SANDIA REPORT
SAND2007-4216
Unlimited Release
Printed July 2007

CADS
Cantera Aerosol Dynamics Simulator

Harry K. Moffat

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

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

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@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd.
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND 2007-4216
Unlimited Release
Printed July 2007

CADS
Cantera Aerosol Dynamics Simulator

Harry K. Moffat
Multiphase and Nanoscale Transport Division
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-0836

Abstract

This manual describes a library for aerosol kinetics and transport, called `CADS` (Cantera Aerosol Dynamics Simulator), which employs a section-based approach for describing the particle size distributions. `CADS` is based upon Cantera, a set of C++ libraries and applications that handles gas phase species transport and reactions. The method uses a discontinuous Galerkin formulation to represent the particle distributions within each section and to solve for changes to the aerosol particle distributions due to condensation, coagulation, and nucleation processes. `CADS` conserves particles, elements, and total enthalpy up to numerical round-off error, in all of its formulations. Both 0-D time dependent and 1-D steady state applications (an opposing-flow flame application) have been developed with `CADS`, with the initial emphasis on developing fundamental mechanisms for soot formation within fires. This report also describes the 0-D application, `TDcads`, which models a time-dependent perfectly stirred reactor.

Acknowledgment

CADS was developed under Engineering Science Research Foundation Funding. The author would like to thank John Hewson, John Brockmann, Sheldon Tiesdon, Prof. Adel Sarofim, Prof. Phil Smith, Zhiwhe Yang, Dave Lignell and many others for the helpful comments during the process of developing the code and reviewing this manual.

Table of Contents

1. Introduction

Cantera Aerosol Dynamics Simulator (CADS) is a tool for developing predictive phenomenological models for aerosol formation incorporating detailed kinetics mechanisms that yield the concentrations, particle sizes, and temperature, in reactive flow environments. CADS is a generalized library able to address particle formation and transport processes, including particle coagulation, gas phase reactions, surface chemistry (growth and oxidation), transport by Brownian diffusion, thermophoresis, and diffusiophoresis, and particle nucleation from molecular species.

CADS is based on a sectional description of particle sizes. The continuous distribution of particle sizes, which may typically span from the sub-nanometer regime up to 100 microns is discretized into individual bins. Each bin may have multiple degrees of freedom, with 1or 2 degrees of freedom being used to describe the particle-size distribution within a bin. Additional degrees of freedom within each bin are used to describe the composition of a multicomponent particle phase, assuming the full mixture state approximation for the particles, allowing also for the calculation of intra-particle condensed phase chemistry.

As a library, CADS may be incorporated into computational fluid dynamics (CFD) codes to handle particle source terms and gas source terms due to particle-related reactions and to provide transport properties for particles. The libraries' application programming interface (API) includes general calls for particle and gas-phase species source terms due to nucleation, coagulation, and condensation reactions. Particle source terms are on a per-sectional-bin basis. Transport properties for particles, both diffusion rates and particle terminal velocities which are influenced by external forces acting on the particles, are obtained on a per section-bin basis. Applications based on CADS handle the solution of conservation reactions describing the transport and reaction of particles in zero or multiple dimensions by calling CADS source-term and transport-parameters routines.

Within the CADS package is a time-dependent batch reactor and perfectly-stirred reactor (PSR) simulator named TDcads. This application has been very useful to address algorithm issues within the CADS' aerosol modules, to verify mechanism phenomena, and to address particle phenomena that can be addressed in a 0D setting.

CADS is layered on top of Cantera [1], an object-oriented tool for solving chemical kinetics, thermodynamics, and transport processes in reacting flows. Cantera's main role has been to solve applications involving reacting laminar gas flows, including combustion, chemical vapor deposition, and solid fuel-cell systems. Cantera has application codes for PSR's, 1D reacting flow simulations, and equilibrium solvers. It has be used as a third-party library in external reacting-

flow simulation codes, such as FLUENT, using fortran, C++, python, or matlab interfaces, to evaluate properties and chemical source terms that appear in the application's governing equations. Therefore, CADS may be considered to be an extension of Cantera's constitutive modeling package paradigm to encompass constitutive modeling of particle interactions.

The initial target application has been to describe soot formation in laminar flames. Soot dominates radiative heat transfer in a fire. Soot formation, volume fraction, particle size, temperature, and radiative properties all need to be modeled in order to calculate heat transfer within a fire. Current soot models are highly empirical, fuel specific, and without particle size description and temperature, i.e., inadequate for our needs. Soot morphology strongly suggests a specific, nearly universal, soot formation sequence. Molecular transport mechanisms play an important role in making this sequence happen and in determining the location and temperature of radiating hot soot as well as the size of the primary particles. Modeling must be used to come up with an explanation for reconciling the observations on soot morphology with those on soot formation. It appears that molecular transport mechanisms such as thermophoresis are important in locating the soot particles relative to the flame zone and consequently determining the soot particle temperature which determined the radiative energy transfer. Chemistry, also dependent on temperature and particle age, determines the radiative properties of the soot.

A sub-grid model for soot formation, that may be used for applications on the engineering scale, that yields both the concentration and temperature, must include this physics. Therefore, a computational framework is needed to develop the specific models to allow us to calculate the position, temperature, and radiative properties of soot particles in flames. It is hoped that the general open-source tool, CADS, will help the fire community to develop these detailed mechanisms for soot formation, that may then be used for generating sub-grid soot models suitable for understanding heat fluxes in fires. Additional promising applications for CADS involve the modeling of nano particle production, micro-electro-mechanical systems (MEMS), chemical vapor deposition (CVD), and microcontamination processes.

1.1 Equations that CADS solves

The formulation for CADS is based around providing constitutive models for all of the terms in the multidimensional aerosol general dynamics equation [2] for the particle concentration distribution function, n. $n(\mu, t)$ is the concentration of particles at position $\boldsymbol{x}$ and time t with total monomer unit numbers between μ and $\mu + d\mu$.

$$\frac{d}{dt}n(\mu, t) + \nabla\bullet(\boldsymbol{v}n(\mu, t)) + \nabla\bullet(\boldsymbol{c}n(\mu, t)) + \frac{\partial}{\partial\mu}(G(\mu, t)n(\mu, t)) = \nabla\bullet(D\nabla n(\mu, t)) + \frac{1}{2}\int_0^{\mu}\beta(\tilde{\mu}, \mu - \tilde{\mu})n(\tilde{\mu})n(\mu - \tilde{\mu})d\tilde{\mu} - \int_0^{\infty}\beta(\mu, \tilde{\mu})n(\tilde{\mu})n(\mu)d\tilde{\mu} + S_{nucl}^{part} \quad (1.1)$$

$n = n(\mu, t)$ is a function of the number of monomer units (i.e., molecules) in the particle, μ, time, and the spatial dimension. $G(\mu, t)$ is the net growth rate of particles of size μ due to interactions with the gas phase; it may be negatively or positively valued. The resulting flux of particles, $G(\mu, t)n(\mu, t)$, in the particle coordinate, μ, is called the particle current and the corresponding term in Eqn. (1.1) is called the condensation operator. The two integral expressions in Eqn. (1.1), the continuum formulation of the Smoluchowski expression, constitutes the coagulation operator. $\beta(\mu, \tilde{\mu})$ is the coagulation kernel, which is the cross-section for coagulation for particles of size μ and $\tilde{\mu}$. D is the particle diffusion coefficient, a strong function of the particle size, for small particles. $\boldsymbol{c}$ is the deviation of the particle terminal velocity (i.e., the assumption that the particles are in inertial equilibrium is implicitly made here) from the mass averaged velocity of the gas medium, $\boldsymbol{v}$, due to the influence of external forces such as thermophoresis, diffusiophoresis, and gravity. S_{nucl}^{part} is the source term for particle creation due to nucleation from molecular species. In CADS, nucleation also refers to cases wherein particles are destroyed due to physical processes such as the oxidation of carbon.

The first step in solving the integro-differential equation of Eqn. (1.1) is to discretize the particle distribution coordinate, μ, into distinct sectional bins. Then, the solution of Eqn. (1.1) may be undertaken on a "per-bin" basis, with the μ value acting substantially like a distinct coordinate dimension.

There has been an extensive treatment of section models in the literature. In early work, Gelbard and Seinfeld [3] solved analytically the equation for coagulation and growth of a multicomponent aerosol for two components and simplified coagulation rates and growth laws. The sectional solution approach to the multicomponent continuous general dynamic equation has been developed by Gelbard and Seinfeld [4]; it is this work and the resulting MAEROS application [5] that is forerunner to the model presented here. Recent work on combustion systems have included an approach where sectional bins are treated as an extension of individual molecules [6].

Early on in the program, it was realized that conservation properties for both particle mass/composition and particle number were important to the overall goal of obtaining mechanistic understanding of particle inception and growth in fires. There had been some work previously on "2 component" sectional models that achieve this, through keeping track of both particle number and particle mass within a section [7, 8]. This approach is currently the favorite for global climate and marine environment simulations. However, from previous work on advection dominated systems,

a promising approach using discontinuous Galerkin approximation to the condensation operator has been adopted within CADS. This approach has proven to be extremely accurate both for the calculation of the condensation and coagulation operators, because of the inherent low numerical diffusion properties of the discontinuous Galerkin approximation and due to the fact in contrast with [7, 8] the particle concentrations within a section are distributed throughout the section. Difficulties associated with maintaining positivity of the particle distribution within each section have been encountered and largely overcome.

The condensation kernel uses a discontinuous Galerkin method to calculate the transfer rates of particles along the particle number axis due to the condensation process. Growth or particle etching rates are determined on a per surface area basis. Spherical particles are currently assumed. Surface growth rates for particles are determined in a separate kernel that has the ability to treat the surface of particles as a separate phase consisting of surface species, with surface site fractions describing their concentrations. Surface growth processes are then described via mass-action surface reactions involving gas phase, surface phase, and bulk phase species. The particle surface growth kernel uses an internal implicit solver, so that surface site fractions may be implicitly solved for during a residual calculation, so that they don't explicitly appear in the solution vector.

The coagulation kernels are precalculated using 2D Romberg integrations and stored for later use. The dependence of the coagulation kernel on temperature, composition, and pressure are factored in by calculating a ratio of the present value of the coagulation kernel to a stored value of the coagulation kernel used in the precalculation. No limitations on sectional bin discretizations are imposed by the coagulation kernel. The software does not support coagulation kernels for charged particles at this time. The concept of a fractal dimension that determines the volume to surface area of particles and affects the coagulation kernel is not fully implemented within the library, yet.

S_{nucl}^{part} in Eqn. (1.1) refers to all phenomena that involving the creation (or destruction) of particles from interactions with gas phase species. The modeling approach used here treats nucleation as a special case of agglomeration. In this treatment, the evolution of the particle size distribution by agglomeration is explicitly included in chemical reaction-like mechanisms that transfer material from the molecular species to particles. This class of nucleation phenomena is sometimes called rapid nucleation. These routines maybe incorporated into flow and transport codes in the same manner that the Cantera subroutines are used to calculate gas phase chemistry. This treatment facilitates the inclusion of particle formation and growth in existing codes that now perform chemistry calculations. The treatment in CADS can optionally handle reversible nucleation reactions. Just as their molecular counterparts, particles have Gibbs free energies of formation assigned to them. These values normally come from treating the particles as ideal solid solutions of monomer units which have standard state thermodynamic properties. However, individual particles in-

volved with the nucleation mechanism may have specific thermodynamic properties assigned to them, which overwrite the default values. With the Gibbs free energies of particles specified, reversible nucleation reactions may be formulated through the principle of microscopic reversibility. The idea of reversible nucleation reactions has grown out of the concept, from soot nucleation that since the precursor species to particles in flames are small PAH species and since particles in the lowest bin sections are larger PAH species, then the relative abundance of small to large PAH species in the active region of a flame must involve thermodynamic factors to a great extent.

Classical nucleation theory (and the associated parameterization of the effect of surface tension on the Gibbs free energy of small particles) isn't in `CADS` yet. In classical nucleation theory, monomers have a concentration slightly above their saturation limit of the gas. The monomers cluster together through a reversible coagulation process, mitigated by the effects of surface tension which affects the stability of the cluster. Surface tension forces generally increase as the particle diameter decreases, leading to smaller particles having a larger effective Gibbs free energy. This Kelvin effect leads to a minimum in the distribution of the cluster distribution. The minimum in the distribution is called the critical cluster size. The particle flux through the minimum cluster size is associated with the classical particle nucleation rate.

At the start of the program, it was thought that special attention to the cluster-size regime was going to be paid via a model akin to Wu and Flagan [9], wherein a discrete description of the monomer and cluster interactions have been melded into a treatment of particles separated into sectional bins at higher monomer numbers. However, due to overall time constraints, this was dropped. Moreover, for the initial soot problem class, this may not have been necessary. Essentially, the small cluster regime for combustion mechanisms is dominated by the study of the chemistry and thermodynamics of large PAH species [10]. The equivalent of classical nucleation theory comes down to understanding the thermodynamics of large PAH molecules, perhaps using a variant of group additivity theory [11].

All kernel operations within `CADS` have "analytical" jacobian routines. However, some of these routines actually use numerical differencing schemes to obtain "quasi-analytical" jacobians for the individual operators. Because the particle number coordinate used in the sectional representation partially behaves like a dimensional coordinate, the calculation of these analytical jacobians is much faster than the calculation of purely numerical jacobians, by a factor of 10. The analytical jacobian formulation takes advantage of the interaction sparsity to reduce the operation count.

Within `CADS`, the assumption is made that particles travel at their terminal velocities. The terminal velocity may be different than the mass-averaged velocity of the gas phase, due to external forces. The relative difference between the terminal velocity and the mass-averaged gas velocity is equal

to $\boldsymbol{c}$ in Eqn. (1.1). External forces currently implemented within CADS are the thermophoretic and diffusiophoretic forces. Gravity as an external force, as well as the gravitational settling process as a particle sink) is currently not implemented, because the initial problem class didn't warrant its inclusion. This small inertial assumption creates a great deal of simplification with respect to the solution of the multidimensional particle transport equation. However, it limits the problem class that CADS may be applied to [12]. Routines for the evaluation of the particle diffusion coefficient, D, are also included within CADS. Only models for the Brownian diffusion coefficient are currently covered. In all cases of the evaluation of transport properties, interpolation formulas between the free-molecular regime and the continuum regime are used.

How does this method compare with the moments method? Sectional methods will not compete in speed and computational efficiency with moments methods, because they utilize many more degrees of freedom to represent the particle distributions than do moments methods [13]. However, they may be used to check the veracity of assumptions to the gross shape of the particle distribution that moments methods must innately make and to understand how these assumptions affect the answer. Detailed molecular modeling of the aerosol nucleation chemistry and the thermodynamics of particles/species in the boundary between the molecular species and the particle region is possible also with sectional methods but not with moments methods.

1.2 Overall Layout and Structure of the Input Files

CADS employs a block text input file as its main input. The left side of Figure 1.1 presents a much shortened layout for that block text file. Separate blocks are included in that file pertaining to the main categories of input. Some blocks are only associated with the TDcads application, while other blocks, such as the ADS Model Definition block and the Section Model Definition Block, are needed by all applications that use CADS.

Within the ADS Model Definition block are the names of several XML or CTI input files, which specify the several chemistry mechanisms involved with conducting a particle simulation. CTI is a human readable Cantera file format that get's preprocessed into Cantera's XML input file format. As CADS reads its main block text input file, it looks for the names of these auxiliary input files and reads their contents in during the initialization of the package. The main block of the input file therefore serves as a directory for needed file locations.

