SAND2010-1168C

Stabilized Finite Element Scheme for High Speed Flows with Chemical Non-Equilibrium

Steven W. Bova[*] and Ryan B. Bond[†]

Sandia National Laboratories, P. O. Box 5800, Albuquerque, NM, 87185, USA

and

Benjamin S. Kirk[‡]

NASA Lyndon B. Johnson Space Center, Houston, TX, 77058, USA

A streamline upwind Petrov-Galerkin finite element method is presented for the case of a reacting mixture of thermally-perfect gases, using chemical non-equilibrium. Details of the stabilization scheme and nonlinear solution are presented. The authors have independently implemented the proposed algorithm in two separate codes, for both single temperature and and two temperature models. Example problems invoving a cylinder in Mach 20 crossflow, as well as a three-dimensional blunt nosetip are shown and compared to established codes.

Nomenclature

$\boldsymbol{\mathcal{F}}$	vector of source/sink terms in Navier-Stokes equations
$\boldsymbol{\mathcal{L}}$	differential operator of homogeneous Navier-Stokes equations
$\boldsymbol{\tau}_s$	matrix of intrinsic time scales for SUPG operator
$\boldsymbol{A_0}^{-1}$	inverse of symmetrizer matrix, used as metric for inner products
$\boldsymbol{F}_i$	Euler flux in direction i
$\boldsymbol{G}_i$	viscous flux in direction i
$\boldsymbol{J}$	Jacobian of nonlinear residual
$\boldsymbol{r}$	nonlinear residual
$\boldsymbol{U}$	state vector of conserved variables
$\boldsymbol{W}$	vector of test functions in weighted residual statement of Navier-Stokes equations
$\mathcal{C}$	range of master element parametric coordinates
χ_s	partial derivative of pressure with respect to the species density at constant temperature
δ_{ij}	Kronecker's delta
γ	ratio of specific heats
$\hat{\nu}$	interpolant of nodal reconstruction of residual-based shock-capturing parameter
$\hat{n}_i$	component of outward unit normal vector in direction i
κ	partial derivative of pressure with respect to internal energy at constant mixture density
μ	dynamic viscosity
ν	residual-based shock-capturing parameter
Ω	computational domain
$\partial\Omega$	computational domain boundary
ϕ_m	finite element basis function associated with node m
ρ	mixture density
ρ_s	density of species s
τ_c	time scale associated with continuity equation
τ_e	time scale associated with energy equation
τ_m	time scale associated with momentum equation

[*] Computational Thermal and Fluid Mechanics Department, MS 0382, Senior Member, AIAA.

[†] Aerosciences Department, MS 0825, Senior Member, AIAA.

[‡] Applied Aeroscience and CFD Branch, 2101 NASA Parkway, Mail Code EG3, Member AIAA

τ_{ij}	viscous stress tensor
ξ_k	master element parametric coordinate in direction k
c_p	specific heat at constant pressure for ideal gas or mixture
c_v	specific heat at constant volume for ideal gas or mixture
c_{vs}	specific heat at constant volume for species s
D_s	diffusivity of species s
E	total energy per unit mass
e	internal energy per unit mass
g^{ij}	element contravariant metric tensor
g_{ij}	element covariant metric tensor
h_s	static enthalpy of species s
$h_{\boldsymbol{u}}$	flow-aligned element length scale
$h_{s,0}$	heat of formation of species s
k	thermal conductivity
P	pressure
P_r	Prandtl number
q_i	heat flux vector
Q_v	translational/vibrational energy exchange rate
R	ideal gas constant
R_s	ideal gas constant for species s
T	temperature
t	time
u_i	velocity component in direction i
x_i	Cartesian coordinate direction i
y_s	mass fraction of species s

1. Introduction

The structured-mesh finite volume scheme has generally been the preferred method for solving hypersonic aerodynamics problems for the past two decades. The scheme extends easily to second-order on structured meshes, and when appropriate limiters are used and the mesh is aligned with the shock, 1-D Riemann-solver-based fluxes provide good shock capturing behavior. The past two decades have also seen research into alternative methods, including unstructured-mesh finite volume schemes, the discontinuous Galerkin finite element method, and the streamline upwind Petrov-Galerkin (SUPG) finite element method. The finite element method is well-suited for implementation on unstructured grids, and readily extends to h-adaptive and non-homogeneous meshes. Although most work in the computational aerodynamics community has been directed towards finite volume discretizations, robust and accurate formulations on arbitrary meshes has remained elusive. The goal of our work is to develop fully coupled algorithms for solving hypersonic heat transfer problems, including detailed thermal analyses of the flight vehicle. This paper demonstrates that recent advances in SUPG make it a viable alternative to unstructured finite volume methods. In the present work, we propose an SUPG algorithm for chemically reacting flows. We have implemented this algorithm in two separate codes. Numerically, these two codes differ primarily in their method of nonlinear solution. The Aria code, which is under development at Sandia National Laboratories, explicitly assembles the Jacobian of the nonlinear residual at each time step. The FIN-S code,[1] which is under development at NASA Johnson Space Center, uses a Newton-Krylov strategy and never assembles the Jacobian of the nonlinear residual.

2. Mathematical Model

2.1. Governing Equations

Under certain assumptions (e.g. see[2]), the conservation of mass, momentum, and energy for a compressible fluid may be written as

$$\begin{aligned}\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u_i)}{\partial x_i} &= 0 \qquad (1)\\ \frac{\partial (\rho u_j)}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i u_j + P\delta_{ij}) &= \frac{\partial \tau_{ij}}{\partial x_i} \qquad (2)\\ \frac{\partial (\rho E)}{\partial t} + \frac{\partial}{\partial x_i}\left[(\rho E + P)\, u_i\right] &= \frac{\partial q_i}{\partial x_i} + \frac{\partial (u_j \tau_{ij})}{\partial x_i}, \qquad (3)\end{aligned}$$

The total energy per unit mass, E, may be written as the sum of the internal energy, e and the kinetic energy,

$$E = e + \frac{u_k u_k}{2}. \tag{4}$$

In order to close this set of equations, constitutive relations are necessary for τ_{ij} and the heat flux vector q_i. For a Newtonian fluid, we may write

$$\begin{aligned}\tau_{ij} &= \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{2}{3}\mu\frac{\partial u_k}{\partial x_k} \qquad (5)\\ q_i &= -k\frac{\partial T}{\partial x_i}, \qquad (6)\end{aligned}$$

where we have introduced the viscosity, μ, themal conductivity, k, and temperature, T. For many flows of engineering interest, an empirical relation such as Sutherland's Law

$$\mu = \mu_{\text{ref}}\,\frac{T^{\frac{3}{2}}}{T + T_{\text{ref}}} \tag{7}$$

or Keyes's Law

$$\mu = 10^{-6}\frac{a_0 T^{\frac{3}{2}}}{T + 10^{\frac{-a_1}{T}} a} \tag{8}$$

is used. In (8), if $a_0 = 1.488$, $a_1 = 122.1$, and $a = 5$, then the viscosity of air is modelled in SI units of kg/(m-s). Finally, the thermal conductivity is given by a constant Prandtl number model,

$$k = \frac{\mu c_p}{P_r} \tag{9}$$

where c_p is the specific heat at constant pressure. For air, $P_r = 0.71$.

2.2. Ideal Gas Model

For flows at low to moderate Mach numbers, say up to Mach 8 or so, the ideal gas assumption is reasonable. In this case,

$$P = \rho R T \tag{10}$$

where the gas constant, R, is given by the difference of specific heats at constant pressure and volume: $R = c_p - c_v$. It may be shown that the internal energy is proportional to the temperature: $e = c_v T$, and that the speed of sound, c, is given by $c = \sqrt{\gamma R T}$, where $\gamma = \frac{c_p}{c_v}$.

