}$$

where $\tau_c$ is the convection timescale, $\tau_r$ is the radiation timescale, and $\tau_t = t_{end}/n_{min}$. In other words, $\tau_t$ is a timescale that is defined by the minimum number of timesteps that should be taken over time interval $[0, t_{end}]$.

The convective timescale is obtained from the analytic solution of the pure convective problem

$$T(t) = T_\infty + [T(0) - T_\infty] \exp\left(\frac{hA_{tc}}{\rho c_p V_{tc}} t\right), \tag{12.24}$$

implying that

$$\tau_c = \frac{\rho c_p V_{tc}}{hA_{tc}}. \tag{12.25}$$

The radiative timescale is currently determined in a very heuristic manner as

$$\tau_r = \left(\frac{\rho c_p V_{tc}}{\sigma A_{tc}}\right)^{1/4}. \tag{12.26}$$

# DISTRIBUTION

**Email—Internal (encrypt for OUO)**

| Name | Org. | Sandia Email Address |
|---|---|---|
| Technical Library | 01177 | libref@sandia.gov |

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.