These XML or CTI input files are read by low-level Cantera routines, and serve to initialize the gas-phase chemistry mechanism, the gas-phase particle nucleation mechanism, the particle surface chemistry mechanism, and the bulk particle chemistry mechanism. The rough content of each of these files are listed in Figure 1.1. Details about each of these files, as well as details about

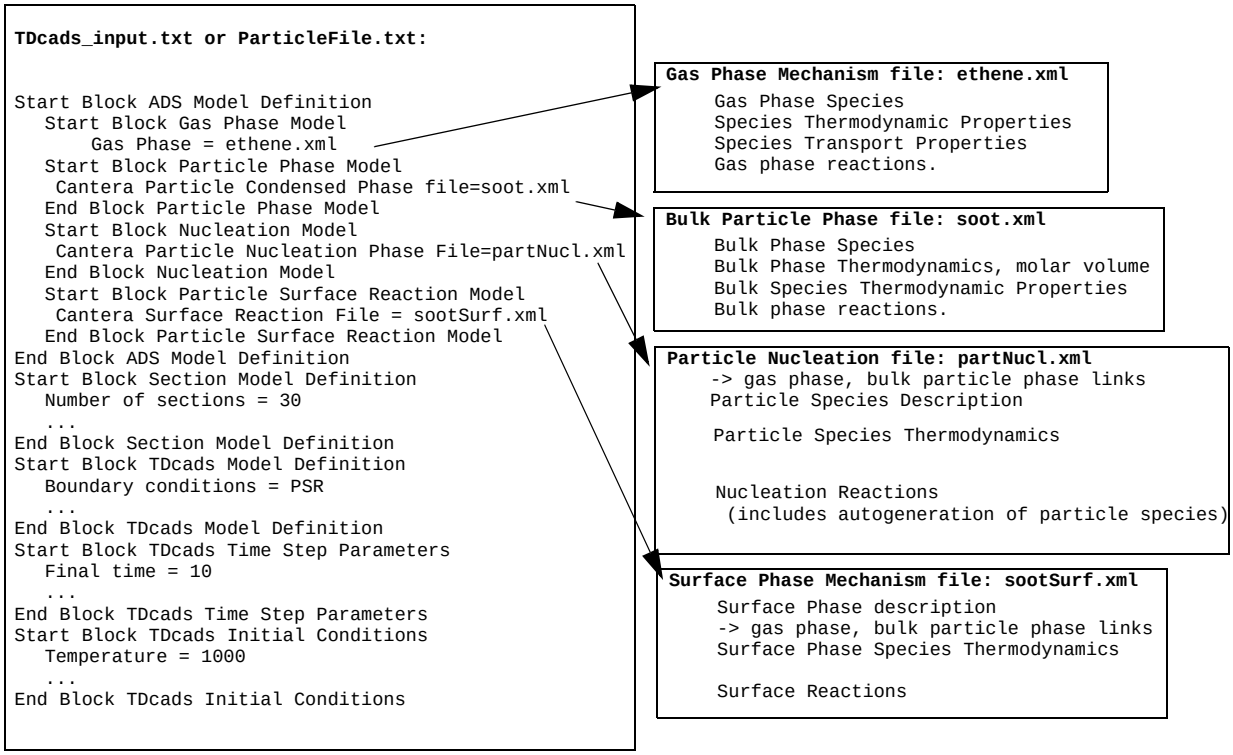

Figure 1.1 Layout of CADS and TDcads input files

the content and options in the main block text input file is presented in the pertinent chapters of this manual.

CADS shares with Cantera the additional paradigm of attempting to separate out the data from the solution method. This methodology has a strong tradition in the combustion field. For example the calculation of the Gibbs free energies of particles is a separate knowable fact that may be separated, parameterized, and stored in a data file; it may be correlated with the observation of particle number from a certain simulation, but its parameterization is separable. The rate that acetylene adds to radical benzene-like sites on a soot particle may be parameterized and stored in a data file; its value may be highly correlated with particle number in a sooting flame. However, given enough information its rate constant is separately determinable. The data for CADS are given in the XML files, while the choices for solution method is given in the block text input file.

Note, this same layout is used in multiple applications, because the block I/O routines are built into the CADS library routines. Right now there are 2 applications based on CADS. A zero-D code that emulates PSR's, with particle creation. The other application is based on an 1D opposing flow flame code. Both of these applications share the same CADS input file formats.

1.3 How To Compile and Use CADS

CADS is based on the open source software code Cantera to calculate reaction rates and evaluate thermodynamic quantities. Cantera has a Berkeley open source license, and is freely available for download from http://sourceforge.net. CADS will also be available on http://sourceforge.net, and has a Berkeley open source license associated with it.

To use CADS, Cantera and CADS must be downloaded from http://sourceforge.net. Autoconf scripts automate the installation of Cantera and CADS. In order to use CADS, Cantera must be compiled and installed. Then, CADS applications link against the installed Cantera libraries as a Cantera application. Further directions for installation of the software are included in the downloaded files.

Cantera has both a python and a matlab front end to it. Currently, CADS does not have these front ends. In order to use Cantera's CTI file format, Cantera must be installed with at least the "minimal python installation" option.

1.4 Outline of the Report

The report is broken up into chapters, each one focussing on a major physical process instantiated as a kernel operation within CADS. The first chapter is focussed on describing the sectional representation and numerical details associated with this representation. The following chapters concentrate on individual kernel options within CADS that comprise its functionality: condensation, coagulation, nucleation, particle surface growth (a distinct component operation of condensation), bulk chemistry, and transport properties. Then, the last two chapters describe the equations behind TDcads, the particle PSR/batch reactor simulator, and provide a couple of example problems using TDcads. Additional TDcads examples are provided in the downloadable source code.

2. Equation and Section Model Setup

In this chapter, some of the critical terminology for the section method is outlined. The independent variables are delineated. How the sectional bins are set up from the input file is described. Lastly, numerical issues related to all of the individual source terms are discussed.

2.1 Particles versus Clusters

The terms, clusters and particles, will refer to distinct concepts in this document. Particles, a general term, refers to any condensed phase object of any size in the particle phase. The term, cluster, refers to a subclass of particles, where the modeling package will follow them as either distinct particles or will bin them via their discrete sizes, while averaging their compositions to produce a single derived composition for a given discrete size. Currently, clusters are not implemented in the current version of the package, and represent a research opportunity.

2.2 Monomer Unit

The concept of the monomer unit (MU for short) is central to the technical description and implementation of the modeling package and its connection to the underlying gas-phase kinetics mechanism. A monomer unit represents the fundamental building block for units within the particle modeling package. There may be more than one type of building block within the modeling package, in which case particles will have varying concentrations of these monomer units as a function of particle size. A monomer unit is closely akin to the concept of a “molecule” in the gas package. Each type of MU is made up of a fixed number of elements and has a stoichiometry based on the number of elements. The underlying gas package dictates what elements are defined in the combined package. However, a monomer unit doesn’t necessarily have to be defined as a molecule in the gas package. For example, we might define “SiO2”, as a monomer unit in the particle package. It is comprised of one silicon element and 2 oxygen elements. The molecule, SiO_2, doesn’t necessarily have to exist as a gas phase molecule in the gas phase mechanism. A cluster of size 10 comprised of this monomer unit would consist of 10 silicon atoms and 20 oxygen atoms. However, in all technical specifications in the particle package it would be referred to having a size of 10, even though it actually consists of 30 atoms.

In applications of `CADS` that use Cantera’s solid-phase thermodynamics capabilities, each monomer unit must correspond to a solid phase species in the thermodynamic phase that represents the condensed phase of the particle. In this case, there is a one-to-one correspondence between monomer units and solid phase species within Cantera.

2.3 Discrete Size Regime

Originally this project was to include a discrete size regime. We may get there yet in later years.... For now, this section will just introduce the nomenclature employed in a discrete size regime submodel. Following, we refer to the entire model as discrete section model (DSM). However, the size regime of the DSM may be broken up into a Discrete Size Regime (DSR) and a Sectional Size Regime (SSR). Our DSM model only contains a SSR region. The SSR regime is where a range of particle sizes are included in a single bin. A DSR regime is where each bin only includes a single total monomer unit value. Variations in mole fractions of individual monomer unit values may occur as well as degeneracies in the underlying geometry within a DSR bin. In the future a DSR region may be added in order to understand nucleation further. However, it would seem not to be worth it, unless detailed thermodynamic and kinetic information about molecules in the DSR region were available to make the extra computational cost worth it.

2.4 Description of Independent Variables Within the Particle Package

Within CADS, all particle sizes are expressed in terms of monomer units as the fundamental independent variable. For example, a cluster containing 25 monomer units is size 25. Describing the particles in this manner provides an unambiguous size even when multicomponent composition may produce a situation in which a particle with more monomer units has a smaller diameter or mass than a particle with fewer MU's. The concentration of clusters and particles will also be tracked in monomer units. For example, 10^{10} particles per cm^3 with size 25 monomer units is a concentration of $2.5x10^{11}$ MU per cm^3 or 4.15×10^{-13} moles of MU per cm^3.

The choice of fundamental units within a section method application is not a straightforward decision. Other researchers have used different fundamental units. However, the advantage to using monomer units is that they are easily added and subtracted, and their use allows for a method that may rigorously conserve mass, element, and particle number. Quantities such as particle diameter or particle mass are considered to be derived quantities, easily calculated from an equation of state for the particle and from the fundamental independent variables expressed in terms of MU's.

2.5 Description of a Section

The size of a particle is a continuous function of the total monomer unit size, μ. In general the particle size may vary between zero to infinity. The evolution of a distribution of particles is in general an integral-differential equation. A section method breaks this problem up by separating the continuous distributions into a finite number of sections. Section boundaries are given in units

of MU's, or monomer units. The lower bound for section 0 is the particle with the smallest number of MU's; the name is μ_0, (named `m_v0` in the code). μ_0, an important parameter for a section model implementation, is an input to the particle modeling package. Below this quantity all particles are lumped into clusters and species. Each section *L* will extend from a lower MU bound of μ_L to an upper MU bound of μ_{L+1}. The number of sections, `m_numsections`, in the section model is an input parameter to `CADS` in the `Section Model Definition` block. This is described later in Section 2.11

Within `CADS`, there is no inherent limitation in the spacing of section boundaries. One popular choice for spacing is to use a geometric progression, using a factor of 2 in size between each subsequent section, e.g. Eqn. (2.1). Spacings of this size and greater tends to minimize the cost of evaluating the coagulation operator.

$$\mu_{L+1} = 2\mu_L \tag{2.1}$$

Thus, a collision between one particle in section *L - 1* and any other particle with equal or lower MU value will result in an agglomerated particle will never fall into a section higher than section *L*. Eqn. (2.1) is a specific example of geometric spacing of sections where each section boundary is a multiple, *g*, of the next section boundary, Eqn. (2.2).

$$\mu_{L+1} = g\mu_L \tag{2.2}$$

The maximum MU value followed by the section model is named $\mu_M = \mu_{\text{m\_numsection}}$, labeled `m_vmax` within the program. μ_M is also input from the setup file. However, it may be ignored in some cases. As an alternative to geometric distributions, internal boundaries within the sections may be determined via an equispaced log base 2 distribution function, Eqn. (2.3).

$$\mu_i = 2^{\frac{\ln\mu_1 + \left(\ln\frac{\mu_M}{\mu_1}\right)\left(\frac{i-1}{M}\right)}{\ln 2}} \tag{2.3}$$

The overall control for how the boundaries of the sections are determined and what the minimum and maximum values of the section bounds are is determined by the "`Section Spacing Type`" card, described later in Section 2.11.

A sectional model is constructed by binning the continuous representation of the particle distribution function, $n(\mu, t)$, or monomer distribution function, $q(\mu, t)$, into N_s bins. Eqn. (2.4) presents the formulas for the total concentration of monomer units in the i^{th} bin, $Q_i(t)$, and the total concentration of particles in the i^{th} bin, $N_i(t)$.

$$Q_i(t) = \int_{\mu_i}^{\mu_{i+1}} q_i(\mu, t) d\mu \qquad N_i(t) = \int_{\mu_i}^{\mu_{i+1}} \frac{q_i(\mu, t)}{\mu} d\mu \tag{2.4}$$

The basic building blocks for CADS involves the monomer unit distribution function, $q_i(\mu, t)$. $q_i(\mu, t)$ is distributed within each section via the following manner. Within each section, $q_i(\mu, t)$ is described by a discontinuous Galerkin finite element representation with varying order:

$$q_i(\mu, t) = \sum_{j=1}^{NDOF} \phi_i^j(\mu) b_i^j(t). \tag{2.5}$$

$\phi_i^j(\mu)$ is the basis function for the j^{th} degree of freedom in the representation, and $b_i^j(t)$ is the coefficients for the j^{th} basis function. $b_i^j(t)$ are the basic independent variables in the section model. The term discontinuous refers to the fact that the distribution may be discontinuous across section boundaries, thus $q_{i-1}(\mu_i, t) \neq q_i(\mu_i, t)$. Note this allowed discontinuity in particle distributions does not present a problem with satisfaction with the underlying integro-differential equation representing the particle distribution. That equation, which involves but is more general than a purely advection equation, may allow for abrupt discontinuities in the particle distribution as a function of size that change as a function of time and space.

The next section will describe the types of basis functions that have been implemented.

2.6 Basis Functions for Sections

Basis functions are implemented via a virtual base class. There are currently 3 instantiations of the basis class, 2 involving a single degree of freedom (DOF-1) within the section and one involving two degrees of freedom (DOF-2) within the element. The two single degree of freedom classes differ in what is held constant within the section. This issue has received considerable attention within the literature; a lot of work has been carried out to determine what implementation gives the best overall results for the condensation and coagulation operators [14]. We note here that the virtual base class implementation of this discrete section model allows for an interested user to develop different implementations of basis functions which hold different quantities fixed rather easily.

The `sectionBF1.cpp` implementation holds the MU distribution constant within a section. This is equivalent for the equations of state used so far to holding the particle volume constant within each section.

$$q_i(\mu, t) = \phi_i^L(\mu) b_i^L(t) \text{ where } \phi_i^L(\mu) = 1 \tag{2.6}$$

Another class implementation `sectionBF1_logv.cpp` treats the MU sectional distribution as lognormal dependent on μ within the section, Eqn. (2.7), though little work has been done with this implementation.

$$q_i(\mu) = \phi_i^L(\mu) b_i^L \text{ where } \phi_i^L(\mu) = \frac{1}{\ln\left(\frac{\mu_{i+1}}{\mu_i}\right)\mu} \tag{2.7}$$

The two degree of freedom implementation, `sectionBF2.cpp`, uses the MU values at the top and bottom v values of the sections as unknowns. Then, it assumes a linear distribution of MU's within a section, Eqn. (2.8).

$$q_i(\mu) = \phi_i^L(\mu) b_i^L + \phi_i^H(\mu) b_i^H \tag{2.8}$$

where $\phi_i^L(\mu) = \frac{\mu_{i+1} - \mu}{\mu_{i+1} - \mu_i}$ and $\phi_i^H(\mu) = \frac{\mu - \mu_i}{\mu_{i+1} - \mu_i}$.

There are many choices and different alternative formulations that could have been made. For example, degrees of freedom based on increasing order of the interpolation polynomial (i.e., basis functions of 1 and μ in this case) are the common choices for discontinuous Galerkin treatments. However this choice was not made here. Also, a linear function of the particle number within the section could have been implemented instead of the MU number. The merits of these choices and alternatives can be determined in the future.

2.7 Details about Dependent vs. Independent Variables in the CADS Package

Independent variables are the formal unknowns that are part of the solution vector of an application. Dependent variables are derived from functions of independent variables. In order to formulate an exact Jacobian, for example, for the particle source terms, which is required for an implicit solution algorithm, it must be made very clear what is an independent variable and what is a dependent variable. There are two types of independent unknowns within the `CADS` package. The first type of independent variable are the coefficients for the basis functions for the distribution of the total monomer units in a section. The formula for the particle monomer unit distribution function in a section i, $q_i(\mu, t)$, for a degree of freedom 2 distribution is given by Eqn. (2.9).

$$q_i(\mu, t) = \phi_i^L(\mu) b_i^L(t) + \phi_i^H(\mu) b_i^H(t) \tag{2.9}$$

$\phi_i^L(\mu)$ and $\phi_i^H(\mu)$ are the basis functions for the low and high modes in section *i*. They are currently defined as traditional linear finite element functions of μ (see Eqn. (2.8)). b_i^L and b_i^H are the coefficients for those basis functions, and they are considered the independent variables within CADS.

The total density of monomer units in section *i*, Q_i, may be expressed in terms of the distribution, $q_i(\mu)$, by the following integral formula,

$$Q_i = \int_{\mu_i}^{\mu_{i+1}} (\phi_i^L(\mu) b_i^L + \phi_i^H(\mu) b_i^H) d\mu = d_i^L b_i^L + d_i^H b_i^H = \left(\frac{\mu_{i+1} - \mu_i}{2}\right)(b_i^L + b_i^H), \tag{2.10}$$

where d_i^L and d_i^H are coefficients of the section integral. The number of particles in a bin, N_i, is given by the following integral formula,

$$N_i = \int_{\mu_i}^{\mu_{i+1}} \frac{(\phi_i^L(\mu) b_i^L + \phi_i^H(\mu) b_i^H)}{\mu} d\mu = c_i^L b_i^L + c_i^H b_i^H, \tag{2.11}$$

where c_i^L and c_i^H, Eqn. (2.12), are coefficients of the section integral in Eqn. (2.11).

$$c_i^L = 1 - \frac{\mu_i}{\Delta\mu} \ln\left(\frac{\mu_{i+1}}{\mu_i}\right) \qquad c_i^R = \frac{\mu_{i+1}}{\Delta\mu} \ln\left(\frac{\mu_{i+1}}{\mu_i}\right) - 1 \tag{2.12}$$

Note, the corresponding Q_i and N_i coefficients for the DOF-1 implementation given in Eqn. (2.6) are derived in Eqn. (2.13).

$$d_i^1 = \mu_{i+1} - \mu_i, \; c_i^1 = \ln\left(\frac{\mu_{i+1}}{\mu_i}\right) \tag{2.13}$$

When there is more than one type of MU in the section, additional degrees of freedom are defined. These are defined as $Z_{i,k}$, the total monomer unit density of monomer unit type *k* in section *i*. $Z_{i,0}$ becomes a dependent variable given by the following formula:

$$Z_{i,0} = Q_i - \sum_{k=1}^{N^{\mathrm{MU\_type}}-1} Z_{i,k}, \tag{2.14}$$

and is never explicitly included in a vector of independent unknowns. Eqn. (2.14) and its dependencies must be reflected in the Jacobian entries, whose evaluations are influenced by the formal definition of independent vs. dependent variables. Note, because of this definition, the source terms for condensation and coagulation are independent of the values for $Z_{i,k}$, in many practical cases. This fact leads to a computational savings, because the dependence of the coagulation and condensation operators on the multiple MU unknowns, $Z_{i,k}$, may be neglected in practice.