2.3. Non Equilibrium Chemically Reacting Gas Model

For flows at Mach numbers higher than about Mach 8, the ideal gas model produces temperatures that are unreasonably high. For this case (10) is not valid, and we model the gas as a chemically reacting mixture.

For this case, the mixture density, ρ, is a sum of the species densities, which may be written as

$$\rho = \sum_{s=1}^{\mathrm{S}} \rho_s \tag{11}$$

Since the species are assumed to be thermally-perfect, Dalton's law of partial pressures holds, namely

$$P = \sum_{s=1}^{\mathrm{S}} \rho_s R_s T. \tag{12}$$

However, high temperatures vibrationally excite the gas species, so a calorically-perfect assumption is inappropriate. Instead, each species has specific heats that are functions of temperature. The species enthalpy is the integral of the specific heat at constant pressure from the reference temperature up to the mixture temperature plus the heat of formation of that species, namely

$$h_s(T) = h_{s,0} + \int_{T_r}^{T} C_p(\theta) d\theta. \tag{13}$$

The NASA Glenn curve fits and coefficient database[3] are used for these species-specific quantities.

Each species is governed by its own continuity equation, which includes mass diffusion for viscous problems. Hence we may write

$$\frac{\partial \rho_s}{\partial t} + \frac{\partial (\rho_s u_i)}{\partial x_i} = \frac{\partial}{\partial x_j}\left(\rho D_s \frac{\partial y_s}{\partial x_i}\right) + \dot{w}_s \tag{14}$$

The diffusivity of each species is determined by a constant Schmidt number (Sc). This mass diffusion also introduces an additional term to the energy equation, since the diffusion of species with different enthalpies results in an energy flux. The energy equation for a chemically reacting mixture may therefore be written as

$$\frac{\partial (\rho E)}{\partial t} + \frac{\partial}{\partial x_i}\left[(\rho E + P) u_i\right] = \frac{\partial q_i}{\partial x_i} + \frac{\partial (u_j \tau_{ij})}{\partial x_i} + \frac{\partial}{\partial x_j}\left(\rho \sum_{s=1}^{\mathrm{S}} h_s D_s \frac{\partial y_s}{\partial x_i}\right). \tag{15}$$

The symmetric form of a two temperature model for the chemically reacting Navier-Stokes equations, which can be used as a starting point for a stabilized finite element discretization, was published by Chalot et al.[4] This form was later simplified for the case of chemical equilibrium, and a finite element method was proposed with supporting numerical results.[5] The present results are the first presented for the case of nonequilibrium chemistry (to the knowledge of the authors). The only change to the numerics of the SUPG scheme required by these additional physics involves the Jacobian and inverse symmetrizer matrices, which are presented in the Appendix.

Although the two temperature model has not yet been implemented in the Aria code, it has been implemented in the FIN-S code, although currently the Arrhenius rates are determined exclusively from the translational temperature, T. When a two temperature model is used, the additional vibrational energy equation

$$\frac{\partial (\rho e^{\mathrm{vib}})}{\partial t} + \frac{\partial}{\partial x_i}\left[(\rho e^{\mathrm{vib}}) u_i\right] = \frac{\partial q_i^{\mathrm{vib}}}{\partial x_i} + \frac{\partial}{\partial x_j}\left(\rho \sum_{s=1}^{\mathrm{S}} e^{\mathrm{vib}} D_s \frac{\partial y_s}{\partial x_i}\right) + \sum_{s=1}^{\mathrm{S}} e_s^{\mathrm{vib}} \dot{w}_s + Q^{\mathrm{vib}} \tag{16}$$

is solved.

3. Stabilized Finite Element Method

Before proceeding, it is useful to recast (1)-(3) in vector form. To this end, let

$$\boldsymbol{U} = \begin{pmatrix} \rho \\ \rho u_j \\ \rho E \end{pmatrix} \tag{17}$$

$$\boldsymbol{F}_i(\boldsymbol{U}) = \begin{pmatrix} \rho u_i \\ \rho u_i u_j + P\delta_{ij} \\ (\rho E + P) u_i \end{pmatrix} \tag{18}$$

$$\boldsymbol{G}_i(\boldsymbol{U}) = \begin{pmatrix} 0 \\ \tau_{ij} \\ -q_i + \tau_{ij} u_j \end{pmatrix} \tag{19}$$

In the above definitions, the subscript $()_i$ denotes the coordinate direction associated with each flux vector $\boldsymbol{F}_i$ and $\boldsymbol{G}_i$. The subscript $()_j$ denotes the component of the momentum equation, which expands the length of each vector according to the spatial dimension: e.g., for two spatial dimensions, $\boldsymbol{U}$, $\boldsymbol{F}_i$, and $\boldsymbol{G}_i$ are each of length four. Now (1)-(3) may be written as

$$\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial \boldsymbol{F}_i(\boldsymbol{U})}{\partial x_i} - \frac{\partial \boldsymbol{G}_i(\boldsymbol{U})}{\partial x_i} = \boldsymbol{\mathcal{F}}, \tag{20}$$

where we have introduced the vector $\boldsymbol{\mathcal{F}}$ to account for the mass and energy source terms in (14), (15), and (16). Let

$$\boldsymbol{\mathcal{L}U} = \frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial \boldsymbol{F}_i(\boldsymbol{U})}{\partial x_i} - \frac{\partial \boldsymbol{G}_i(\boldsymbol{U})}{\partial x_i} \tag{21}$$

so that (20) may be written as the residual equation

$$\boldsymbol{\mathcal{L}U} - \boldsymbol{\mathcal{F}} = \boldsymbol{0} \tag{22}$$

3.1. Weak Form

The stabilized finite element method that we use is a variant of the so-called SUPG (streamline upwind Petrov-Galerkin) method.[6–11] First, we approximate $\boldsymbol{U}$ by introducing the trial functions $\{\phi^m\}$, and expand $\boldsymbol{U}$ and $\boldsymbol{F}_i$ in this basis. Thus we may write

$$\boldsymbol{U}_h \simeq \boldsymbol{U}$$

$$\boldsymbol{U}_h(x_k,t) = \sum_{m=1}^{N} \boldsymbol{U}^m(t)\phi^m(x_k) \tag{23}$$

$$\boldsymbol{F}_i(x_k,t) = \sum_{m=1}^{N} \boldsymbol{F}_i(\boldsymbol{U}^m(t))\phi_m(x_k) \tag{24}$$

where ϕ_m is the m-th function in our finite element basis, and N is the number of nodes. In this work, we exclusively use the linear Lagrange basis. Note that, since the fluxes are nonlinear functions of $\boldsymbol{U}$, there is a choice to be made. A person can first compute the fluxes using the nodal values $\boldsymbol{U}^m(t)$ and then interpolate the result, as indicated in (24). Or, a person can interpolate $\boldsymbol{U}_h$ to the desired spatial location and then calculate the flux of this interpolant. It has been shown that the approach indicated by (24) has improved stability.[12]