To sum up, for the DOF-2 case with *nmu* different types of monomer units, the unknown vector for each section *i*, $\boldsymbol{Y}_i$, can be described by Eqn. (2.15).

$$\boldsymbol{Y}_i = \begin{bmatrix} b_i^L \\ b_i^H \\ Z_{i,1} \\ \cdots \\ Z_{i,nmu-1} \end{bmatrix} \tag{2.15}$$

If there is only one monomer unit type, then there are no $Z_{i,k}$ degrees of freedom in the unknown vector. The unknown solution vector for the DOF-1 case with *nmu* different degrees of freedom is given by Eqn. (2.16).

$$\Xi_i = \begin{bmatrix} b_i^L \\ Z_{i,1} \\ \cdots \\ Z_{i,nmu-1} \end{bmatrix} \tag{2.16}$$

2.8 General CADS Source Term and Discussion of Conservation Properties

We start off with the specification of the population balance equations for particles undergoing a certain set of coagulation, nucleation, surface growth, and surface evaporation reactions, Eqn. (2.17), which is the same as Eqn. (1.1) without spatial gradients. $n(\mu, t)$ is the concentration

distribution of particles (kmol particles m^{-3} MU^{-1}) having monomer unit size μ at time, t. $q(\mu, t)$ is the concentration distribution of monomer units (kmol particles m^{-3} MU^{-1}).

$\beta(\mu, \mu')$ is the coagulation rate constant for particles of monomer unit size μ and μ' in units of m^3 (kmol particles)$^{-1}$ s^{-1}. The expression in Eqn. (2.17) reflects the hyperbolic nature of the growth term [15], by recognizing that the condensation term in Eqn. (2.17) represents a continuous particle flux (i.e., velocity) in the monomer unit direction [2].

$$\frac{\partial}{\partial t}n(\mu, t) + \frac{\partial}{\partial \mu}(G(\mu, t)n(\mu, t)) = \frac{1}{2}\int_0^{\mu} n(\mu - \mu', t)n(\mu', t)\beta(\mu - \mu', \mu')d\mu' - \int_0^{\infty} n(\mu, t)n(\mu', t)\beta(\mu, \mu')d\mu' + S_{nucl}^{part}(\mu) \tag{2.17}$$

$G(\mu, t) = S(\mu, t) - E(\mu, t)$ represents the net growth rate of particles of size μ at time t. $E(\mu)$ is the total evaporation rate of particles with monomer size μ; $E(\mu)$ is comprised of the sum of the individual evaporation rates of each different monomer unit type. $S(\mu)$ is the surface growth reaction rate for particles of monomer unit size μ.

We seek a numerical method which is robust according to the following metrics. The main metrics have to do with the conservation of integral moments, M_α, of the above equation, Eqn. (2.17). For example, the zeroth moment is a specification of the change in the total number of particles, while the first moment expresses the total conservation of mass in the system. These moments are important due to the fact that they are the observables of the system, especially with regards to light scattering experiments (i.e., the moment corresponding to the square of the volume). Also, certain operations observe conservation properties with respect to these moments. For example, surface growth conserves particle number and, in combination with the gas phase, conserves mass. In our initial discussion, we will assume that the model includes just one type of monomer unit in order to simplify the formulas to focus the discussion on these numerical issues.

$$M_\alpha = \int_0^{\infty} \mu^\alpha n(\mu, t)d\mu \tag{2.18}$$

Another metric for numerical robustness of a discretization scheme is to reduce to a minimum the numerical diffusion associated with the solution of Eqn. (2.17). Numerical diffusion is the unphysically large spreading out of the particle concentration during the transient process of coagulation or condensation, especially leading to the creation of a high monomer unit tail in the

particle distribution function. This issue has been of much concern in the literature recently, though its importance in terms of understand soot formation schemes and its influence on heat transfer and kinetics within flames is an open question.

We seek to derive a sectional model by binning the continuous representation of the particle distribution function, $n(\mu, t)$, or the monomer distribution function, $q(\mu, t)$, into N_s bins. Eqn. (2.4) presents the formulas for the total concentration of monomer units in the i^{th} bin, $Q_i(t)$, and the total concentration of particles in the i^{th} bin, $N_i(t)$.

The basic building blocks for our model will involve the monomer unit distribution function, $q_i(\mu, t)$. With the usage of stoichiometric coefficients to describe reaction rates, the use of $q_i(\mu, t)$ will lead to conservation of mass properties. Conservation of particles numbers may also be assured if careful attention is paid to each source term. Here conservation of particles means that the particle number changes strictly according to that dictated by the coagulation and nucleation operator and not due to numerical artifact.

In order to calculate the time varying change in the independent variables, the section basis coefficients, and the total monomer units of type k, $Z_{i,k}(t)$, source terms for each section, $\mathbf{S}_i$, are constructed. This source term vector is one of the basic outputs from CADS. The source term for DOF-2 cases has the following format:

$$\mathbf{S}_i = \begin{bmatrix} S_i^{\text{part}} \\ S_i^{\text{TMU}} \\ S_{i,1} \\ \cdots \\ S_{i,nmu-1} \\ \mathbf{S}_i^{gas} \end{bmatrix} = \begin{bmatrix} S_i^{\text{cond, part}} + S_i^{\text{coag,part}} + S_i^{\text{nucl,part}} + S_i^{\text{bulk, part}} \\ S_i^{\text{cond,TMU}} + S_i^{\text{coag,TMU}} + S_i^{\text{nucl,TMU}} + S_i^{\text{bulk,TMU}} \\ S_{i,1}^{\text{cond}} + S_{i,1}^{\text{coag}} + S_{i,1}^{\text{nucl}} + S_{i,1}^{\text{bulk}} \\ \cdots \\ S_{i,nmu-1}^{\text{cond}} + S_{i,nmu-1}^{\text{coag}} + S_{i,nmu-1}^{\text{nucl}} + S_{i,nmu-1}^{\text{bulk}} \\ \mathbf{S}_i^{\text{cond,gas}} + \mathbf{S}_i^{\text{nucl,gas}} + \mathbf{S}_i^{\text{bulk,gas}} \end{bmatrix}. \qquad (2.19)$$

S_i^{TMU} is the source term for total monomer units within section bin i. It represents the source rate of monomer units to the particle phase for particles with sizes ranging from μ_i to μ_{i+1}; it has units of kmol MU m^{-3} s^{-1}. S_i^{part} is the source term for particles within section bin i, with units of (kmol particles) m^{-3} s^{-1}. $S_{i,k}$ is the source term for monomer units of type k within section i. It has units of (kmol MU_k) m^{-3} s^{-1}. $\mathbf{S}_i^{gas}$ is the source term for gas phase species k due to interactions with particles in (or being created within) section i. It has units of (kmol) m^{-3} s^{-1}. Each source term component is separated into 4 different kernel operations: condensation, coagulation, nucleation, and bulk-particle kinetics. Each are separable source terms and are discussed in their own chapters

in this manual. The condensation source term involves particle - gas reactions taking place on the surface of particles that don't change the number of particles. The coagulation source term involves particle-particle binary reactions and doesn't involve the gas phase. The nucleation source term involves reactions that create or destroy particles, in which gas phase species comprise either all of the reactants or all of the products. The bulk-particle kinetics involves intra-particle phase chemistry reactions that optionally may also create gas-phase species.

The source term for DOF-1 case, Eqn. (2.20), differs from the DOF2 case in that the source term for particles is dropped.

$$\mathbf{S}_i = \begin{bmatrix} S_i^{\mathrm{TMU}} \\ S_{i,1} \\ \dots \\ S_{i,nmu-1} \\ \mathbf{S}_i^{gas} \end{bmatrix} = \begin{bmatrix} S_i^{\mathrm{cond,TMU}} + S_i^{\mathrm{coag,TMU}} + S_i^{\mathrm{nucl,TMU}} + S_i^{\mathrm{bulk,TMU}} \\ S_{i,1}^{\mathrm{cond}} + S_{i,1}^{\mathrm{coag}} + S_{i,1}^{\mathrm{nucl}} + S_{i,1}^{\mathrm{bulk}} \\ \dots \\ S_{i,nmu-1}^{\mathrm{cond}} + S_{i,nmu-1}^{\mathrm{coag}} + S_{i,nmu-1}^{\mathrm{nucl}} + S_{i,nmu-1}^{\mathrm{bulk}} \\ \mathbf{S}_i^{\mathrm{cond,gas}} + \mathbf{S}_i^{\mathrm{nucl,gas}} + \mathbf{S}_i^{\mathrm{bulk,gas}} \end{bmatrix} \tag{2.20}$$

Essentially, because there is only one degree of freedom representing the distribution of particles within section *i*, there can be only one independent source term representing the moment of the distribution for that section. The total monomer unit source term is retained because it represents the conservation of mass.

Introducing more nomenclature, the vector sectional source term for section *i*, $\mathbf{S}_i$, may be broken down by physical effect, via the following Eqn. (2.21).

$$\mathbf{S}_i = \mathbf{S}_i^{coag} + \mathbf{S}_i^{cond} + \mathbf{S}_i^{nucl} + \mathbf{S}_i^{bulk} \tag{2.21}$$

$\mathbf{S}_i^{coag}$, $\mathbf{S}_i^{cond}$, $\mathbf{S}_i^{nucl}$, and $\mathbf{S}_i^{bulk}$ are the sectional vector source terms due to coagulation, condensation, nucleation, and bulk-phase particle reactions. These source terms, along with particle transport and radiation properties, are the basic building blocks for formulating zero-, one- and multi-dimensional simulations of the evolution of particle distributions.

In addition to the source term, CADS provides quasi-analytical Jacobians of these source terms with respect to the CADS independent variables and gas-phase species concentrations. "Quasi-analytical" here means that the final analytical result may have been partly or wholly derived from taking intelligently chosen numerical differences of individual source terms, under the CADS "hood". Because the monomer unit coordinate, μ, behaves somewhat like another physical dimension, intelligently choosing the numerical derivatives below the operator level, i.e., underneath the

$\mathbf{S}^{coag}$, $\mathbf{S}^{cond}$, $\mathbf{S}^{nucl}$, and $\mathbf{S}^{bulk}$ level, has been shown to lead to at least an order-of-magnitude speed-up in the calculation of a jacobian over a method which takes numerical derivatives at the overall particle source term level $\mathbf{S}_i$, due to a reduction in operation count [16].

CADS rigorously conserves elements, mass, enthalpy, and particle number in the following sense. Each of its physical effect operators, $\mathbf{S}^{coag}$, $\mathbf{S}^{cond}$, $\mathbf{S}^{nucl}$, and $\mathbf{S}^{bulk}$, conserves these quantities independently up to numerical roundoff error. The DOF-1 case conserves particle number, even though there is no explicit source term for particles; it does this, because each of the physical source terms are constructed to achieve this, albeit at the expense of increased numerical diffusion, as will be shown.

2.9 Treatment of the First and the Last Section Bins

Problems with conservation principles arise when the source terms that lead to higher particle MU values are applied to the last section in a section method. If left untreated, the conservation of mass principle and conservation of particles treatment will be violated, because for any fixed section method, there is a maximum value for the monomer unit to particle ratio, $\mathrm{MU}^{H}_{N^S-1}$. There are several different methodologies for handling the last section bin in a sectional method. The first methodology is to let large particles "disappear" from the calculation. For example, any particle which through coagulation events gets larger than $\mathrm{MU}^{H}_{N^S-1}$, simply is dropped from the calculation, with the underlying thought that extremely large particles have small Brownian diffusivities, small relative surface areas, and thus low reactivities.

However, CADS uses a different approach, which adheres to the maintenance of conservation principles. CADS turns off all source terms for the last section where particles in the last section are reactants and which leads to an increased monomer unit to particle ratio. For example, it turns off the surface growth term in the last section, if and only if the surface growth term is positive. It turns off coagulation source terms in the last section if one of the reactants for the coagulation reaction is in the last section. Therefore, the coagulation kernel can add to the concentration of particles in the last section, but it can't create particles larger than $\mathrm{MU}^{H}_{N^S-1}$.

This treatment of the last section, therefore, preserves conservation of mass and particles, at the expense of treating the last section bin as a relatively inert holding bin (except for etching surface reactions).

Etching condensation reactions applied to the first section bin require special treatment as well. Normally, for etching reactions, the condensation reaction would create a flux of particles that bin *L* to section bin *L-1*. However, this process breaks down for the section bin 0. Through various

numerical experiments, an adequate treatment for this problem has been found to be that the surface reaction in section bin 0 should remain unchanged. However, when the monomer unit to particle ratio in section bin 0 has been reduced to the minimum permissible value, MU_0^L, then the total particle number should be reduced in such a quantity that maintains conservation of mass/ elements, at the expense of the conservation of particles. The major effect of this is that it allows for etching surface growth reactions to fully depopulate a particle distribution, if given enough time. This treatment will be explained fully in the condensation chapter.

2.10 Thermodynamics

`CADS` make extensive use of Cantera's internal thermodynamics capabilities. Both the surrounding gas phase and the particle's bulk phase utilize Cantera's generic ThermoPhase object, and therefore must have a full equation of state implementation associated with it. In order for an aerosol to form and be stable with respect to the surrounding environment, formation of the bulk particle phase from elements of the surrounding gas phase, must be thermodynamically stable, i.e., must lead to a lowering of the total system Gibbs free energy. `CADS` is a truly reversible system.

However, particle surface tension effect have so far been ignored within `CADS` thermodynamics implementation in terms of their overall driving force on condensation reaction rates (p. 257, ref. [2]). Thus, the Kelvin relation has been neglected. This is probably appropriate for the initial class of problems used for `CADS`, i.e., solid phase, high temperature reactions. However, it isclearly inappropriate for liquid-phase aerosol particles at lower temperatures. In these problems, the Kelvin relation, which relates the supersaturation of the gas phase that is necessary for equilibrium to hold with respect to the size of the particle, and its effect on classical nucleation theory limits the smallest aerosol size that is attainable. For water at room temperature, this size is roughly 0.2 μm [2]. However, this limitation doesn't hold for gas->solid aerosol reactions at high temperature, because the effects of the surface energy increase is much reduced, practically to the point where nanometer-sized particles are stable relative to their monomer precursors (e.g., see the Soot, Si particle and SiO_2 particle literature).

2.11 Section Model Definition Block of The Particle Input File

Much of the set-up of the sectional bins is handled in the input block, `Section Model Definition`. Figure 2.1 displays a commented listing of the line entries in the block. The first couple of entries lay out the number and type of the particle section bins, as well as the monomer unit boundaries for the bins. The number of sections, the number of degrees of freedom in a section, and the lower bound for section 0 are required entries, and their values are always honored. The

```
START BLOCK SECTION MODEL DEFINITION
   !-------------------------------------------------------------------
   !  Number of Sections = int  (required) (default = 20)
   Number of sections = 100
   !-------------------------------------------------------------------
   ! Number of DOF's per section  = int  (required) (default = 2)
   !  limits = 1 or 2
   Number of DOF's per section = 2
   !-------------------------------------------------------------------
   !  Lower bound for section region = int
   !    (required) (default = 20)
   !    Value of the monomer unit number for the bottom boundary of the first section.
   Lower bound for section region = 35.
   !-------------------------------------------------------------------
   ! Upper bound for section region = {number}
   !      (required) (default = 2.0E6)
   !   Value of the monomer unit number for the top boundary of the highest section.
   Upper bound for section region = 280.
   !-------------------------------------------------------------------
   ! Section Spacing Type = [ 1 - 5]
   !   (required) (default = 2)
   Section spacing type = 3
   !-------------------------------------------------------------------
   ! Section Spacing File = [filename]
   !        (conditionally required)
   Section Spacing File = 1.5
   !-------------------------------------------------------------------
   ! Spacing Geometric Ratio
   !      (required)
   Spacing Geometric Ratio = 1.5
   !-------------------------------------------------------------------
   ! Number monomer unit types = [number]
   !   (conditionally required) (default = 1)
   Number of monomer unit types = 1
   !-------------------------------------------------------------------
   ! Identity of monomer unit types = species_1 species_2 ...
   !        (conditionally required)
   ! Identity of monomer unit types = C2H2
   !-------------------------------------------------------------------
   ! Dimensionality of transport = {number}
   !     (required) (limit = 1 to 3) (default = 3)
   Dimensionality of transport = 1
   !-------------------------------------------------------------------
   !  Thermal Conductivity of Particle = number
   !      (required) (mks units)
   Thermal Conductivity of Particle = 80.7
   !-------------------------------------------------------------------
   ! BLOCK MU Molecular Weights
   !       Conditionally required
   start block MU Molecular Weights
       C2H2 = 26.0
   end block   MU Molecular Weights
   !-------------------------------------------------------------------
   ! BLOCK Molar Volumes
   !        Conditionally required (units m^3/kmol
   start block Molar Volumes
       C2H2 = 0.00619
   end block Molar Volumes
   !-------------------------------------------------------------------
END BLOCK SECTION MODEL DEFINITION
```

Figure 2.1 Section Model Definition Block

`section spacing type` keyline determines how the section boundaries are laid out. The difference between linear, log base 2, and geometric spacing has been covered in section 2.5. The following numbers are valid entries for the key line.

```
      0 -  Logarithmic spacing preserving factor of 2
           geometry requirement. Upper bound input, vmax,
```

```
     is ignored.
1 -  Linear spacing.
2 -  Logarithmic spacing between specified max and min
3 -  Geometric spacing, given by the section lower bound
     and the value of the Spacing Geometric Ratio card.
     Disregard the section upper bound value.
4 -  Geometric spacing of section boundaries
     using a fixed vmax. Geometric factor is ignored.
5 - The exact spacing is read in from a file.
```

Many of the entries (# 0, 3, 5) cause the value in the subsequent `Upper bound for section region` card to be ignored. Entry #4 causes the `Spacing Geometric Ratio` card to be ignored, in contrast. If entry #5 is selected, the exact spacing may be read in from an ascii input file, with one double per line. The name of the file is read in from the `Section Spacing File` card.

The next group of cards are only needed if Cantera is not used to specify the composition of the particle bulk-phase (see Chap. 7). These cards set the number of monomer unit types (`Number monomer unit types`) and their ascii names `Identity of monomer unit types`. If Cantera is used, the information is obtained from the Cantera XML or CTI file listed in the `Cantera Particle Condensed Phase File` card. This file contains the thermodynamic description of the particle condensed phase, which of course includes the number and names of the bulk-phase species in that phase. Note, these cards may be included when Cantera is also used. The number and names of the bulk species must match exactly with what's in the Cantera file.

The `dimensionality of transport` card is used to size loops and memory storage with respect to the number of physical dimensions. This is used in transport routines that determine the terminal particle velocity, for example.

The `Thermal Conductivity of Particle` card is a required parameter to set the particle thermal conductivity (in MKS units). The particle thermal conductivity is used in the calculation of the terminal particle velocity. See Chapter 8 and ref. [17].

The last two items in the `Section Model Definition Block` are sub blocks that specify the molecular weights of monomer unit species and the molar volumes (MKS units) of monomer units. These blocks are required if Cantera is not used. If Cantera is used, then the information is redundant, but checked for internal consistency with Cantera's data. `CADS` uses a constant partial molar volume approximation for the condensed particle phase.

Note, the paradigm used by `CADS` is that if Cantera can supply the information, then the input file doesn't need to supply the same information. However, if the input file does supply the same information, an error condition results unless Cantera and `CADS` exactly agree.

2.12 Relative and Absolute Error Tolerance Conditions Within CADS

The unknowns in the discrete section method involve a range of q_i for a range of particle sizes, expressed on a grid of μ_i that varies considerably in spacing. The particles sizes can vary to such a great extent that convergence requirements for predictor-corrector methods must also take the different grid spacings in μ_i into account as well. Predictor-Corrector time-stepping methods use error control that is based on relative error control, RTOL, and an absolute error control, ATOL, approach [18, 19]. RTOL is the relative tolerance allowed in each solution variable at each time step, while ATOL is the scale of the solution variable to be used at each time step so that small values of a variable don't create undo influence on error norms. Typically a weight vector, $w[i]$ is formed for each solution component, *i*,

$$w[i] = \mathrm{RTOL}\left|y_{old}[i]\right| + \mathrm{ATOL}[i], \tag{2.22}$$

where $y_{old}[i]$ is a lagged value of the solution component, *i*, and then the weight vector is used in the calculation of a dimensionless measure of nonlinear convergence results or time-step truncation error via Eqn. (2.23).

$$Error = \left[\sum_N \frac{1}{N}\left(\frac{\Delta y[i]}{w[i]}\right)^2\right]^{1/2} \tag{2.23}$$

RTOL may be considered dimensionless, while ATOL carries with it the dimension of the solution variable, $y[i]$. Given that the independent unknowns in CADS have widely different scalings, given that they represent particles concentrations ranging in molecular sizes to millimeter sizes, and given the fact that the b_i^L and the $Z_{i,k}$ independent unknowns have different formal units, a systematic method for assigning values for $\mathrm{ATOL}[i]$ must be formulated. Below we derive a method to do this:

Let Q_{total}, Eqn. (2.24), be the total amount of monomer units in the particle phase.

$$Q_{total} = \sum_i \int_{\mu_i}^{\mu_{i+1}} q\,dv = \sum_i \overline{q}_i \Delta\mu_i \tag{2.24}$$

Typically, changes in quantities that involve factors of less than $(\mathrm{ATOL}_{\mathrm{part}})(Q_{total})$, where $\mathrm{ATOL}_{\mathrm{part}}$ is typically around 10^{-12} even for extremely accurate calculations, are insignificant and may be ignored, and additionally should be ignored. This is the essential trade-off we make when we use an ATOL requirement in error tolerances. Below a specified value of the total value,

numerical round-off errors prevent the accurate calculation of a result; convergence stalls may occur if more accuracy is requested than can be delivered given finite-precision arithmetic.

Eqn. (2.24) will induce different requirements in each of the sections due to the disparities in $\Delta\mu_i$ values. Absolute error tolerances for sectional basis coefficient unknowns, b_i^L and b_i^H, are then defined by Eqn. (2.25).

$$\mathrm{ATOL}_i = (\mathrm{ATOL}_{\mathrm{part}})\frac{(Q_{total}^*)(NDOF)}{\Delta\mu_i} min\left(1, \frac{30}{N^{\mathrm{sect}}}\right) \quad . \tag{2.25}$$

where $Q_{total}^* = Q_{total} + Q_{total}^{base}$

Eqn. (2.25) equalizes the absolute error tolerances, in terms of the variation in Q_{total} that each section *i* causes. Q_{total}^{base} is a nominally small monomer unit amount that is added in order to ensure that $ATOL_i > 0$ always. Q_{total}^{base} is set such that it is comparable to the ATOL parameter used for the surrounding gas phase species. Gas-phase species unknowns are usually expressed in terms of mass fractions. If so, the units must be put on the correct basis via Eqn. (2.26), for example assuming an ideal gas.

$$Q_{total}^{base} = \frac{RT}{P}\frac{\mathrm{ATOL}_{gas}}{\mathrm{ATOL}_{\mathrm{part}}}, \tag{2.26}$$

The other variable type in the section equation set are $Z_{i,\,imu}$'s for MU types greater than one. They require a different expression, Eqn. (2.27), for their absolute tolerances, since $Z_{i,\,imu}$ has units that are the same as Q_{total}.

$$\mathrm{ATOL}_i = (\mathrm{ATOL}_{\mathrm{part}})(Q_{total}^*)\left(min\left(1, \frac{30}{N^{\mathrm{sect}}}\right)\right). \tag{2.27}$$

Note that the absolute error tolerances will systematically increase as a function of section refinement; this problem has been corrected to first order by adding a mitigating $(N^{\mathrm{sect}})^{-1}$ multiplicative refinement factor to all error tolerances.

A unified way to represent the ATOL calculation is to define the following quantity, Q_{ATOL}.

$$Q_{\mathrm{ATOL}} = \frac{(\mathrm{ATOL}_i)\Delta\mu_i}{NDOF} \tag{2.28}$$

Q_{ATOL} is the section-bin concentration of monomer units (kmol MU m^{-3}) that CADS considers high enough to solve accurately. Concentrations lower than Q_{ATOL} are too small for CADS to worry about. Concentration higher than Q_{ATOL} are solved for accurately, independent of section bin number and particle number density.

2.13 Description of Low-Cutoffs Limits Within CADS

For the DOF-2 implementation, many of the individual algorithms for kernel operations within CADS (e.g., coagulation or condensation) have two forms of the source terms that are used: the accurate form and the stable, nonnegative form. Depending upon the value of Q_i, the total monomer units in a section, a switch is made between the stable source term and unstable source term with a ramp. One of the key items needed in the decision on whether to implement a ramp is the relative concentration of monomer units in the section bin. If the section bin has a "small concentration" of particles, then a ramp will be imposed between the accurate form and the stable, nonnegative formulation. As the concentration of particles in a section decreases to zero, the stable, nonnegative form of the kernel is employed exclusively. When there is a large concentration of monomer units in a section, the accurate form of the kernel is used.

Therefore, CADS maintains a definition of what a "small concentration" is for each section *L*, called Q_T^{cutoff}. The relationship between Q_T^{cutoff} and Q_{ATOL} was explored in terms of the numerical behavior of the algorithm [20]. The relative values of Q_T^{cutoff} and Q_{ATOL} have a large influence on numerical behavior. Q_T^{cutoff} is currently give by

$$Q_T^{cutoff} = Q_{total}^{*}\mathrm{RTOL}^{cutoff}, \tag{2.29}$$

and, the cutoff value for the section basis coefficients and MU> 1 $Z_{i,k}$ values are:

$$b_i^{cutoff} = \frac{Q_{total}^{*}\mathrm{RTOL}^{cutoff}(NDOF)}{\Delta\mu_i} \tag{2.30}$$

$$Z_i^{cutoff} = Q_{total}^{*}\mathrm{RTOL}^{cutoff}.$$

RTOL^{cutoff} is set to be a couple of orders of magnitude larger than $\mathrm{ATOL}_{\mathrm{part}}$, following the recommendations of [20].

2.14 Specification of Ramps Within CADS

As noted in the previous section, CADS makes use of ramps in its source terms to create continuous first derivative changes from accurate to numerically stable behavior for DOF-2 discretizations. Here, we describe the numerical form of the ramps. The ramps consist of cubic polynomials of the form:

$$R(\Theta) = \begin{cases} 0 & \Theta < 0 \\ 3\Theta^2 - 2\Theta^3 & 0 < \Theta < 1 \\ 1 & \Theta > 1 \end{cases} \quad \text{where } \Theta = \frac{x - x_a}{x_s - x_a} \text{ and } x_a > x_s \tag{2.31}$$

x is a functional of the current solution vector, $\boldsymbol{Y}_i$, Eqn. (2.15) (and not the time step or source term itself). x_a is the solution value or solution condition where the accurate source term evaluation method is used. x_s is the solution value or solution condition where the stable source term evaluation method is used. Ramps are used to provide final source terms of the general form:

$$\boldsymbol{S}^{final}(\boldsymbol{Y}) = (1 - R(\Theta))\boldsymbol{S}^{accurate}(\boldsymbol{Y}) + R(\Theta)\boldsymbol{S}^{stable}(\boldsymbol{Y}). \tag{2.32}$$

There are two cases where ramps are used. The first case is when the total monomer units in a section is approaching the minimum cutoff level b_i^{cutoff} discussed in the previous section. Then, typical functions for x are $x = b_i^L / b_i^{cutoff}$. $x = b_i^H / b_i^{cutoff}$ or $(min(b_i^L, b_i^H))/b_i^{cutoff}$. Typical values of x_a and x_s are $x_s = 0.1x_a$.

The second case occurs even for sections with a large populations of monomer units. If the skew in a section, i.e., the ratio of the section basis coefficient with the highest value of monomer units to the section basis coefficient with the lowest value of monomer units gets higher than a cutoff, then a ramp is imposed between source term formulations which are accurate to source term formulations which are numerically stable. For example if b_i^L is greater than b_i^H, the following value of x is defined:

$$x = \frac{b_i^L}{b_i^H} \text{ with } x_a = 10 \text{ and } x_s = 30. \tag{2.33}$$

Section bins with coefficients whose relative values don't differ by a factor of 10 are handled with accurate source term treatments. Section bins whose coefficients differ by more than a factor of 30 are handled with numerically stable source term treatments. Sections bins with coefficients have intermediate relative values are handled with hybrid ramped source term treatment with a continuous first derivative.

2.15 Interplay Between Cutoffs and Ramps

Ramps are necessary to maintain non negativity of sectional basis coefficients for the DOF-2 implementation. However, the imposition of ramps often creates a situation where the nonlinear solution problem will not be able to be solved without numerical damping of the update vectors. In general, the use of numerical damping factors is not a problem, especially far from the convergence criteria, because numerical damping is almost always needed anyway on practical problems involving kinetic systems.

However, near the convergence criteria (and below), it is always prudent to have a nonlinear system that doesn't need numerical damping to converge. If this is to be achieved, then the monomer unit cutoff values Q_T^{cutoff} must be set to be a couple of order of magnitudes greater than the Q_{ATOL} values. Then, this will ensure that the relative error tolerance will dominate the weighting vector if the solution to the system is in the ramp section, and it will probably ensure that the system will be in Newton's domain of convergence (without damping) when the criteria for satisfaction of the nonlinear equations are satisfied.

The drawback to having Q_T^{cutoff} greater than Q_{ATOL} is that numerical damping will be necessary in more cases, and there will also inevitably more time step truncation error failures, because the system becomes more nonlinear for magnitudes of the sectional basis coefficients which the predictor-corrector solver is trying to solve accurately.

However, in contrast, if Q_T^{cutoff} is set lower than Q_{ATOL}, then some of the sectional basis coefficients may be negative (though their magnitudes are below Q_{ATOL}) when the nonlinear solver has achieved convergence. A significant degree of solution cropping will have to employed to maintain positivity. In general less time-step truncation errors will result from this method, because the ramps will occur for magnitudes of the sectional basis coefficients which the predictor-corrector solver is not trying to solve accurately.

However, time-step truncation errors are a relatively small price to pay for increased non negativity and quadratic convergence properties at the end of a nonlinear iteration. And, their frequency may be mitigated by adjustments of the severity of the ramps. Some of this interplay was discovered late in the development cycle of `CADS` [20]. Therefore, the theory has only been partially implemented within the source code. Essentially, this is an important practical subject when solving the resulting non-linear equations system.

3. Condensation Operator

In this section, the numerical method for calculating the source term for condensation will be described. The part of Eqn. (2.17) related to the condensation kernel is replicated below as Eqn. (3.1). The source term comes about by separating Eqn. (3.1) into two terms representing the time derivative of the state variables of the particle distribution (e.g., see Eqn. (2.15)) and the source term due to the condensation kernel, $\boldsymbol{S}^{\text{coag}}$ (e.g., see Eqn. (2.19)). This chapter will describe the formulation for $\boldsymbol{S}^{\text{coag}}$. For a description how the time derivatives of the state variables are formulated and for how it all goes together in one case, see the later TDcads chapter.

$$\frac{\partial}{\partial t} n(\mu, t) + \frac{\partial}{\partial \mu}(G(\mu, t) n(\mu, t)) = 0 \tag{3.1}$$

$G(\mu, t)$ represents the net growth rate of particles (i.e., growth minus etching) of size μ at time t, in units of # of monomer units per second. In implementing the surface growth rate, all surface growth rates in the code are calculated on a "per surface area" basis. Then, the surface growth rates per surface area are multiplied by the surface area of a particle with μ total monomer units to obtain $G(\mu, t)$. Note, $G(\mu, t)$ is not on a kmol MU basis. $G(\mu, t)$ moves particles along the sectional basis coordinates axis, e.g., μ_0, μ_1, μ_2,..., which is on a straight "number of monomer units" basis. Further description of the calculation of $G(\mu, t)$ will be delayed until the Surface Growth Chapter. In the current chapter, it will be assumed to be a given function.