Now, the problem may be stated as follows: Find the approximate discrete solution $\boldsymbol{U}_h \simeq \boldsymbol{U}$ such that

$$\int_\Omega \boldsymbol{W} \cdot (\boldsymbol{\mathcal{L}U}_h - \boldsymbol{\mathcal{F}}) \quad d\Omega + \int_\Omega \frac{\partial \boldsymbol{W}}{\partial x_i} \cdot \boldsymbol{A}_i \boldsymbol{\tau}_s (\boldsymbol{\mathcal{L}U}_h - \boldsymbol{\mathcal{F}}) \quad d\Omega + \int_\Omega \nu \frac{\partial \boldsymbol{W}}{\partial x_i} \cdot g^{ij} \frac{\partial \boldsymbol{U}_h}{\partial x_j} \quad d\Omega = \boldsymbol{0} \tag{25}$$

for all admissible test functions $\boldsymbol{W}$. In (25), the first integral is associated with Galerkin's method, which is analogous to a central finite difference scheme. The second integral is the SUPG stabilization term, and the third integral is the so-called discontinuity-capturing operator (DCO). We have introduced the flux Jacobian matrices $\boldsymbol{A}_i = \frac{\partial \boldsymbol{F}_i}{\partial \boldsymbol{U}}$, explicit forms of which appear in the Appendix. The stabilization parameter $\boldsymbol{\tau}_s$, which also appears in (25) plays a key role in the method. We discuss this important parameter further in the text below. The DCO is written in terms of gradients in computational space by virtue of the contravariant metric tensor

$$g^{ij} = \frac{\partial x_i}{\partial \xi_k}\frac{\partial x_j}{\partial \xi_k} \tag{26}$$

where ξ_k is the k-th parametric coorinate of the master element. Consistency with the original partial differential equation system is maintained by ν, which is a scalar function of the residual. Different choices for ν exist.[11,13] Note that ν has units of (time)$^{-1}$. In this work, we use

$$\nu = \sqrt{\frac{\left(\boldsymbol{\mathcal{L}}\boldsymbol{U}_h - \boldsymbol{\mathcal{F}}\right)^t \boldsymbol{A_0}^{-1}\left(\boldsymbol{\mathcal{L}}\boldsymbol{U}_h - \boldsymbol{\mathcal{F}}\right)}{\left(\Delta\boldsymbol{U}_h\right)^t \boldsymbol{A_0}^{-1}\Delta\boldsymbol{U}_h + g^{ij}\left(\frac{\partial \boldsymbol{U_h}}{\partial x_i}\right)^t \boldsymbol{A_0}^{-1}\frac{\partial \boldsymbol{U}_h}{\partial x_j}}} \tag{27}$$

In (27), $\Delta\boldsymbol{U}_h$ represents the change of $\boldsymbol{U}_h$ from one time step to the next and in practice is calculated via

$$\Delta\boldsymbol{U}_h = \frac{\partial \boldsymbol{U}_h}{\partial t}\Delta t$$

The symmetric, positive definite matrix $\boldsymbol{A_0}^{-1}$ is associated with the transformation from entropy variables to conservation variables $\boldsymbol{U}$, and arises as a natural metric for inner products. It is important to note that this matrix is the second derivative of a generalized entropy function with respect to $\boldsymbol{U}$, but we leave the details to the literature.[4,11] This matrix is given in the Appendix for both ideal and chemically reacting gases.

The weak form of our finite element discretization is obtained by returning to (25) and integrating the Galerkin terms involving the fluxes $\boldsymbol{F}_i$ and $\boldsymbol{G}_i$ by parts. After performing the required manipulations, we obtain

$$\int_\Omega \boldsymbol{W}\cdot\left(\frac{\partial \boldsymbol{U}_h}{\partial t} - \boldsymbol{\mathcal{F}}\right)\,d\Omega + \int_\Omega \frac{\partial \boldsymbol{W}}{\partial x_i}\cdot\left(\boldsymbol{G}_i - \boldsymbol{F}_i\right)\,d\Omega + \int_{\partial\Omega}\boldsymbol{W}\cdot\left(\boldsymbol{F}_i - \boldsymbol{G}_i\right)\hat{n}_i\,d\Gamma + \int_\Omega \frac{\partial \boldsymbol{W}}{\partial x_i}\cdot\boldsymbol{A}_i\boldsymbol{\tau}_s\left(\boldsymbol{\mathcal{L}}\boldsymbol{U}_h - \boldsymbol{\mathcal{F}}\right)\,d\Omega + \int_\Omega \nu\frac{\partial \boldsymbol{W}}{\partial x_i}\cdot g^{ij}\frac{\partial \boldsymbol{U}_h}{\partial x_j}\,d\Omega = \boldsymbol{0} \tag{28}$$

where $\partial\Omega$ is the boundary of Ω, and $\hat{n}_i$ is the i-th component of the unit vector normal to and pointing out of $\partial\Omega$. It should be noted that, if linear basis functions are used, then the second derivative terms in the definition of $\boldsymbol{\mathcal{L}}\boldsymbol{U}_h$ vanish in the stabilization terms, since they are not integrated by parts.

3.2. Stabilization Parameter

In order to complete the method, a definition for $\boldsymbol{\tau}_s$ is needed. Conceptually, $\boldsymbol{\tau}_s$ is a matrix of intrinsic time scales associated with the discrete solution. There are different approaches to deriving this parameter (e.g. see[11,13,14]), and unfortunately the solution quality is strongly dependent on this choice. Recent developments in multiscale methods[8] have suggested a canonical formula for $\boldsymbol{\tau}_s$, but a unique practical definition remains elusive. In this work, we use a diagonal matrix, which in three dimensions may be written as

$$\boldsymbol{\tau}_s = \text{diag}\left(\tau_c, \tau_m, \tau_m, \tau_m, \tau_e\right) \tag{29}$$

where

$$\tau_c = \left[\left(\frac{(\|\boldsymbol{u}\| + c)}{h_{\boldsymbol{u}}}\right)^2 + \nu^2\right]^{-1/2} \tag{30}$$

$$\tau_m = \left[\left(\frac{1}{\tau_c}\right)^2 + \left(\frac{\mu}{\rho h_{\boldsymbol{u}}^2}\right)^2\right]^{-1/2} \tag{31}$$

$$\tau_e = \left[\left(\frac{1}{\tau_c}\right)^2 + \left(\frac{k}{\rho c_p h_{\boldsymbol{u}}^2}\right)^2\right]^{-1/2} \tag{32}$$

In (30)-(32), $h_{\boldsymbol{u}}$ is a flow-aligned length scale, which we define as[15]

$$h_{\boldsymbol{u}} = \mathcal{C}\sqrt{\frac{u_k u_k}{u_j \frac{\partial \xi_k}{\partial x_j} u_i \frac{\partial \xi_k}{\partial x_i}}} \tag{33}$$

where $\mathcal{C}$ is a constant that is determined by the range of computational coordinates in the master element. For example, if $-1 \leq \xi_k \leq 1$, then $\mathcal{C} = 2$. If $0 \leq \xi_k \leq 1$, then $\mathcal{C} = 1$. The definition (33) is clearly a flow aligned length scale once it is realized that the denominator is the norm of the projection of the velocity vector onto the gradient of the computational coordinates. We can rewrite (33) as

$$h_{\boldsymbol{u}} = \mathcal{C}\sqrt{\frac{u_k u_k}{u_i g_{ij} u_j}} \tag{34}$$

where

$$g_{ij} = \frac{\partial \xi_k}{\partial x_j}\frac{\partial \xi_k}{\partial x_i} = \left[g^{ij}\right]^{-1}$$

is the covariant metric tensor.

We remark that for inviscid problems, $\boldsymbol{\tau}_s$ reduces to the scalar τ_c. The first term in the square brackets in (30) is a measure of the time scale required for information to be convected across a finite element. The presence of ν in (30) is novel and may be somewhat controversial. Its purpose is to reduce the total artificial diffusion in the presence of shocks, so that the SUPG stabilization is reduced when the DCO is large. In our experience, adding this term significantly increases the accuracy across strong shocks. The second term in (31) is an estimate of the time scale associated with diffusion across an element. Similarly, the second term in (32) is an estimate of the time scale associated with the conduction of heat across an element. The inverse square norm serves to smoothly switch among the various time scales.