The expression in Eqn. (3.1) is a purely hyperbolic equation [15]. The condensation term in Eqn. (3.1) may be considered to represent a continuous particle flux (i.e., velocity) in the monomer unit direction [2], analogous to other types of hyperbolic equations.

It will be shown that a first-order discontinuous Galerkin treatment [21, 22, 23, 24] of the condensation reaction equation leads to a numerical method which conserves mass and particle number and which exhibits a small amount of numerical diffusion. Also, because the mesh is fixed with respect to the boundaries of the sectional bins, the method allows for the addition of one or more physical coordinates into the equation system with relative ease.

First, the MU coordinate direction is split into N_s sections, which are considered as "elements." A solution defined on the finite-element space, W_h, of piecewise polynomials of degree $r \geq 0$, $P_r(K)$, is sought, with no continuity requirements across interelement boundaries, defined on the set of finite elements, T_h, whose size is bounded in all dimensions by the size, h.

$$W_h = \{\Phi \in L_2(\Omega): \ \phi|_K \in P_r(K) \quad \forall K \in T_h\} \tag{3.2}$$

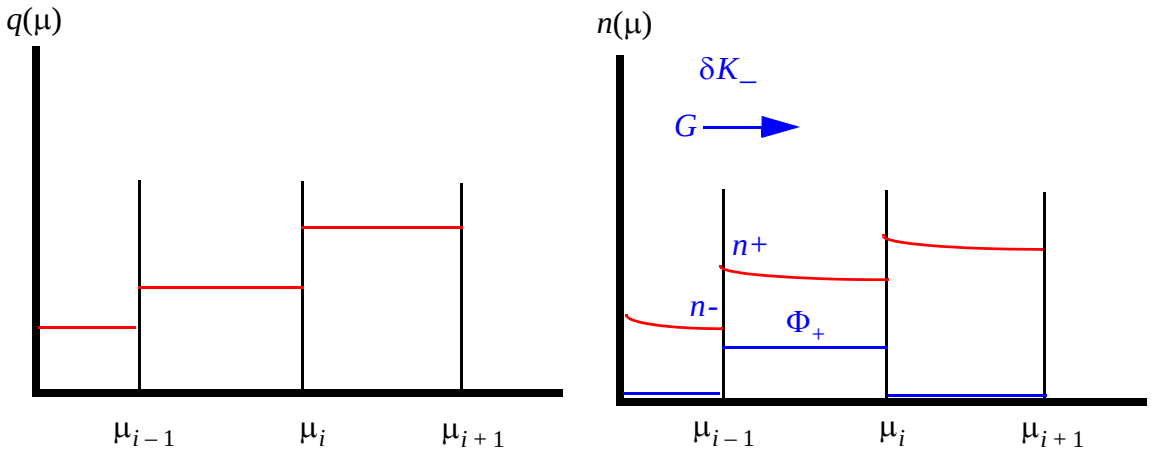

Figure 3.1 Schematic of the setup of the 0-D Discontinuous Galerkin Method. $q(\mu)$ is the monomer unit distribution within a section, which is constant. $n(\mu)$ is the particle concentration in a section, which is equal to $q(\mu)/\mu$. The terms for the setup of the boundary integral in Eqn. (3.5) are depicted in blue.

Next, the boundary ∂K of the finite element K is split according to the direction of the particle flux, i.e., the sign of $\boldsymbol{G}$ used in the previous direction. Let ∂K_- represent the inflow part of the finite element boundary, while ∂K_+ will represent the outflow part of the boundary.

$$\partial K_- = \{x \in \partial K: \ \boldsymbol{n}(x) \bullet \boldsymbol{G}(x) < 0\} \tag{3.3}$$

$$\partial K_+ = \{x \in \partial K: \ \boldsymbol{n}(x) \bullet \boldsymbol{G}(x) \geq 0\} \tag{3.4}$$

$\boldsymbol{n}(x)$ is the outward unit normal to ∂K at x. Therefore, $\boldsymbol{n}(x) \bullet \boldsymbol{G}(x)$ represents an inflow into the finite element.

Following the original literature on the discontinuous galerkin method [21,22,23], the equation representing particle condensation, Eqn. (3.5), may be written as

$$\int_{\partial K_-} (n_- - n_+)(\boldsymbol{n}(x) \bullet \boldsymbol{G}(x))\phi_+ ds + \int_K \left(\frac{\partial}{\partial t} n(\mu, t) + \frac{\partial}{\partial \mu}(G(\mu, t) n(\mu, t))\right)\phi d\mu = 0 \tag{3.5}$$

Extra surface terms are only added for upstream interfaces, ∂K_-, not downstream interfaces, ∂K_+. $n_- - n_+$ represents the jump discontinuity across the ∂K_- interface created by the discontinuous interpolation of the independent variable, $n(\mu)$, across element boundaries. n_- represents the upstream component, which in this case, is calculated from the finite element interpolation of n

from the adjacent upstream element. n_+ represents the finite element interpolation of n at the $\partial K\_$ boundary. ϕ_+ is the value of the finite element basis function on the + side of the interface at the $\partial K\_$ boundary. K represents a one dimensional finite element, i.e. a discretization of the monomer unit coordinate, μ. Extensions to higher dimensional finite elements via inclusion of physical coordinates will be discussed later. Also, some authors have included the time coordinate as another coordinate of the finite element instead of treating the time derivative via the method of lines; this is called the "space-time method", and has been shown to be effective. The space-time method won't be used here, as the applications that use the particle package all use either the method of lines or a simple backwards Euler time-stepping strategy to handle time derivatives. Also, the particle's time derivatives will eventually be coupled into the global time derivatives of the application, which has its own time scales.

The inclusion of the jump discontinuity in Eqn. (3.5) may be understood by noting that if the other term are integrated by parts, then Eqn. (3.5) would resemble the standard finite element method applied to the single element, K, with weakly imposed boundary conditions applied to the upstream boundaries. Now, the Galerkin method may be formulated by specifying that the solution should be such that for all $n^h \in W_h$, such that n^h satisfies Eqn. (3.5) for all $\phi \in W_h$.

Each finite element, i, will correspond to a sectional bin. If for the moment, G is assumed to be positive everywhere for the case of pure condensation without etching, Eqn. (3.5) may be simplified to Eqn. (3.6).

$$(n_+(\mu_i) - n\_(\mu_i))G(\mu_i)\phi_+(\mu_i) + \int_{\mu_i}^{\mu_{i+1}} \left(\frac{\partial}{\partial t}n(\mu, t) + \frac{\partial}{\partial \mu}(G(\mu, t)n(\mu, t))\right)\phi d\mu = 0 \qquad (3.6)$$

First, formulate an example for the case where $\phi_i(\mu) = 1$ is a constant function for section *i*, and zero in all other sections. The notation $\phi_i^c(\mu)$ will be used for this basis function. Each section, *i*, has a constant value for the monomer unit concentration within the section, $q_i = b_i(t)\phi_i^c(\mu)$. The particle concentration within each section has the form, $n_i = (b_i(t)\phi_i^c(\mu))/\mu$.

Using this expression for n_i with $\phi_i^c(\mu) = 1$ within section *i*, Eqn. (3.6) reduces to Eqn. (3.7), which can be seen to a standard upwind differencing scheme applied to a hyperbolic equation system for a control volume centered at the section midpoint:

$$\left(\ln\left(\frac{\mu_{i+1}}{\mu_i}\right)\right)\frac{\partial b_i}{\partial t} + G(\mu_{i+1})\frac{b_i}{\mu_{i+1}} - G(\mu_i)\frac{b_{i-1}}{\mu_i} = 0 \ . \qquad (3.7)$$

And, equivalently the equation for the control volume *i-1* becomes:

$$\left(\ln\left(\frac{\mu_i}{\mu_{i-1}}\right)\right)\frac{\partial b_{i-1}}{\partial t} + G(\mu_i)\frac{b_{i-1}}{\mu_i} - G(\mu_{i-1})\frac{b_{i-2}}{\mu_{i-1}} = 0 \tag{3.8}$$

It can be seen from Eqn. (3.7) and Eqn. (3.8) that the scheme conserves particles, since the same number of particles flow out of a lower section as flow into the section just above it. However, let's determine whether Eqn. (3.7) conserves monomer units.

The continuum expression for monomer unit conservation applied to a single section *i* may be derived from Eqn. (3.1) by multiplying first by μ and then integrating over the section, yielding Eqn. (3.9).

$$\frac{\partial Q_i}{\partial t} + G(\mu_i)q(\mu_{i+1}) - G(\mu_i)q(\mu_i) = \int_{\mu_i}^{\mu_{i+1}} \frac{q(\mu)}{\mu} G(\mu)d\mu = R_i \tag{3.9}$$

R_i is defined by Eqn. (3.9). The discrete analog of Eqn. (3.9) summed up over all sections may be written as Eqn.(3.10) assuming intersectional particle fluxes conserve mass and the growth rate is linearly interpolated.

$$\sum_{i=1}^{N_s}\left(\frac{b_i^{\,n} - b_i^{\,n-1}}{\Delta t}\right)\Delta\mu_i = \sum_{i=1}^{N_s}\ln\left(\frac{\mu_{i+1}}{\mu_i}\right)b_i\left(\frac{G(\mu_{i+1}) + G(\mu_i)}{2}\right) \tag{3.10}$$

Eqn. (3.8) can not be made equivalent to Eqn. (3.10). In fact, if G and n are constant over all sections, then Eqn. (3.8) would yield an unphysically zero growth rate for the total monomer units in contrast to Eqn. (3.10). The fact that the discretization scheme produced by the zeroth order, one-unknown-per-section, discontinuous Galerkin method expressed by Eqn. (3.7) doesn't conserve mass should not be a surprise considering there is only one degree of freedom in the section and Eqn. (3.7) was constructed from the discontinuous Galerkin method to conserve particles.

Instead of Eqn. (3.7), a discretization scheme is sought that conserves both particle number and monomer units. This may be achieved by starting with Eqn. (3.9) and (3.10) to define the end result of monomer unit conservation. Following ref. [25], a certain number of particles are moved into the next highest section in such a way as to conserve particle number while satisfying the generalized form of Eqn. (3.9). To start, the flux of particles into bin *i* from bin $i-1$ is ignored for now. Let MU_i and MU_{i+1} be the average number of monomer units per particle in section i and $i+1$; For one degree of freedom sections, these are constant quantities. Then, the conservation of particles property during a backwards Euler step from $t = t_{n-1}$ to $t = t_n$ is

$$\frac{1}{\mathrm{MU}_i}\left(\frac{Q_i^{\,n} - Q_i^{\,n-1}}{\Delta t}\right) + \frac{f_{Q,\,i+1}^{n}}{\mathrm{MU}_{i+1}} = 0 \tag{3.11}$$

In Eqn. (3.11), $f_{Q,\,i+1}^{n}$ is the flux of monomer units from section *i* to section $i+1$. $f_{Q,\,i+1}^{n}$ will be determined from the mass conservation principle. The generalized form of Eqn. (3.9) that accounts for the addition of MU's into section $i+1$ is

$$\left(\frac{Q_i^{\,n} - Q_i^{\,n-1}}{\Delta t}\right) + f_{Q,\,i+1}^{n} = R_i^n . \tag{3.12}$$

Eqns. (3.11) and (3.12) represent a two equation - two unknown system that may be solved to yield the final result:

$$\left(\frac{Q_i^{\,n} - Q_i^{\,n-1}}{\Delta t}\right) = \frac{-\mathrm{MU}_i R_i^n}{\mathrm{MU}_{i+1} - \mathrm{MU}_i}$$
$$f_{Q,\,i+1}^{n} = \frac{\mathrm{MU}_{i+1} R_i^n}{\mathrm{MU}_{i+1} - \mathrm{MU}_i} \tag{3.13}$$

$Q_i^{\,n} - Q_i^{\,n-1}$ is always negative as $\mathrm{MU}_{i+1} - \mathrm{MU}_i > 0$. Thus, the requirement of simulatneously conserving particles and mass leads to a negative net monomer production rate for section *i* due to condensation processes within section *i*. Monomer units are promoted to the next section where the number of monomer units per particle is greater in order not to grow the number of particles even when the total monomer units in the particle phase is increasing. Adding in the flux from section $i-1$ to section *i*, which we initially ignored, the general expression for section *i* may be written as

$$\left(\frac{Q_i^{\,n} - Q_i^{\,n-1}}{\Delta t}\right) + f_{Q,\,i+1}^{n} = R_i^n + f_{Q,\,in}^{n} . \tag{3.14}$$

Numerical diffusion, where the particle distribution function is unphysically spread to higher monomer unit particles, will result from the scheme. However, Eqn. (3.14) is far preferable due to its conservation properties than Eqn. (3.7), especially considering the availability of the alternative scheme discussed in the next section which exhibits little numerical diffusion. Therefore, Eqns. (3.13) and (3.14) has been adopted as the condensation algorithm for 1 DOF-per-section interpolations with `CADS`.

Note, if R_i^n is negative in Eqn. (3.13), i.e., etching of particles in section *i* is occurring, an analogous 2x2 system is solved, involving the calculation a particle flux from section *i* to section $i-1$, that also conserves monomer units and particles.

3.1 First-order Discontinuous Galerkin Derivation

Let's now formulate an expression for the first-order discontinuous Galerkin discretization of Eqn. (3.6). The monomer unit concentration in section i is represented by a standard discontinuous linear finite element basis function interpolation:

$$q(\mu, t) = b_i^L(t)\phi_i^L(\mu) + b_i^H(t)\phi_i^H(\mu), \tag{3.15}$$

where

$$\phi_i^L(\mu) = \frac{\mu_{i+1} - \mu}{\mu_{i+1} - \mu_i} \text{ and } \phi_i^H(\mu) = \frac{\mu - \mu_i}{\mu_{i+1} - \mu_i}$$

Additionally, $n = q/\mu$. For the purposes of the next derivation where conservation properties are derived, this may be rephrased via a linear combination as

$$\begin{aligned} q(\mu, t) &= b_i^L(t)(\phi_i^L(\mu) + \phi_i^H(\mu)) + (b_i^H(t) - b_i^L(t))\phi_i^H(\mu) \\ &= a_i + b_i\phi_i^H(\mu) \end{aligned}, \tag{3.16}$$

where $a_i = b_i^L(t)$ and $b_i = b_i^H(t) - b_i^L(t)$, and the basis functions redefined as $\phi_i^o = 1$ and $\phi_i^1 = \phi_i^H(\mu)$. Then, Eqn. (3.5) may be applied to section *i*, first using ϕ_i^o to obtain

$$\left(\frac{a}{\mu_i} - n_-(\mu_i)\right)G(\mu_i) + \int_{\mu_i}^{\mu_{i+1}} \left(\frac{\partial}{\partial t}n(\mu, t)\right)d\mu + G(\mu_{i+1})\frac{(a+b)}{\mu_{i+1}} - G(\mu_i)\frac{a}{\mu_i} = 0,$$

which simplifies to yield

$$\int_{\mu_i}^{\mu_{i+1}} \left(\frac{\partial}{\partial t}n(\mu, t)\right)d\mu + G(\mu_{i+1})\frac{(a+b)}{\mu_{i+1}} - G(\mu_i)n_-(\mu_i) = 0. \tag{3.17}$$

or

$$\int_{\mu_i}^{\mu_{i+1}} \left(\frac{\partial}{\partial t} n(\mu, t)\right) d\mu + G(\mu_{i+1}) \frac{q_i(\mu_{i+1})}{\mu_{i+1}} - G(\mu_i) \frac{q_{i-1}(\mu_i)}{\mu_i} = 0 \tag{3.18}$$

Eqn. (3.18) is the particle conservation equation assuming a first order interpolation of the particle density within the section (note, $n(\mu_{i+1}) = q_i(\mu_{i+1})/\mu_{i+1}$ and $n(\mu_i) = q_i(\mu_i)/\mu_i$).