4. Implementation details

Generally speaking, the stabilized finite element method as presented in Section 3.1 works well for flows at moderate Mach numbers where the shocks are not very strong. However, for flows with free stream Mach numbers that exceed 4 or 5, problems begin to occur. For example, the method tries to capture shocks across a single element. As the shocks increase in strength, this begins to set up very large gradients. Since in general, the elements are not aligned with the shock, the shock “snaps” from one element to the next, and the mesh structure adversely affects the shape of the shock wave. This can generate fictitious waves that degrade accuracy. As the Mach number is further increased, these waves can generate numerical oscillations that can lead to instability. Furthermore, for the steady-state Euler equations, the solution to the energy equation under adiabatic conditions is that the total enthalpy is a constant. It is easy to show that the DCO in (28) does not preserve this property. This can lead to incorrect jump conditions across a shock. In this section, we present a few specific implementation details that have been designed to correct these deficiencies. Some of this work follows closely ideas developed by Kirk.[16]

4.1. Nodal reconstruction of stabilization parameters

In the absence of the DCO, the SUPG method is as its name suggests, a Petrov-Galerkin method in which the standard Galerkin basis is modified in an upwind direction. To see this, consider that removing the DCO from (25) results in

$$\int_\Omega \boldsymbol{W} \cdot \left(\boldsymbol{\mathcal{L}}\boldsymbol{U}_h - \boldsymbol{\mathcal{F}}\right)\, d\Omega + \int_\Omega \frac{\partial \boldsymbol{W}}{\partial x_i} \cdot \boldsymbol{A}_i \boldsymbol{\tau}_s \left(\boldsymbol{\mathcal{L}}\boldsymbol{U}_h - \boldsymbol{\mathcal{F}}\right)\, d\Omega = \boldsymbol{0}$$

After combining the two integrals, we may write

$$\int_\Omega \left(\boldsymbol{W} + \boldsymbol{\tau}_s^t \boldsymbol{A}_i^t \frac{\partial \boldsymbol{W}}{\partial x_i}\right) \cdot \left(\boldsymbol{\mathcal{L}}\boldsymbol{U}_h - \boldsymbol{\mathcal{F}}\right)\, d\Omega = \boldsymbol{0}$$

So it is clear that if $\boldsymbol{U}_h$ is expanded in the basis $\{\phi^m\}$, and, e.g., $\boldsymbol{W} = (\phi^m, 0, 0, 0, 0)^t$, then a test function for the continuity equation may be written as

$$\hat{\phi}^m = \phi^m + \boldsymbol{\tau}_s^t \boldsymbol{A}_i^t \frac{\partial \phi^m}{\partial x_i}$$

Hence the test function may be interpreted as the standard Galerkin function, plus a perturbation proportional to the gradient of the function. More specifically, the flux Jacobian matrices determine the direction of the upwinding, and $\boldsymbol{\tau}_s$ normalizes the size.

Now, a person has a choice as to where these two matrices are evaluated. For example, one obvious choice is to evaluate them at the quadrature points. Since the definition (33) involves the mesh metric terms, then $\boldsymbol{\tau}_s$ is discontinuous over the patch of elements that supports node m. This also means that the direction and magnitude of the upwinding that occurs at a given node will vary across the patch of its supporting elements. Numerical experiments have shown that it is more desirable to have the upwinding that is associated with a particular node be constant over the patch, so that each element contribution sees the same direction and magnitude of upwinding. The flux Jacobians, and all of the terms in (30)-(32), can be evaluated directly at the nodes, except for h_u. For this reason, we perform an L_2 projection to recover h_u at the nodes.

Let h_u be the discontinuous element values calculated according to (33). Then we seek a continuous reconstruction by projecting against our finite element basis and solving

$$\sum_{e=1}^{E} \int_{\Omega_e} \left(\sum_{m=1}^{m} h_u^m \phi^m - h_u \right) \, d\Omega_e = 0 \tag{35}$$

We solve (35) simultaneously with the conservation equations, in a decoupled fashion using a lumped mass approximation. The lumped mass approximation is necessary in order to ensure positivity of the reconstructed field. This procedure has the added benefit of forming a diagonal matrix so that the solution of (35) is fully explicit. Then these nodal values h_u^m are used to calculate (30)-(32) at the nodes. Finally, the flux Jacobians that multiply the test function gradiaent are also evaluated at the nodes. In this way the test function for node m has a consistent direction and scaling over its patch of elements. It should be emphasized that when the $\boldsymbol{A}_i$ appear in the residual, $\boldsymbol{\mathcal{L}}\boldsymbol{U}_h - \boldsymbol{\mathcal{F}}$, they are evaluated at the quadrature points, not at the nodes, because the residual must be sampled at the quadrature points for numerical integration.

4.2. Nodal reconstruction of ν

Recall that the weak formulation (28) involves the DCO,

$$\boldsymbol{I}_{DCO} = \int_{\Omega} \nu \frac{\partial \boldsymbol{W}}{\partial x_i} \cdot g^{ij} \frac{\partial \boldsymbol{U}_h}{\partial x_j} \, d\Omega \tag{36}$$

In this integral, the function ν plays the role of an artificial viscosity or diffusivity. Since ν is a function of the residual, which involves gradients, ν is necessarily discontinuous from one element to the next. If we consider an analogy to heat transfer, the heat flux normal to the interface between two materials is continuous, but if the diffusivities are discontinuous, then jumps in the temperature gradient will arise. The purpose of the DCO is to enable control over strong gradients and smooth the solution. But allowing ν to be discontinuous can lead to jumps in the solution gradients $\frac{\partial \boldsymbol{U}_h}{\partial x_j}$, which is counterproductive. Furthermore, it has recently been shown that a continuous artifical viscosity function is necessary to obtain higher-order accuracy in Discontinuous Galerkin methods, and can also lead to increased robustness for problems with strong shocks.[17] For these reasons, we perform an L_2 projection to recover a continuous field for ν, and then interpolate this field to the quadrature points to evaluate the DCO. This projection is performed exactly in the same way as described above for (35). More specifically, we solve

$$\sum_{e=1}^{E} \int_{\Omega_e} \left(\sum_{m=1}^{m} \hat{\nu}^m \phi^m - \nu \right) \, d\Omega_e = 0 \tag{37}$$

for the nodal values $\hat{\nu}^m$. We remark that this projection can be viewed as a special case of the PDE-based approach described by Barter and Darmofal.[17] Our piecewise discontinous representation for ν plays the role of their shock sensor, and the source term in their PDE is essentially our L_2 projection. We have no spatial diffusion term or time derivative terms: therefore we do not need to apply initial or boundary conditions on ν. Kirk[16] has used an even simpler approach: he assigns the maximum value of the artifical viscosity found in the patch of supporting elements to each node and interpolates this back to the quadrature points.

4.3. Preservation of enthalpy

The observation that the DCO involves an artificial viscosity term suggests another improvement. Jameson found that carelessly introducing artificial viscosity into the Euler equations can destroy an important property.[18] It may easily be shown that, for steady, adiabatic flows, the energy equation becomes a statement that the total enthalpy of the flow is a constant along a streamline. The energy equation may then be written as the continuity equation scaled by the total enthalpy. Now, if we consider (25) for the case $\boldsymbol{\tau}_s = \mathbf{0}$, it is easy to show that the energy equation does not satisfy this property. The reason is that the total enthalpy per unit mass

$$H = \frac{\rho E + P}{\rho}$$

appears in the energy flux, but we are smoothing ρE in the DCO. Preservation of iso-enthalpic flows can be recovered simply by altering the DCO to operate on ρH instead of ρE.[16] Thus, we modify the DCO and compute it as

$$\boldsymbol{I}_{DCO} = \int_\Omega \hat{\nu} \frac{\partial \boldsymbol{W}}{\partial x_i} \cdot g^{ij} \frac{\partial \hat{\boldsymbol{U}}}{\partial x_j} \, d\Omega \tag{38}$$

where $\hat{\nu}$ is the continuous, nodal field determined by our L_2 projection, and

$$\hat{\boldsymbol{U}} = (\rho, \rho u_j, \rho H)^t \tag{39}$$

4.4. Summary

We now rewrite the weak form (28) with notation to emphasize the implementation details presented in this section. Find the approximate discrete solution $\boldsymbol{U}_h \simeq \boldsymbol{U}$ such that

$$\begin{aligned} \int_\Omega \boldsymbol{W} \cdot \left(\frac{\partial \boldsymbol{U}_h}{\partial t} - \boldsymbol{\mathcal{F}} \right) \, d\Omega + \int_\Omega \frac{\partial \boldsymbol{W}}{\partial x_i} \cdot (\boldsymbol{G}_i - \boldsymbol{F}_i) \, d\Omega + \int_{\partial\Omega} \boldsymbol{W} \cdot (\boldsymbol{F}_i - \boldsymbol{G}_i) \, \hat{n}_i \, d\Gamma + \\ \int_\Omega \frac{\partial \boldsymbol{W}}{\partial x_i} \cdot \boldsymbol{A}_i^l \boldsymbol{\tau}_s^l (\boldsymbol{\mathcal{L}} \boldsymbol{U}_h - \boldsymbol{\mathcal{F}}) \, d\Omega + \int_\Omega \hat{\nu} \frac{\partial \boldsymbol{W}}{\partial x_i} \cdot g^{ij} \frac{\partial \hat{\boldsymbol{U}}_h}{\partial x_j} \, d\Omega = \mathbf{0}, \end{aligned} \tag{40}$$

where the superscript $()^l$ indicates the quantity is to be evaluated at the node l that is associated with the test function $\boldsymbol{W}$, $\hat{\nu}$ is the interpolant of our nodally reconstructed function, and $\hat{\boldsymbol{U}}_h$ is given by (39).

5. Nonlinear Solution

Equation (40) is a semidiscrete representation of the chemically reacting Navier-Stokes equations. The authors have implemented for first order backward Euler, second order trapezoid rule, and second order backward difference formula time discretizations. Because we are currently interested in time-iterative to steady-state solutions, all of the results presented in the present work were obtained using the backward Euler scheme. We then form the fully discrete nonlinear resiudual

$$\begin{aligned} \boldsymbol{r} = \int_\Omega \boldsymbol{W} \cdot \left(\frac{\partial \boldsymbol{U}_h}{\partial t} - \boldsymbol{\mathcal{F}} \right) \, d\Omega + \int_\Omega \frac{\partial \boldsymbol{W}}{\partial x_i} \cdot (\boldsymbol{G}_i - \boldsymbol{F}_i) \, d\Omega + \int_{\partial\Omega} \boldsymbol{W} \cdot (\boldsymbol{F}_i - \boldsymbol{G}_i) \, \hat{n}_i \, d\Gamma + \\ \int_\Omega \frac{\partial \boldsymbol{W}}{\partial x_i} \cdot \boldsymbol{A}_i^l \boldsymbol{\tau}_s^l (\boldsymbol{\mathcal{L}} \boldsymbol{U}_h - \boldsymbol{\mathcal{F}}) \, d\Omega + \int_\Omega \hat{\nu} \frac{\partial \boldsymbol{W}}{\partial x_i} \cdot g^{ij} \frac{\partial \hat{\boldsymbol{U}}_h}{\partial x_j} \, d\Omega \end{aligned} \tag{41}$$

and use Newton's method to solve for $\boldsymbol{U}$ at each timestep. More specifically, we form $\boldsymbol{J} \equiv \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{U}}$ and solve

$$\boldsymbol{J} \Delta \boldsymbol{U} = -r \tag{42}$$

at each timestep. The Jacobian is calculated from each expression in (41) using a combination of analytic sensitivities, finite differences, and forward automatic differentiation. We neglect the sensitivites to $\tau_{\mathbf{s}}^{\mathbf{l}}$ and to $\hat{\nu}$. Each linear system is solved using GMRES with incomplete LU factorization as a preconditioner. We have found that including the sensitivities to the flux Jacobian matrices is important to achieving second-order convergence. In summary, our solution strategy is

```
given $\boldsymbol{U}(t = 0)$
save $\boldsymbol{r}_0 = \boldsymbol{r}(\boldsymbol{U}(t = 0)$
**for** $n = 0$ ; $n < n_{max}$; $n + +$ **do**
```

```
  $\boldsymbol{U}^0 = \boldsymbol{U}(t = t_n)$
  for $k = 0$ ; $k < k_{max}$; $k + +$ do
    solve the diagonal systems (35) and (37)
    solve $\boldsymbol{J}^k \Delta \boldsymbol{U}^{k+1} = -\boldsymbol{r}^k$
    $\boldsymbol{U}^{k+1} = \boldsymbol{U}^k + \Delta \boldsymbol{U}^{k+1}$
    if $\boldsymbol{r}^{k+1}/\boldsymbol{r}^0 < \epsilon_k$ then
      break
    end if
  end for
  $\boldsymbol{U}(t = t_{n+1}) = \boldsymbol{U}^{k+1}$
  if $\boldsymbol{r}^{k+1}/\boldsymbol{r}_0) < \epsilon_n$ then
    quit
  end if
end for
```

For the results presented in the present work, $\epsilon_k = 0.1$ and $\epsilon_n = 1e - 10$. The authors have also implemented a Newton-Krylov scheme, wherein the matrix $\boldsymbol{J}$ is never explicitly formed.

6. Results

Both inviscid and viscous test cases have been performed. An inviscid solution of dissociating nitrogen over a 1 m diameter circular cylinder has been compared to results from the SACCARA[19] finite volume code. Free stream conditions are, in SI units, $\rho_\infty = 0.000121704$, $u_\infty = 6155.63$, $T_\infty = 227.98$. This corresponds to approximately Mach 20. Although the simulation is nitrogen only, the free stream temperature and density correspond to the same values for air in the standard atmosphere at approximately 66.5 km altitude. Figure 1 shows the convergence history of the inviscid simulation. Both time step size and initial nonlinear residual norms are shown as a function of time step. The time step size starts at 1×10^{-9} s and is ramped adaptively as the simulation progresses. As the time step exceeds 1×10^{-4} s, the convergence of the solution accelerates. The jagged pattern in the timestep history is caused by occasional halving of the timestep by the adaptive algorithm. This happens whenever the nonlinear norm reduction tolerance for a given time step (0.1) is not reached in the maximum number of allowable nonlinear steps (10). Both of these parameters are user-controlled.

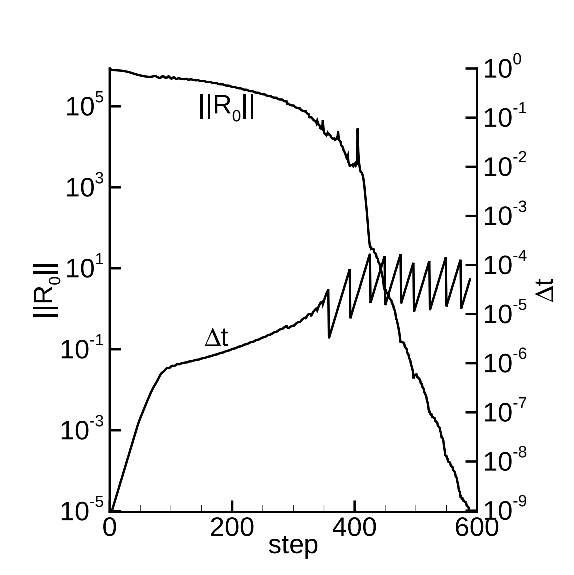

Figure 1. Initial nonlinear residual norm and timestep.