The second equation, Eqn. (3.19), of the two equation, two unknown system for the unknowns corresponding to linear basis functions may be derivied by applying ϕ_i^H in Eqn. (3.5).

$$\int_{\mu_i}^{\mu_{i+1}} \left(\frac{\partial}{\partial t} n(\mu, t) + \frac{\partial}{\partial \mu}(G(\mu, t) n(\mu, t))\right)\left(\frac{\mu - \mu_i}{\mu_{i+1} - \mu_i}\right) d\mu = 0 \tag{3.19}$$

The jump discontinuity term for this equation is zero, because the linear basis function, $\phi_i^H(\mu)$, is zero at μ_i. A coordinate transformation, $\zeta = (\mu - \mu_i)/(\mu_{i+1} - \mu_i)$ is carried out next on Eqn. (3.19) to give

$$(\Delta\mu_i)\int_0^1 \frac{\partial}{\partial t} n(\mu, t)\zeta d\zeta + \int_0^1 \frac{\partial}{\partial \zeta}(G(\mu, t) n(\mu, t))\zeta d\zeta = 0 .$$

After integration by parts, Eqn. (3.19) may be expressed as

$$\int_{\mu_i}^{\mu_{i+1}} \frac{\partial}{\partial t} n(\mu, t)(\mu - \mu_i) d\mu + G(\mu_{i+1}) n(\mu_{i+1}) \Delta\mu_i - \int_{\mu_i}^{\mu_{i+1}} G(\mu, t) n(\mu, t) d\mu = 0 . \tag{3.20}$$

Next, addition of Eqn. (3.17) multiplied by μ_i yields

$$\int_{\mu_i}^{\mu_{i+1}} \frac{\partial}{\partial t} q(\mu, t) d\mu + G(\mu_{i+1}) q_i(\mu_{i+1}) - G(\mu_i) q_{i-1}(\mu_i) = \int_{\mu_i}^{\mu_{i+1}} G(\mu, t) n(\mu, t) d\mu . \tag{3.21}$$

Eqn. (3.21) represents the equation for conservation of monomer units within the section. Satisfaction of the total conservation of monomer units equation, obtained by summing Eqn. (3.21) over all sections, follows.

We have shown that an out-of-the-box first-order discontinuous Galerkin treatment of the hyperbolic discrete section model results in a numerical scheme which is conservative with respect to total mass and particle number. Since these conservation properties represent the zeroth and first moment of the particle number density integral, an additional question to answer is whether a second order discontinuous Galerkin method would additionally conserve the second moment of the particle number density. This conjecture is true, but not shown here.

In summary, for the DOF-2 implementation of condensation within CADS, Eqns. (3.18) and (3.21) are used to determine the unknown coefficients basis function coefficients, $b_i^L(t)$ and $b_i^H(t)$ in section i.

Note, in the above treatment, the assumption that G was positive was used implicitly. The case where G is negative is an extension of the above treatment. In this case, Eqns. (3.18) and (3.21) hold with the particle and total monomer unit TMU concentration vectors to use taken from the "upstream" side of the interface. However, now the upstream side of the interface is switched. The etching equations are given in Eqn. (3.22) and (3.23), where $G(\mu) < 0$:

$$\int_{\mu_i}^{\mu_{i+1}} \left(\frac{\partial}{\partial t} n(\mu, t)\right) d\mu + G(\mu_{i+1}) \frac{q_{i+1}(\mu_{i+1})}{\mu_{i+1}} - G(\mu_i) \frac{q_i(\mu_i)}{\mu_i} = 0 \tag{3.22}$$

$$\int_{\mu_i}^{\mu_{i+1}} \frac{\partial}{\partial t} q(\mu, t) d\mu + G(\mu_{i+1}) q_{i+1}(\mu_{i+1}) - G(\mu_i) q_i(\mu_i) = \int_{\mu_i}^{\mu_{i+1}} G(\mu, t) n(\mu, t) d\mu \tag{3.23}$$

3.2 Extension of Discontinuous Galerkin Treatment to the Multiple Monomer Unit Case

The treatment derived in the previous section may be extended to the case where there are multiple monomer units in each particle. There is no direct analog for conservation equation for one monomer unit type like Eqn. (3.21), which pertains to the total MU's in the particles and to the particle number density. Instead, a monomer unit balance on a single monomer unit type due to condensation processes, Eqn. (3.24), is constructured.

$$\int_{\mu_i}^{\mu_{i+1}} \frac{\partial}{\partial t} q_{jmu,\,i}(\mu, t)\,d\mu + G(\mu_{i+1}) q_{jmu,\,i}(\mu_{i+1}, t) - G(\mu_i) q_{jmu,\,i-1}(\mu_i, t) = \int_{\mu_i}^{\mu_{i+1}} G_{jmu}(\mu, t) n_i(\mu, t)\,d\mu \tag{3.24}$$

$q_{jmu}(\mu, t)$ is the monomer unit concentration of type *jmu*.

$$q_{jmu,\,i}(\mu, t) = X_{jmu,\,i} q_i(\mu, t) \tag{3.25}$$

$X_{jmu,\,i}$ is the mole fraction of monomer units of type *jmu* in section *i*. Other relations follow:

$$q_i(\mu, t) = \sum_{jmu=0}^{nmu} q_{jmu,\,i}(\mu, t) \qquad 1 = \sum_{jmu=0}^{nmu} X_{jmu,\,i}$$

Thus, Eqn. (3.24) represents a species balance of *jmu* with respect to the current section; the first term on the lhs is the time derivative of the *jmu* monomer unit on the section, *i*. The other terms on the left side are due to the flux of of species *jmu* in particles entering and leaving the section due to condensation. $G(\mu)$ is the growth rate of a particle with total monomer unit concentration μ, with units of kmol MU sec^{-1}. $G_{jmu}(\mu, t)$ represents the condensation rate of species *jmu* on particles with total monomer unit numbers equal to μ. $n(\mu, t)$ is the concentration of particles with size μ. With the discrete Galerkin treatment, we must be careful to take into account the jump discontinuities in the monomer unit concentrations at the interfaces between sections when applying Eqn. (3.24). This requires that the $q_{jmu}(\mu, t)$ used in the intersectional flux expressions in Eqn. (3.24) be taken from the correct “upwind” side of the intersectional boundary.

Eqn. (3.24) provides one equation for the total concentration of each monomer unit within the section. If Eqn. (3.24) is summed up over all different monomer unit types, Eqn. (3.21) is properly obtained. Therefore, one of the balance equations for the distinct monomer unit types, the first one for `CADS`, is not linearly independent and must be discarded when solving the system. Eqn. (3.24) only provides one equation for each monomer unit type. Therefore, `CADS` does not allow variations in MU mole fractions within a section.

3.3 Filtering Techniques for DOF-2 Discontinuous Galerkin Method

Numerical experiments quickly indicated that while the discontinuous Galerkin method showed great promise, it also needed various filtering schemes to maintain monotonicity and numerical robustness [23, 26]. For example, during the growth of an initial gaussian particle distribution in a repleneshing growth medium, one issue that arose occurred in the tail of the distribution. The sections did not drain uniformly to zero particle and MU concentration as they should have. Instead, they drained to a skewed MU distribution whose overall particle density was very near zero, but still $q(\mu_{i+1}) > 0$ and $q(\mu_i) < 0$.

The equations for a single section may be analyzed to understand why this occurs with the raw equation set, Eqns. (3.18) and (3.21). These are presented in Eqn.(3.26), where we have ignored any particles entering the i^{th} section, an appropriate assumption for a growing distribution.

$$\begin{bmatrix} \left(\frac{\mu_{i+1}}{\Delta\mu}\ln\left(\frac{\mu_{i+1}}{\mu_i}\right)-1\right) & \left(1-\frac{\mu_i}{\Delta\mu}\ln\left(\frac{\mu_{i+1}}{\mu_i}\right)\right) \\ \frac{\Delta\mu}{2} & \frac{\Delta\mu}{2} \end{bmatrix} \begin{bmatrix} \frac{\partial b_i^L}{\partial t} \\ \frac{\partial b_i^H}{\partial t} \end{bmatrix} = \begin{bmatrix} -\frac{q_i(\mu_{i+1})}{\mu_{i+1}}G(\mu_{i+1}) \\ \int_{\mu_i}^{\mu_{i+1}} \frac{q(\mu)}{\mu}G(\mu)d\mu - q_i(\mu_{i+1})G(\mu_{i+1}) \end{bmatrix} \tag{3.26}$$

b_i^L is the MU concentration on the lower node of the i^{th} section, i.e., at an MU value of μ_i. b_i^H the MU concentration on the upper node of the i^{th}section, i.e., at an MU value of μ_{i+1}. $q_i(\mu_{i+1})$ is the concentration of MU's in particles at μ_{i+1}, equal to b_i^H. $G(\mu)$ is the growth rate in MU per second of a particle of MU size μ. Even if the growth rate per unit surface area is independent of μ, there is a surface area dependence on μ that makes $G(\mu)$ a nontrivial function of μ. $q(\mu)$ is a linear function in μ of the following form:

$$q_i(\mu) = b_i^L\phi_i^H(\mu) + b_i^H\phi_i^H(\mu) \text{ where} \tag{3.27}$$

$$\phi_i^L(\mu) = \frac{\mu_{i+1}-\mu}{\mu_{i+1}-\mu_i} \text{ and } \phi_i^H(\mu) = \frac{\mu-\mu_i}{\mu_{i+1}-\mu_i}.$$

The *L* and *H* refer to the lower and higher side of the sectional distribution within a section. Thus, we may rewrite Eqn. (3.26) with no loss of generality to the following form:

$$\begin{bmatrix} \left(\frac{\mu_{i+1}}{\Delta\mu}\ln\left(\frac{\mu_{i+1}}{\mu_i}\right)-1\right) & \left(1-\frac{\mu_i}{\Delta v}\ln\left(\frac{\mu_{i+1}}{\mu_i}\right)\right) \\ \frac{\Delta\mu}{2} & \frac{\Delta\mu}{2} \end{bmatrix} \begin{bmatrix} \frac{\partial b_i^L}{\partial t} \\ \frac{\partial b_i^H}{\partial t} \end{bmatrix} = \begin{bmatrix} -\frac{q_i(\mu_{i+1})}{\mu_{i+1}}G(\mu_{i+1}) \\ G_V^L b_i^L + G_V^H b_i^H - q_i(\mu_{i+1})G(\mu_{i+1}) \end{bmatrix} \quad (3.28)$$

where

$$G_V^L = \int_{\mu_i}^{\mu_{i+1}} \frac{\phi_i^L(\mu)}{\mu} G(\mu)dv \text{ and } G_V^H = \int_{\mu_i}^{\mu_{i+1}} \frac{\phi_i^H(\mu)}{\mu} G(\mu)d\mu .$$

A necessary condition for the stability of the numerical method and the disappearance of the skewness in the distribution is given below:

$$\frac{\partial b_i^L}{\partial t} \geq 0 \text{ , when } b_i^L = 0 \text{, } b_i^H \geq 0 \text{, and } G_V^{L,H} > 0 . \quad (3.29)$$

In order to ensure that this is the case, we solve Eqn. (3.28) by inverting the matrix:

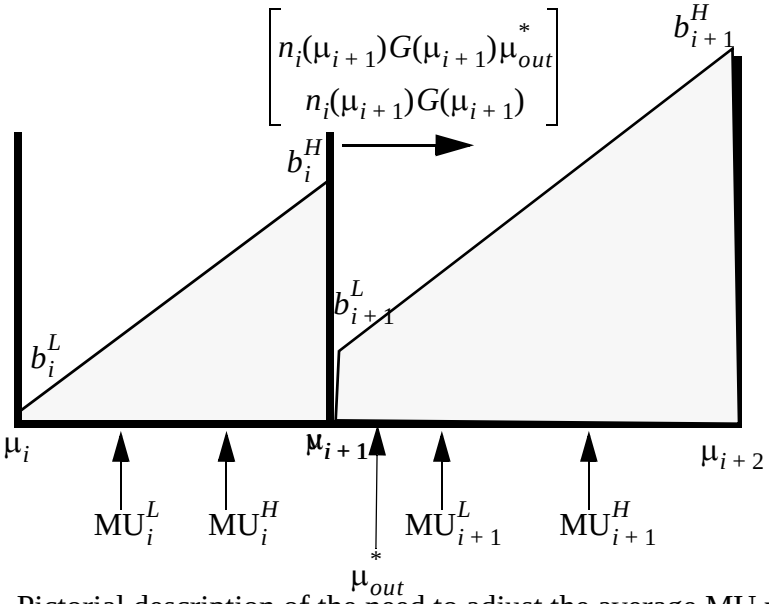

Figure 3.2 Pictorial description of the need to adjust the average MU value of particles for sections which are draining due to the condensation kernel. The value is adjusted from μ_{i+1} to μ_{out}^*.

$$\begin{bmatrix} \frac{\partial b_i^L}{\partial t} \\ \frac{\partial b_i^H}{\partial t} \end{bmatrix} = \frac{1}{DET} \begin{bmatrix} \frac{\Delta\mu}{2} & -c_i^H \\ -\frac{\Delta\mu}{2} & c_i^L \end{bmatrix} \begin{bmatrix} -\frac{q_i(\mu_{i+1})}{\mu_{i+1}} G(\mu_{i+1}) \\ G_V^L b_i^L + G_V^H b_i^H - q_i(\mu_{i+1})\mu_{i+1} G(\mu_{i+1}) \end{bmatrix} \tag{3.30}$$

where

$$c_i^H = \left(1 - \frac{\mu_i}{\Delta\mu}\ln\left(\frac{\mu_{i+1}}{\mu_i}\right)\right),\ c_i^L = \frac{\mu_{i+1}}{\Delta\mu}\ln\left(\frac{\mu_{i+1}}{\mu_i}\right) - 1 \text{, and } n_i = c_i^L b_i^L + c_i^H b_i^H$$

$$DET = -\Delta\mu + \frac{\mu_{i+1} + \mu_i}{2}\ln\left(\frac{\mu_{i+1}}{\mu_i}\right) > 0$$

n is the total particle density in the section, while c_i^L and c_i^H are the coefficients for the evaluation of *n* from the b_i^L and b_i^H, the independent variables in the problem, determined from the volumetric integrations over the section.

Solving the first equation in Eqn. (3.30) for the time derivative of b_i^L, Eqn. (3.31) results.

$$(DET)\frac{\partial b_i^L}{\partial t} = -\frac{\Delta\mu}{2}\frac{q_i(\mu_{i+1})}{\mu_{i+1}} G(\mu_{i+1}) - c_i^H (G_V^L b_i^L + G_V^H b_i^H - q_i(\mu_{i+1}) G(\mu_{i+1}))$$

$$(DET)\frac{\partial b_i^L}{\partial t} = -c_i^H G_V^L b_i^L - \frac{\Delta\mu}{2}\frac{b_i^H}{\mu_{i+1}} G(\mu_{i+1}) - c_i^H G_V^H b_i^H + c_i^H b_i^H G(\mu_{i+1})$$

$$(DET)\frac{\partial b_i^L}{\partial t} = -c_i^H G_V^L b_i^L + b_i^H \left(\frac{c_i^H G(\mu_{i+1})}{\mu_{i+1}}\right)\left(-\mathrm{MU}_i^H - \frac{G_V^H}{G(\mu_{i+1})}\mu_{i+1} + \mu_{i+1}\right) \tag{3.31}$$

$$\text{where } \mathrm{MU}_i^H = \frac{\Delta\mu}{2c_i^H}$$

MU_i^H is the average monomer units per particle in the high mode of the sectional distribution.

Applying the condition in Eqn. (3.29), it's seen that there will be a problem if

$$(\mathrm{MU}_i^H) + \frac{G_V^H}{G(\mu_{i+1})}\mu_{i+1} > \mu_{i+1} \tag{3.32}$$

Unfortunately, Eqn. (3.32) frequently occurs in practice. Therefore, as depicted in Figure. (3.2), an adjustment in the average monomer units per particle leaving section *i*, μ_{out}^{*}, is made to the source term for section *i* when necessary to prevent the skew numerical instability of the particle distribution. The value of μ_{out}^{*} is given by Eqn. (3.33).

$$\mu_{\text{out}}^{*} = \text{MU}_i^H + \frac{G_V^H}{G(\mu_{i+1})}\mu_{i+1} \tag{3.33}$$

For practical cases μ_{out}^{*} turns out to be fairly close but always larger than μ_{i+1}. For example, if we make the assumption that $G_V^H = G(\mu_{i+1})c_i^H$, then test cases for Eqn. (3.33) may be evaluated. If $\mu_{i+1} = 60$ and $\mu_i = 30$, then $\text{MU}^H = 48.88$ and $\mu_{\text{out}}^{*} = 67.295$. If $\mu_{i+1} = 36$ and $\mu_i = 30$, then $\text{MU}^H = 33.94$ and $\mu_{\text{out}}^{*} = 37.12$. Additionally, it can also be shown that if $q_H = 0$ and μ_{out}^{*} is set to Eqn. (3.33), then $dq_H/dt < 0$.

Thus, a filter of the type in Eqn. (2.32) may be created with the following requirements. If $G > 0$, and either $b_i^L \ll b_i^H$ or $b_i^L < b_i^{cutoff}$ defined in Eqn. (2.30), then set μ_{out}^{*} from it's original value of μ_{i+1} to the larger value in Eqn. (3.33). A ramp in setting this condition is provided via Eqn. (2.32) so that final source term vector is continuous.

Note, the determinant, DET, is always positive, but small and proportional to $\Delta\mu^2$, suggesting that there may be problems with numerical behavior of the algorithm appearing for very small values of $\Delta\mu$.