Figure 2 shows the pressure along the stagnation line for both the Aria (SUPG-stabilized finite element) and SACCARA (structured, cell-centered finite volume) solutions. As can be seen from the solutions, the jump conditions across the shock match exactly, but the shock locations are not identical. Although no "correct" shock location is

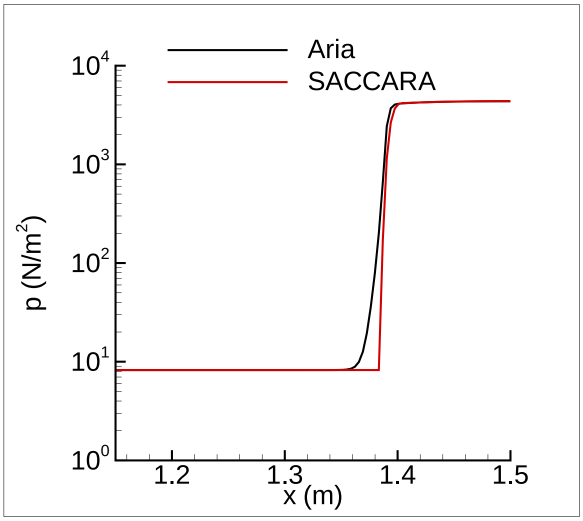

Figure 2. Pressure along stagnation line for both codes.

known for this dissociating case, the two codes converge to the same shock standoff distance with mesh refinement (not shown), although the Aria solutions converge from the front, and the SACCARA solutions converge from the back. The same trend is observed for ideal gas solutions,[15] where the "correct" shock standoff distance can be approximated via an empirical relationship. For the ideal gas case, the error in shock standoff distance is slightly lower for the SACCARA solutions, at the same mesh resolution. This is because the DCO in the SUPG-stabilized scheme is slightly more dissipative in the vicinity of the shock than the flux limiter which is activated in the structured, cell-centered finite-volume scheme. Also, the dissipation in the DCO tends to be biased toward the upstream side of the shock, whereas the dissipation in the flux limiter tends to be biased toward the downstream side of the shock. Figure 3 shows the mass fraction of monatomic nitrogen for both Aria and SACCARA. The figure shows good agreement up to the stagnation point, where the SACCARA solution trends lower into the surface. The SACCARA solution has slightly less dissociation since its shock is farther back, and thus, the integration time of the finite-rate reaction is slightly less for any given point in space. The more significant difference between the two solutions at the stagnation point is caused by SACCARA's zeroth-order extrapolation into the ghost cell at this boundary. Since Aria enforces no such artificial BC at this point, its solution demonstrates the expected behavior.

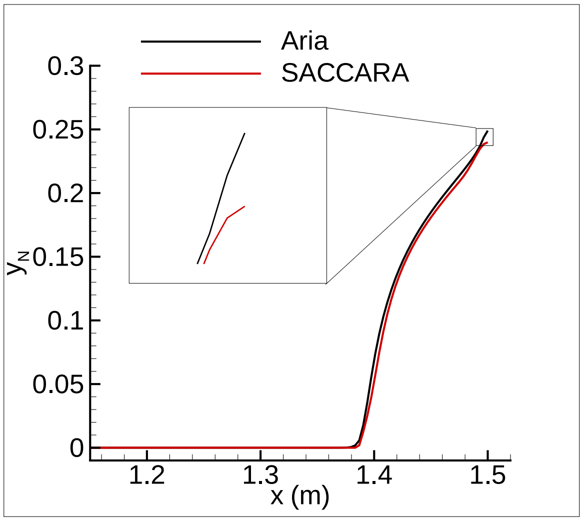

Figure 3. Mass fraction of monatomic nitrogen along stagnation line for both codes.

A viscous solution has also been produced for the same flow conditions with an isothermal wall at 10,000 K and super catalycity (*i.e.*, all of the dissociated nitrogen recombines at the surface). The high wall temperature was chosen to ensure that the catalytic behavior of the wall was correctly being enforced, since lower wall temperatures would cause significant recombination even without wall catalycity. Figure 4 shows the density of monatomic nitrogen and temperature for this case.

Figure 5 shows a viscous solution at the same free-stream conditions around the IRV-2 nosetip. This fluid-only simulation corresponds to a coupled ablation simulation of the carbon-carbon nosetip.[20]

Figure 6 shows the mass fractions of N, O, and NO from both the FIN-S (SUPG stabilized finite element) and DPLR (structured, cell-centered finite-volume) codes for Mach 15 flow of 5-species air over a circular cylinder using a two-temperature model. Figure 7 shows the translational temperature for the same case. The same trends are observed as for the nitrogen-only comparison between Aria and SACCARA: the shock is slightly farther forward for SUPG than finite-volume, due to differences in the distribution of numerical dissipation near the shock, but the jump conditions match.

7. Summary

We have proposed an SUPG algorithm for solving chemically reacting hypersonic flows, and impleented it in two separate codes. Results for both a one temperature and a two temperature model have been shown to achieve the proper shock jump conditions and locations. Because the method is finite-element based, it is well-suited for application to unstructured meshes. Current and future work involves extensions to one and two equation turbulence models, and hetergeneous and h-adaptive meshes. Work is also ongoing to simulate hypersonic reentry flows with conjugate heat transfer.

Appendix

Many variants of stabilized finite element methods based on SUPG with a shock capturing operator have been published for the case of an ideal gas over the last two decades. A key idea for these methods is a transformation of dependent variables from the conservation variables $\boldsymbol{U}$ to the so-called entropy variables $\boldsymbol{V}$ (e.g. see Shakib et al.[11]). The entropy variables are defined by taking the gradient of a scalar entropy function with respect to the conservation variables $\boldsymbol{U}$. This transformation is defined by the chain rule

$$\delta \boldsymbol{U} = \boldsymbol{A}_0 \delta \boldsymbol{V}$$

where

$$\boldsymbol{A}_0 = \frac{\partial \boldsymbol{U}}{\partial \boldsymbol{V}}$$

Since $\boldsymbol{A}_0$ is the Hessian of an entropy function, it is symmetric and positive semi-definite. The matrix $\boldsymbol{A}_0$ is referred to as the symmetrizer because if the convective and viscous flux Jacobian matrices are post-multiplied by $\boldsymbol{A}_0$, then they become symmetric. In fact, this symmetric form of the equations (in terms of the entropy variables) is the starting point for the SUPG method. For convenience, after the method is derived, it is transformed back to conservation variables, which is why the inverse symmetrizer $\boldsymbol{A}_0^{-1}$ appears in (27).

In this Appendix, we present the flux Jacobian matrices and inverse symmetrizer for a mixture of thermally perfect gases in thermochemical nonequilibrium for the case where the translational and vibrational temperatures are identical. In compact notation, the flux Jacobian matrix in Cartesian coordinate direction i may be written as

$$\boldsymbol{A}_{ijk} = \begin{bmatrix} u_i\left(1 - \frac{\rho_1}{\rho}\right) & \dots & -u_i \frac{\rho_1}{\rho} & \frac{\rho_1}{\rho}\delta_{ij} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ -u_i \frac{\rho_{\mathrm{S}}}{\rho} & \dots & u_i\left(1 - \frac{\rho_{\mathrm{S}}}{\rho}\right) & \frac{\rho_{\mathrm{S}}}{\rho}\delta_{ij} & 0 \\ -u_i u_j + \alpha_1 \delta_{ij} & \dots & -u_i u_j + \alpha_s \delta_{ij} & \beta_{ijk} & \kappa \delta_{ij} \\ u_i\left(\alpha_1 - H\right) & \dots & u_i\left(\alpha_{\mathrm{S}} - H\right) & H\delta_{ij} - \kappa \rho u_i u_j & \left(\kappa + 1\right) u_i \end{bmatrix} \tag{43}$$

The index j denotes the row of the momentum equation, and k indicates the momentum component with respect to which the derivative of the flux in direction i is being taken. In other words, the index j expands the rows of the flux Jacobian $\boldsymbol{A}_i$ according to the spatial dimension, and the index k expands the columns. These flux jacobian matrices

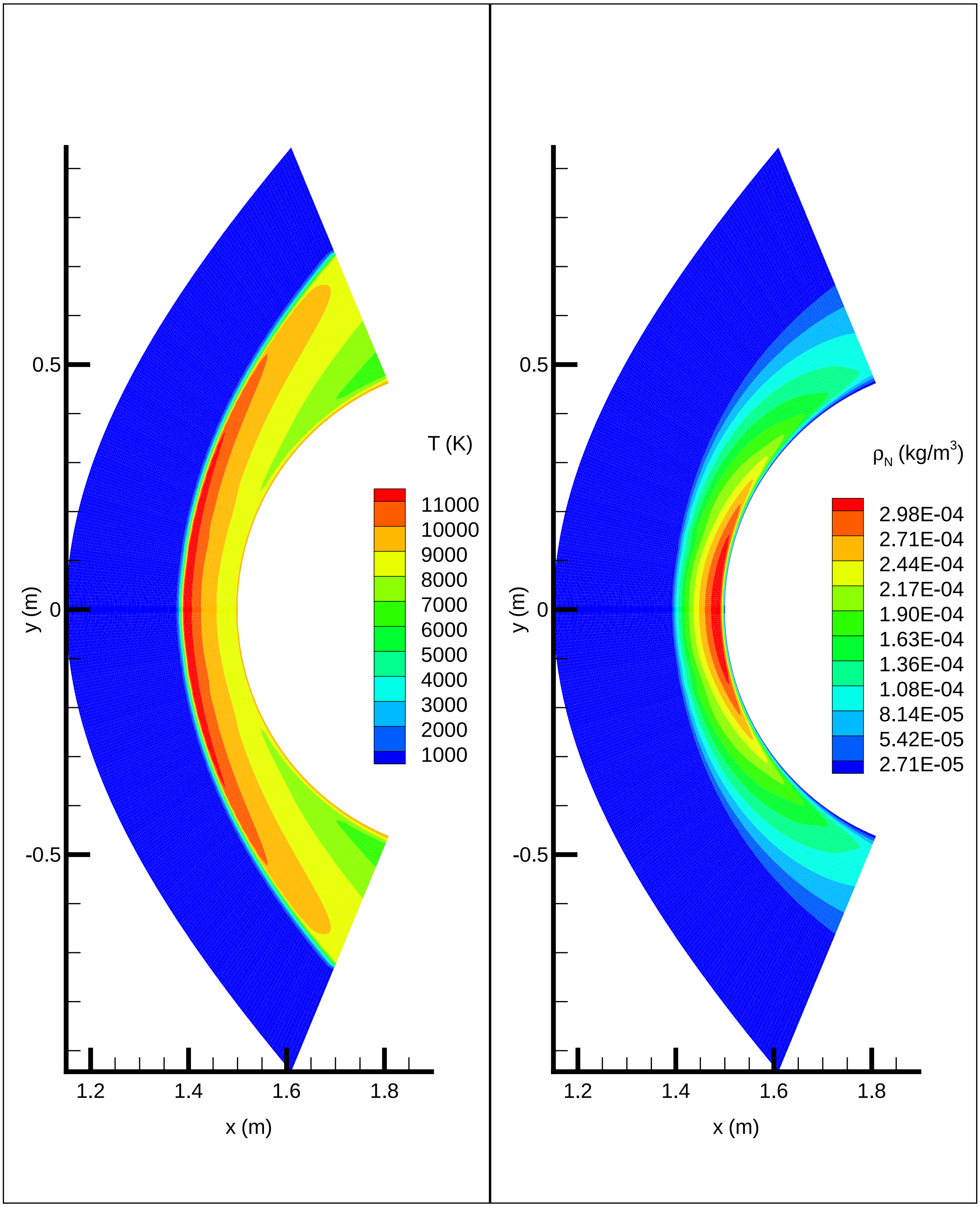

Figure 4. Density of monatomic nitrogen and temperature for the viscous case.

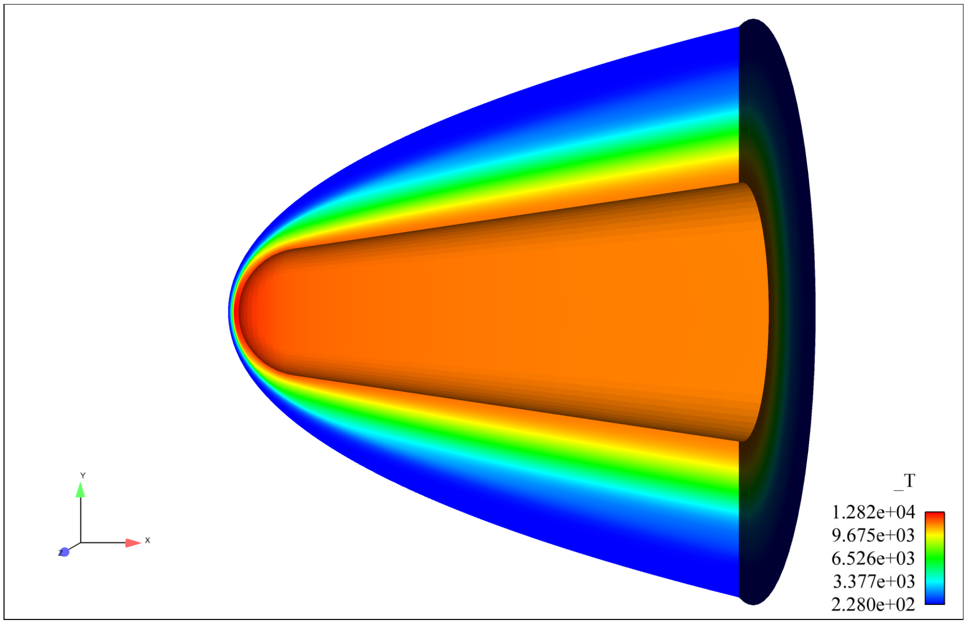

Figure 5. Temperature (K) contours on cut planes in IRV-2 nosetip simulation.

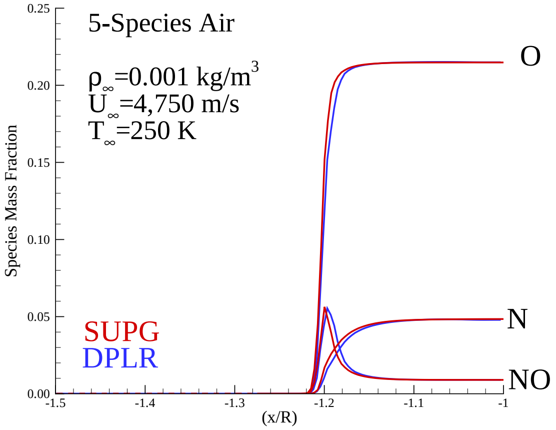

Figure 6. Mass fractions on cylinder case with 5-species air.

are discussed in Liu and Vinokur.[21] In (43), $u^2 = u_k u_k$, $\beta_{ijk} = u_i \delta_{jk} + u_j \delta_{ik} - \kappa u_k \delta_{ij}$, $\alpha_s = \frac{1}{2}\kappa u^2 + \chi_s$,

$$\kappa = \frac{1}{\rho c_v} \sum_{s=1}^{S} \rho_s R_s \tag{44}$$

is the partial derivative of pressure with respect to internal energy at constant mixture density, and χ_s is the partial derivative of pressure with respect to the species density at constant temperature,

$$\chi_s = R_s T - \kappa e_s \tag{45}$$

Finally, R_s and e_s are the gas constant and internal energy of species s, respectively.