The filtering results may be reexpressed in terms of partial source term vectors, originally described in Section 2.8. Unfiltered, assuming postive values of $G(\mu)$, the source term vector due to condensation from growth within section *i* and movement out of section *i*, $\boldsymbol{S}_i^{\text{cond}}$, is equal to the rhs of Eqn. (3.26), repeated here with the extra terms for addition to section $i+1$ spelled out.

$$\boldsymbol{S}_i^{\text{cond}} = \begin{bmatrix} S_i^{\text{cond,part}} \\ S_i^{\text{cond,TMU}} \\ S_{i+1}^{\text{cond,part}} \\ S_{i+1}^{\text{cond,TMU}} \end{bmatrix} = \begin{bmatrix} -n_i(\mu_{i+1})G(\mu_{i+1}) \\ \displaystyle\int_{\mu_i}^{\mu_{i+1}} \frac{q(\mu)}{\mu} G(\mu)d\mu - n_i(\mu_{i+1})\mu_{i+1}G(\mu_{i+1}) \\ n_i(\mu_{i+1})G(\mu_{i+1}) \\ n_i(\mu_{i+1})\mu_{i+1}G(\mu_{i+1}) \end{bmatrix} \tag{3.34}$$

Note, the full condensation source term, $\boldsymbol{S}^{\text{cond}}$, is obtained from

$$\boldsymbol{S}^{\text{cond}} = \sum_{i=0}^{N} \boldsymbol{S}_i^{\text{cond}} \tag{3.35}$$

Then, with the filtering, the unconditionally stable form of the $\boldsymbol{S}_i^{\text{cond}}$ source term (for $G(\mu) > 0$) becomes:

$$\boldsymbol{S}_i^{\text{cond,filtered}} = \begin{bmatrix} S_i^{\text{cond,part}} \\ S_i^{\text{cond,TMU}} \\ S_{i+1}^{\text{cond,part}} \\ S_{i+1}^{\text{cond,TMU}} \end{bmatrix} = \begin{bmatrix} -n_i(\mu_{i+1})G(\mu_{i+1}) \\ \int_{\mu_i}^{\mu_{i+1}} \frac{q(\mu)}{\mu} G(\mu)d\mu - n_i(\mu_{i+1})\mu_{\text{out}}^{*}G(\mu_{i+1}) \\ n_i(\mu_{i+1})G(\mu_{i+1}) \\ n_i(\mu_{i+1})\mu_{\text{out}}^{*}G(\mu_{i+1}) \end{bmatrix}. \tag{3.36}$$

A ramp is used to change between the more accurate treatment of Eqn. (3.34) and the stable treatement of Eqn. (3.36).

$$\boldsymbol{S}_i^{final} = (1 - R(\Theta))\boldsymbol{S}_i^{\text{cond}} + R(\Theta)\boldsymbol{S}_i^{\text{cond,filtered}} \tag{3.37}$$

$R(\Theta)$ is a maximum of two seperate ramps, one based on the skewed coefficients in section *i*, and other on a low MU cutoff criterion for section *i*.

Because each individual source term $\boldsymbol{S}_i^{\text{cond}}$ is treated so that a stable system results in terms of the "draining of a section", the sum of these source terms also leads to a stable system. However, one additional problem remains with $S_{i+1}^{\text{cond,part}}$ and $S_{i+1}^{\text{cond,TMU}}$ source term additions in Eqn. (3.36). These additions may create skewness in section $i+1$. This problem is treated in the next section, where a tie-line calculation resulting in the splitting of these source term additions is used to resolve the instability.

3.4 Filter for Source Term Additions to a Section due to Condensation

Without any filter, for a DOF-2 method, it is observed that the a particle distribution, advancing in size due to condensation from the gas, will create sections, *i*, ahead of the main distribution where $b_i^H < 0$. The reason for this is that the condensation operator is trying to create a mean distribution within the section, $\overline{\text{MU}}_i$, which has the property, $\overline{\text{MU}}_i < \text{MU}_i^L$, where $\text{MU}_i^L = \Delta\mu/(2c_i^H)$ is the average monomer units per particle in the low mode of section *i*, by adding particles with a small MU to particle ratio relative to the bounds of the section. In order to accommodate this request, b_i^H

is driven to negative values. This problem may be solved by splitting the incoming particle stream, e.g., the last two source terms in Eqn. (3.36), into two parts, one with $\overline{\mathrm{MU}}_i = \mathrm{MU}_i^L$, and one with particles diverted into the lower section with either $\overline{\mathrm{MU}}_{i-1} = \mu_i$ or more conservatively $\overline{\mathrm{MU}}_{i-1} = \mathrm{MU}_{i-1}^H$. Note, this means that some of the particles that are moved upwards from section $i-1$ to section *i* by condensation are then put back into the original section $i-1$ due to this filter.

The math involved with this tie-line calculation is essentially equivalent to that used for Section 4.5 involving the coagulation operator. This filter is applied when $G > 0$ and $b_i^H \ll b_i^L$, or alternatively when $G < 0$ and $b_i^H \gg b_i^L$. Below, the $G > 0$ case is described. However, the etching case is completely analogous. Note, section *i* is defined as the destination section for the original source term, and the modifications to the condensation source term due to the drain adjustment described in the previous section are carried out before the adjustments to the source term described here are made.

Let $S_i^{cond,\,\mathrm{part}}$ be the source term for particles incoming into section *i* from section $i-1$ due to condensation. It's name will be shortened to S^{part} below. Let $S_i^{cond,\,\mathrm{TMU}}$ be the source term for total monomer units; it's name will be shortened to S^{TMU} below. Let MU^{cond} be the average MU per particle in that source term.

$$\mathrm{MU}^{\mathrm{cond}} = \frac{S^{\mathrm{TMU}}}{S^{part}} \text{ with the unstable condition that } \mathrm{MU}^{cond} < \mathrm{MU}_i^L.$$

Then, we will distribute this source term partly into the lower section, Section $i-1$, which is alternatively called *Lalt*, under some conditions to preserve positivity of the coefficients. These conditions are when b_i^H is below and b_i^L is above a threshold, and also, it turns out, when both b_i^H and b_i^L are below a threshold. Changing the names of S^{part} and S^{TMU} to $S_i^{part,\,\mathrm{orig}}$ and $S_i^{\mathrm{TMU,\,orig}}$, Eqn. (3.38) expresses the condition for conservation of particles and TMU in the partition.

$$\begin{aligned} S_i^{part,\,\mathrm{orig}} &= S_i^{part,\,\mathrm{new}} + S_{i-1}^{part,\,\mathrm{new}} \\ S_i^{\mathrm{TMU,\,orig}} &= S_i^{\mathrm{TMU,\,new}} + S_{i-1}^{\mathrm{TMU,\,new}} = \mathrm{MU}_i^L S_i^{part,\,\mathrm{new}} + \mu_{i-1}^* S_{i-1}^{part,\,\mathrm{new}} \end{aligned} \tag{3.38}$$

μ_{i-1}^* is the average monomer units per particle in the modified source term for section $i-1$. Several different methods for specifying μ_{i-1}^* were attempted. The method currently in use is to set μ_{i-1}^* equal to MU_{i-1}^H, because the skewness in section $i-1$ is also a concern.

Solving the 2x2 system in Eqn. (3.38):

$$
\begin{aligned}
S_i^{part,\,\text{new}} &= \frac{S_i^{\text{TMU, orig}} - S_i^{part,\,\text{orig}}\mu_{i-1}^{*}}{\text{MU}_i^{L} - \mu_{i-1}^{*}} \qquad & S_i^{\text{TMU, new}} &= S_i^{part,\,\text{new}}\text{MU}_i^{L} \\
S_{i-1}^{part,\,\text{new}} &= \frac{S_i^{part,\,\text{orig}}\text{MU}_i^{L} - S_i^{\text{TMU, orig}}}{\text{MU}_i^{L} - \mu_{i-1}^{*}} \qquad & S_{i-1}^{\text{TMU, new}} &= S_{i-1}^{part,\,\text{new}}\mu_{i-1}^{*}
\end{aligned}
\tag{3.39}
$$

Eqn. (3.39) is a simple tie-line condition specifying the stable formulation of the source term $\boldsymbol{S}^{new}$. A ramp condition described in section (2.14) is used to formulate a final source term that exhibits a continuous first derivative.

$$
\boldsymbol{S}^{final} = (1 - R(\Theta))\boldsymbol{S}^{old} + R(\Theta)\boldsymbol{S}^{new} \tag{3.40}
$$

$R(\Theta)$ is a maximum of two seperate ramps, one based on the skewed coefficients in section *i*, and other on a low MU cutoff criterion for section *i*.

A recap of what has been done to the original condensation source terms, Eqn. (3.26), is in order. First the full condensation operator is broken up into individual section contributions, *i*, where each sectional contribution includes the growth within that section, the transport out of that section, and the addition of into neighboring sections. Next, the sectional contribution is analyzed for skewness creation due to the draining of particles from section i. A solution, Eqn. (3.36), is found for this problem. However, additions to neighboring sections may also cause skewed distributions in the neighboring section. The source term for these additions must be filtered again based on the skewness of the neighboring section, possibly resulting in mass and particles being put back into the original section.

3.5 Special Treatment of the Top and Bottom Sectional Bins

The top section is not allowed to have volumetric growth condensation reactions, therefore preserving the principle of conservation of monomer unit and particles. Etching reactions involving the top section are allowed.

Etching condensation reactions applied to the first section bin require special treatment. Normally, for etching reactions, the condensation reaction would create a flux of particles from bin *L* to section bin *L-1*. However, this process breaks down for the section bin 0. Through various numerical experiments, an adequate treatment for this problem has been found to be that the surface reaction in section bin 0 should remain unchanged. However, when the monomer unit to particle ratio in section bin 0 has been reduced to the minimum permissible value, MU_0^{L}, then the

total particle number should be reduced in such a quantity that maintains conservation of monomer units, at the expense of the conservation of particles. The major effect of this is that it allows for etching surface growth reactions to fully depopulate a particle distribution, if given enough time. The implementation details will now be described.

Recalling that Eqns. (3.22) and (3.23) are the particle and total monomer conservation equations for the condensation operator. Then, the right-hand side of Eqn. (3.23), reproduced below as Eqn. (3.41), is called the volumetric etching TMU source term for section 0, $S_{V,0}^{\text{TMU}}$.

$$S_{V,0}^{\text{TMU}} = \int_{\mu_0}^{\mu_1} G(\mu, t) n(\mu, t) d\mu \tag{3.41}$$

This is the rhs of Eqn. (3.23). There usually is no volumetric etching particle term at all, because condensation reactions do not cause a change in the total number of particles. However, for the zereoth sectional bin a source term is created, when the monomer unit to particle ratio in section bin 0 is close to the minimum permissible value, MU_0^L

$$S_{V,0}^{\text{part}} = S_{V,0}^{\text{TMU}} \left(\frac{\max(R_{skew}(\Theta), R_{cutoff}(\Theta))}{\text{MU}_0^L} \right) \tag{3.42}$$

$R_{skew}(\Theta)$ and $R_{cutoff}(\Theta)$ are ramp defined by Eqn. (2.31), where:

$x = \dfrac{b_0^H}{b_0^L + b_0^H}$, and $x_a = 0.1$, $x_s = \dfrac{1}{30}$, for the skew ramp;

$x = \dfrac{b_0^L + b_0^H}{b_0^{cutoff}}$, and $x_a = 1.0$, $x_s = 0.3$, for the cutoff ramp.

In essence, what we have done is add a rhs to Eqn. (3.22) for section 0. When the ramp is fully activated, particles with total monomer units equal to MU_0^L, the lowest number of monomer units per particle allowed in the section, are being destroyed at a particle etch rate of $S_{V,0}^{\text{part}}$.

The discussion above pertained to the DOF-2 case. In the DOF-1 case, there is always etching from Section 0, unless it is specifically turned off. This means that there are normally particle losses in the condensation operator under etching conditions for the DOF-1 case.

3.6 Input Deck Options

The preceding sections have described the theory behind CAD's condensation kernel. Figure (3.3)

```
START BLOCK ADS MODEL DEFINITION
START BLOCK CONDENSATION MODEL
   ! Method = [ NONE | ADS ]
   !     (required) (default = ADS)
   !   Toggle switch to turn condensation on and off.
   Method = ADS
   !-------------------------------------------------------------
   !  Allow Bottom Section Etching = [boolean]
   !               optional ( default = true)
   !
   !   Allow for the possibility of etching of the bottom
   !   section, even if there is no bottom flux option.
   !   What this achieves it that it allows for a mechanism
   !   to destroy particles via the surface growth
   !   condensation operator.
   !   Basically, you would only want to have this false
   !   if you were checking for conservation properties, or
   !   if you had a better idea of what to do with the
   !   bottom flux.
   Allow Bottom Section Etching = true
   !-------------------------------------------------------------
   !  Section Addition Adjust Method = [ Conservative | Ramp |
   !                                     Always Accurate ]
   !   Method for filling in a section. This sets algorithm
   !   for determining the MU to particle ratios entering a
   !   section as a function of the distribution of
   !   particle ratios already in the section. There is a
   !   balance between accuracy and numerical stability
   !     Conservative = injected amount is always >MUL and
   !                    < MUH.
   !     Ramp         = Injected amount varies depending
   !                    upon the existing amounts.
   !                    (accurate when it can be, and conservative
   !                       when it needs to be) default
   !     Always Accurate = injected amounts are always
   !                       at v_i and v_ip1. Negative
   !                       coefficients are to be expected.
   !   (only applicable to DOF 2 implementations)
   Section Addition Adjust Method = Ramp
   !-------------------------------------------------------------
END BLOCK CONDENSATION MODEL
START BLOCK ADS MODEL DEFINITION
```

Figure 3.3 Input Options for the Condensation Operator

presents the block of the input file for `CADS` that controls the options for the conduction kernel.

The first option, `Method`, toggles on and off the condensation kernel in its entirety. Setting it to `NONE` turns off the kernel. Setting it to `ADS` turns on the kernel.

The next option, `Allow Bottom Section Etching`, handles the treatment of etching for the bottom section. If this option is set to false, etching is turned off for section 0. This has the effect of making the condensation operator conserve particles under etching conditions, at the expense of possibly piling up particles in section 0 under etching conditions.

The option, `Section Addition Adjust Method`, allows the user to adjust the filter for source term additions to a section. There are three options. The default option, `Ramp`, described in section (3.4) is to use a ramp to turn on the tie line calculation when necessary. The option, `Conservative`, implements a method where tie-line calculation is used always. The option, `Always Accurate`, turns off the tie-line calculation in all cases. Note, this option only affects DOF-2 implementations.

3.7 Sample Problem for Particle Growth via Condensation

In order to illustrate the condensation operator, a sample problem will be solved. The sample problem will be one with an analytical solution to Eqn. (3.1). The numerical result will be compared to the analytical result.

3.7.1 Analytical Example Problems for Pure Condensational Growth

The expression for growth of a particle is given by Eqn. (3.1), reproduced below as Eqn. (3.43).

$$\frac{\partial}{\partial t}n(\mu, t) + \frac{\partial}{\partial \mu}(G(\mu, t)n(\mu, t)) = 0 \tag{3.43}$$

Gelbard [27] has previously reported work in the literature [28] concerning an analytical solution to Eqn. (3.43), the solution derivation of which we will repeat here. Friedlander [2, p. 285] also provides analytical solutions for Eqn. (3.43) in special casec.