Our form of the matrix $\boldsymbol{A}_0^{-1}$ for the case of thermal equilibrium was derived by simplifying the form published by

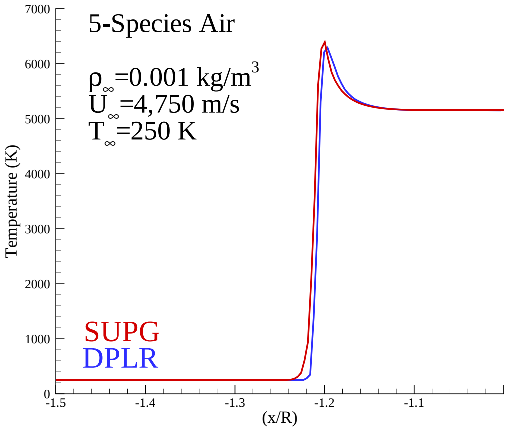

Figure 7. Translational temperature on cylinder case with 5-species air.

Chalot et al,[4] which includes a second temperature for the vibrational energy. This simplified form may be written as

$$\boldsymbol{A}_0^{-1} = \frac{1}{\mathcal{E}} \left[\begin{array}{ccccccc} \bar{a}_{11} & \ldots & \bar{a}_{1\mathrm{s}} & \bar{b}_1 u_1 & \bar{b}_1 u_2 & \bar{b}_1 u_3 & \bar{d}_1 \\ & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & & \bar{a}_{\mathrm{ss}} & \bar{b}_{\mathrm{s}} u_1 & \bar{b}_{\mathrm{s}} u_2 & \bar{b}_{\mathrm{s}} u_3 & \bar{d}_{\mathrm{s}} \\ & & & u_1^2 + c_v T & u_1 u_2 & u_1 u_3 & -u_1 \\ & & & & u_2^2 + c_v T & u_2 u_3 & -u_2 \\ \text{Symmetric} & & & & & u_3^2 + c_v T & -u_3 \\ & & & & & & 1 \end{array} \right] \tag{46}$$

where

$$\mathcal{E} = \rho c_v T^2 \tag{47}$$

$$\begin{aligned} \bar{a}_{sr} &= \left(c_{vs} T + h_s^0 - \frac{u^2}{2} \right) \left(c_{vr} T + h_r^0 - \frac{u^2}{2} \right) + e_s^{\text{vib}} e_r^{\text{vib}} \frac{c_v}{c_v^{\text{vib}}} + \\ & \quad c_v T u^2 + \rho c_v T^2 \left(\sum_{k=1}^{\mathrm{s}} \frac{\rho_k}{R_k} \right)^{-1} \delta_{sr} \end{aligned} \tag{48}$$

$$\bar{b}_s = c_{vs} T + h_s^0 - \frac{u^2}{2} - c_v T \tag{49}$$

$$\bar{c}_s = c_{vs} T + h_s^0 - \frac{u^2}{2} - e_s^{\text{vib}} \frac{c_v}{c_v^{\text{vib}}} \tag{50}$$

$$\bar{d}_s = -\left(c_{vs} T + h_s^0 - \frac{u^2}{2} \right) \tag{51}$$

In the above relations, c_{vs} is the specific heat at constant volume of species s, and e_s^{vib} is the vibrational part of the internal energy of species s. For the case of thermal equilibrium (*i.e.*, single temperature model), terms explicitly referencing the vibrational energy are not included.

References

[1] B.S. Kirk and G.F. Carey. Development and validation of a SUPG finite element scheme for the compressible Navier-Stokes equations using a modified inviscid flux discretization. *International Journal for Numerical Methods in Fluids*, 57:265–293, 2008.

[2] I.G. Currie. *Fundamental Mechanics of Fluids*. McGraw-Hill, 1974.

[3] B. J. McBride, M. J. Zehe, and S. Gordon. NASA Glenn coefficients for calculating thermodynamic properties of individual species. Technical Report NASA/TP—2002-211556, National Aeronautics and Space Administration, 2002.

[4] F. Chalot, T.J.R. Hughes, and F. Shakib. Symmetrization of conservation laws with entropy for high-temperature hypersonic computations. *Computing Systems in Engineering*, 1:495–521, 1990.

[5] F. Chalot and T.J.R. Hughes. A consistent equilibrium chemistry algorithm for hypersonic flows. *Computer Methods in Applied Mechanics and Engineering*, 112:25–40, 1994.

[6] T.J.R. Hughes, L.P. Franca, and M. Mallet. A new finite element formulation for computational fluid dynamics: I. Symmetric forms of the compressible Euler and Navier-Stokes equations and the second law of thermodynamics. *Computer Methods in Applied Mechanics and Engineering*, 54:223–234, 1986.

[7] Thomas J.R. Hughes, Michel Mallet, and Akira Mizukami. A new finite element formulation for computational fluid dynamics: II. Beyond SUPG. *Computer Methods in Applied Mechanics and Engineering*, 54:341–355, 1986.

[8] Thomas J.R. Hughes and Michel Mallet. A new finite element formulation for computational fluid dynamics: III. The generalized streamline operator for multidimensional advective-diffusive systems. *Computer Methods in Applied Mechanics and Engineering*, 58:305–328, 1986.

[9] Thomas J.R. Hughes and Michel Mallet. A new finite element formulation for computational fluid dynamics: IV. A discontinuity-capturing operator for multidimensional advective-diffusive systems. *Computer Methods in Applied Mechanics and Engineering*, 58:329–336, 1986.

[10] Thomas J.R. Hughes, Leopoldo P. Franca, and Michel Mallet. A new finite element formulation for computational fluid dynamics: VI. Convergence analysis of the generalized SUPG formulation for linear time-dependent multidimensional advective-diffusive systems. *Computer Methods in Applied Mechanics and Engineering*, 63:97–112, 1987.

[11] Farzin Shakib, Thomas J.R. Hughes, and Zdeněk Johan. A new finite element formulation for computational fluid dynamics: X. The compressible Euler and Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 89:141–219, 1991.

[12] Benjamin S. Kirk. *Adaptive Finite Element Simulation of Flow and Transport Applications on Parallel Computers*. PhD thesis, The University of Texas at Austin, May 2007.

[13] Guillermo Hauke and Thomas J.R. Hughes. A comparitive study of different sets of variables for solving compressible and incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 153:1–44, 1998.

[14] Tayfun E. Tezduyar and Masayoshi Senga. SUPG finite element computation of inviscid supersonic flows with YZβ shock-capturing. *Computers and Fluids*, 36:147–159, 2007.

[15] Benjamin S. Kirk, Steven W. Bova, and Ryan B. Bond. The influence of stabilization parameters in the SUPG finite element method for hypersonic flows. AIAA Paper 2010-1183, January 2010.

[16] Benjamin S. Kirk. Adiabatic shock capturing in perfect gas hypersonic flows. *International Journal for Numerical Methods in Fluids*, In press, 2009.

[17] Garrett E. Barter and David L. Darmofal. Shock capturing with higher-order, PDE-based artificial viscosity. In *18th AIAA Computational Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics, June 2007.

[18] A. Jameson. Analysis and design of numerical schemes for gas dynamics. 2: Artificial diffusion and discrete shock structure. Technical Report NASA CR-196476, National Aeronautics and Space Administration, August 1994.

[19] C. C. Wong *et al.* PINCA: A scalable parallel program for compressible gas dynamics with non-equilibrium chemistry. Technical Report SAND 94-2436, Sandia National Laboratories, 1995.

[20] Steven W. Bova and Ryan B. Bond. Report of the FY09 ASC level 2 ablation/reentry modelling capability milestone. SNL Internal memo SAND 2009-5899-P, September 2009.

[21] Yen Liu and Marcel Vinokur. Nonequilibrium flow computations. I. an analysis of numerical formulations of conservation laws. *Journal of computational physics*, 83:373–397, 1989.