If the particle growth rate, $G(\mu, t)$, is restricted to the following form,

$$G(\mu, t) = H(t)f(\mu) \ , \tag{3.44}$$

the first order hyperbolic nature of Eqn. (3.43) may be exploited to obtain an initial solution of Eqn. 3.43 in short order.

$$\frac{\partial}{\partial t}n(\mu, t) + \frac{\partial}{\partial v}(H(t)f(\mu)n(\mu, t)) = 0 \tag{3.45}$$

$$\frac{1}{H(t)}\frac{\partial}{\partial t}n(\mu, t) + \frac{\partial}{\partial \mu}(f(\mu)n(\mu, t)) = 0 \tag{3.46}$$

Let

$$F(\mu) = \int \frac{d\mu}{f(\mu)} \text{ and } G(t) = \int H(t)dt \ . \tag{3.47}$$

Then,

$$\frac{\partial}{\partial G}(n(\mu, t)f(\mu)) + \frac{\partial}{\partial F}(f(\mu)n(\mu, t)) = 0 \tag{3.48}$$

And, by the method of characteristics, solutions must be of the following form:

$$\Im = P(F(\mu) - G(t) + G(0)) \ , \tag{3.49}$$

where $P(y)$ is an arbitrary function. Any function $P(y)$ will satisfy Eqn. (3.48). For satisfaction of the initial conditions, a functional form of $P(y) = (\Im)_0(F^{-1}(y))$ is used, where $(\Im)_0(\mu) = f(\mu)n(\mu, 0)$, which leads immediately to the satisfaction of the initial conditions. To conclude the solution is equal to Eqn. (3.50).

$$f(\mu)n(\mu, t) = (\Im)_0(F^{-1}(F(\mu) - G(t) + G(0))) \tag{3.50}$$

3.7.2 Example 1 Surface Reaction Limited Growth of Particles

Let's assume that the growth of spherical particles occurs via a direct surface reaction which doesn't depend upon the size of the particle. Let's also assume that there is an unlimited supply of reactant in the gas phase. Under these conditions, the growth rate of a spherical particle in # per second may be expressed as

$$G(\mu) = R(\pi d_\mu^2)A_v, \tag{3.51}$$

where d_μ is the diameter of a particle of total MU size μ. R is the surface reaction rate, given in units of kmol m^{-2} s^{-1}. A_v is Avogadro's number. If a spherical particle with a constant molar concentration of c_o is assumed, d_μ may be expressed in terms of μ.

$$d_\mu^2 = \left(\frac{6\mu}{\pi c_o A_v}\right)^{\frac{2}{3}} \tag{3.52}$$

These expressions may be written in the nomenclature of the previous section.

$$f(\mu) = a\mu^{\frac{2}{3}}, \text{ where } a = \pi R A_v \left(\frac{6}{\pi c_o A_v}\right)^{\frac{2}{3}} \tag{3.53}$$

Thus,

$$F(\mu) = \frac{3}{a}\mu^{\frac{1}{3}} \text{ and } F^{-1}(y) = \left(\frac{ay}{3}\right)^3 . \tag{3.54}$$

And, the solution is

$$f(\mu)n(\mu, t) = (\Im)_o\left(\left(\frac{a}{3}\left(\frac{3}{a}\mu^{\frac{1}{3}} - t\right)\right)^3\right). \tag{3.55}$$

Another way to look at Eqn. (3.55) is to note that $d \alpha \mu^{\frac{1}{3}}$, and thus the diameter of the distribution of particles should grow linearly in time.

Numerical Implementation

We first carry out a solution which has a constant growth rate per unit surface area, R. Then, $G(\mu) = R(\pi d_\mu^2)A_v$. The equation set is as follows. For each section, *i*:

$$\int_{\mu_i}^{\mu_{i+1}} \frac{\partial}{\partial t} n(\mu, t)d\mu + G(\mu_{i+1})n_+(\mu_{i+1}) - G(\mu_i)n_-(\mu_i) = 0 \tag{3.56}$$

and

$$\int_{\mu_i}^{\mu_{i+1}} \frac{\partial q}{\partial t} d\mu + G(\mu_{i+1})q_+(\mu_{i+1}) - G(\mu_i)q_-(\mu_i) = \int_{\mu_i}^{\mu_{i+1}} G(\mu, t)n(\mu, t)d\mu \tag{3.57}$$

The unknowns in the equation system are the coefficients for the distribution of TMU within each section. There are two of them corresponding to the value of q for μ_i and μ_{i+1} within each section.

A First Attempt at a Numerical Solution

To initialize the simulation, the number of sections is set to 10. The number of degrees of freedom per section is be set to 2. The lower boundary is be set to 100 MU, while the upper boundary is set

to 1.0×10^{10} MU. The monomer unit ID is set to C_2H_2, and the molar volume is set to 0.0255 m^3 (kmol MU)$^{-1}$.

It's necessary to have a gas phase ideal gas mixture in order to properly initialize the Section Model. We will use air with C_2H_2 added in order to ensure that the particle MU is also a gas-phase species. We will start with a relatively large number of particles, such that the particle number is roughly 1 part in 10^4 of the number of molecules. At room temperature and 300 K this requires a particle density value of 4×10^{-9} gmol cm^{-3} = 4×10^{-6} kmol m^{-3}. We will use a gaussian distribution of particles initially, centered at 60 MU with a width of 60. Thus

$$n(\mu, 0) = (4 \times 10^{-6} \text{ kmol m}^{-3}) \exp\left(-\frac{(\mu - 60)^2}{60^2}\right) \tag{3.58}$$

We will assume a growth rate that will double the average particle size in 10 seconds. When the surface area of a 60 MU particle (1.2×10^{-15} m^2) is factored in, this works out to be:

$$R = \frac{60 \text{ MU}}{(10 \text{ sec})(1.2 \times 10^{-15} \text{ m}^2)}\left(\frac{1 \text{ kmol MU}}{6.02 \times 10^{26} \text{ atoms}}\right) = 8.0 \times 10^{-12} \frac{\text{kmol MU}}{\text{m}^2\text{s}} \tag{3.59}$$

$$R = \frac{60 \text{ MU}}{(10 \text{ sec})(1.2 \times 10^{-15} \text{ m}^2)} = 5.0 \times 10^{15} \frac{\text{MU}}{\text{m}^2\text{s}} \tag{3.60}$$

Note, the correct units for inclusion into the program is the second one, Eqn. (3.60), because the growth rate per particle is tracked on a direct MU units basis, not on a kmol MU units basis. Thus, because we track particle sizes in direct MU basis, see Eqn. (3.43), we have to track particle growth on a direct MU basis as well.

3.7.3 Results

Figure 3.4 illustrates the CADS solution vs. analytical solution. The black lines are the analytical results. Particles with an initial gaussian distribution grow by a factor of 10 in monomer unit size. The red curve is the result from the DOF-2 discontinuous Galerkin method using a geometric grid spacing of 2. The blue curve is from the DOF-1 method using a geometric spacing of 1.5. Note, both methods conserve particle numbers up to round-off error. The DOF-1 method exhibits a great deal of numerical diffusion, which leads to a spreading out of the distribution both on the high and low side, and a great deal of underestimation of the peak height. The discontinuous galkerin treatment did exceedingly well at reproducing the analytical distribution. Note, the analytical result does have a discontinuity at the tail, where the distribution falls discontinuously to zero. This is due

to the cutoff of the initial distribution at the bottom section at $t = 0$. Thus, the smallest particle gets bigger throughout the run. In this sample problem, there is no production of particles via a nucleation mechanism. Therefore, the discontinuity in particle concentration is "advected" to higher MU's along a characteristic.

Figure 3.5 provides results for the average diameter predicted by two methods as a function of time for various grid spacings. Results are all fairly clustered around each other except for the DOF1 method with the widest sectional spacing of 1.095.

Details of the resolution of the two methods as a function of grid spacing are now addressed. Figure 3.6 provides results for the DOF-2 discontinuous Galerkin method as a function of the geometric section spacing. The 1.2 spacing results contain roughly 14 degrees of freedom in the MU region where the particle density is significant. Even with only 14 unknowns the distribution of particles is fairly close to the analytical result. Each of the further refinements, the 1.1 and 1.05 spacings, roughly doubles the degrees of freedom in the significant particle density region. The agreement with the analytical results gets progressively better on refinement. Note the hardest part to achieve success on is the C0 discontinuity at the tail end of the distribution. Even here the refinement in grid spacing is seen to promote agreement with the analytical results.

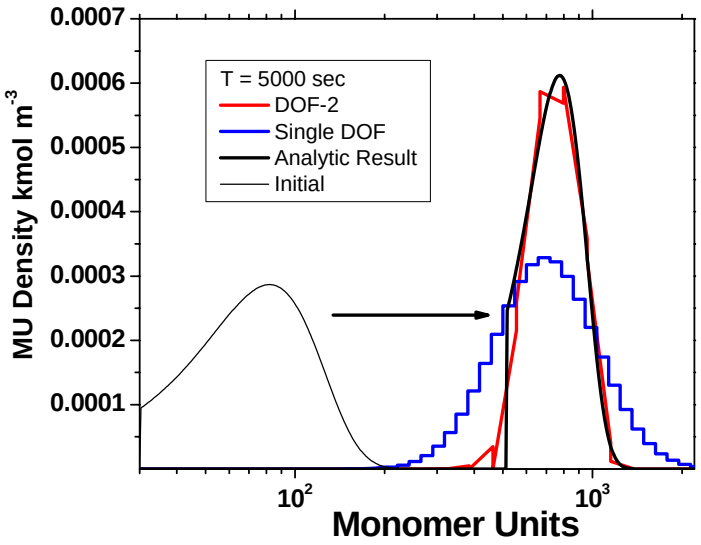

Figure 3.4 MU density of particle distribution versus MU per particle for pure condensation. Growth rate is proportional to the surface area, with spherical particles assumed. An initial gaussian profile (black) is used.

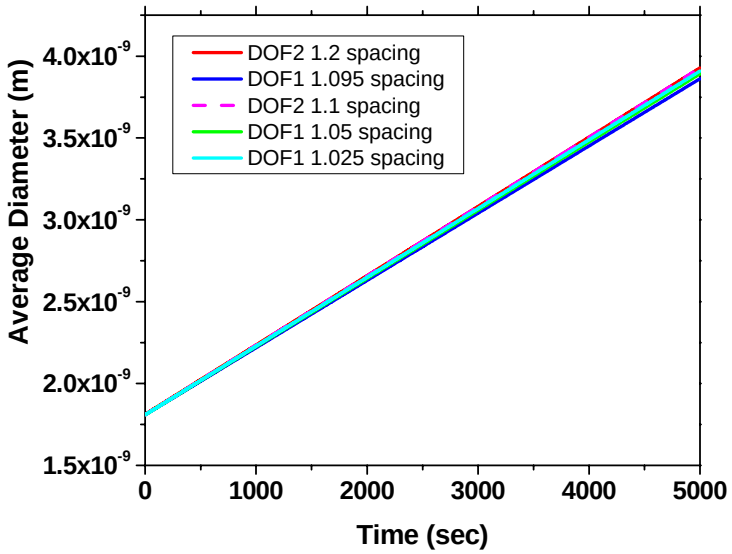

Figure 3.5 Average diameter predicted by the DOF 2 discontinuous Galerkin and Warren/Seinfeld DOF1 methods as a function of time for various section spacings.

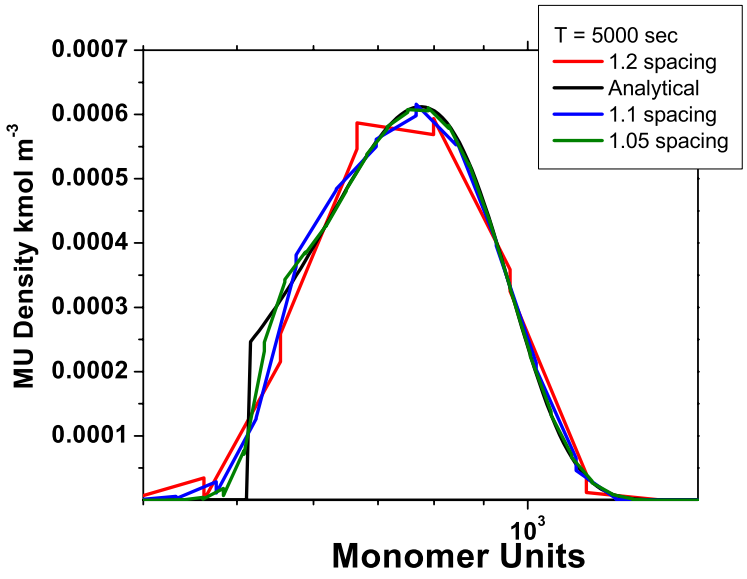

Figure 3.6 Convergence test for discontinuous Galerkin method using two degrees of freedom. Shown is an enlargement of the monomer unit region with nonzero particule density.

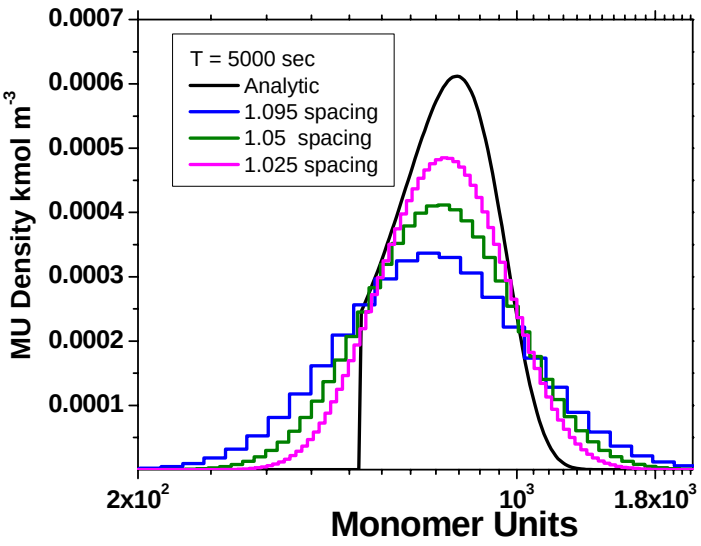

Figure 3.7 Convergence test for DOF-1 method. Shown is an enlargement of the monomer unit region with nonzero particule density.

Figure 3.7 provides refinement results for the DOF1 method, which also preserves particle numbers. Each of the three curves have roughly the same number of unknowns as the curves if Figure 3.6. However, the agreement with the analytical results isn't as good. Figure 3.7 does show that the method is converging towards the analytical result as the grid is refined. Of particular concern are the high and low tails to the distribution, which can cause anomolous behavior, especially if light scattering intensity, which has a d^6 dependence, is an important observable.

Despite the numerical diffusion, there are some benefits to using the Warren and Seinfeld method DOF1. The method is a faster than the discontinuous Galerkin method. For example the 1.025 spacing case took 525 seconds (-g option) and 2662 time steps, while the discontinuous Galerkin method with 1.05 spacing took 6099 time steps and 2348 seconds. Also, there were several time step truncation error failures using the discontinuous Galerkin method. On checking why these occurred, I observed they were due to the sudden onset of the filtering algorithms in a section. There may be more work necessary in the future to fully characterize the filtering algorithm's numerical behavior in order to smooth transitions.

Figure 3.8 compares the raw results without the filtering technique to the filtered result for the case of 1.2 geometric spacing. Inserts in the graph detail the leading and trailing edge of the evolving particle distribution. Figure 3.8 demonstrates that the application of the filters has a rather large effect on the solution quality. In particular, the "draining filter" has a large effect. Without the

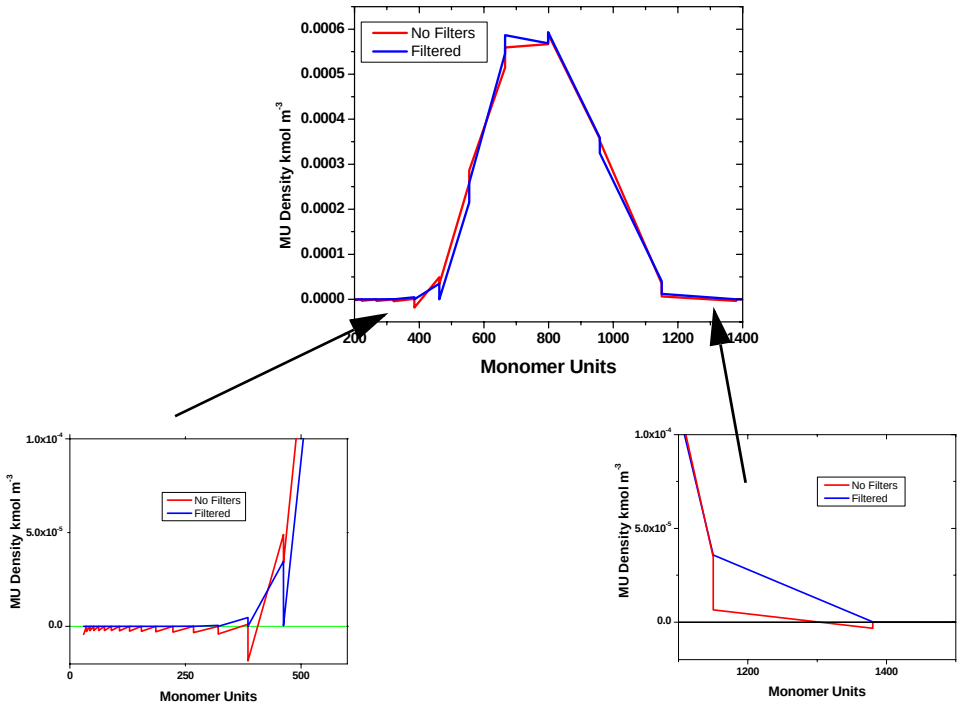

Figure 3.8 Details of the effects of the filters for the 1.2 spacing case.

"draining filter", there remains a small but nonneglible negative particle number and mass left behind in each section that has been drained. Surprising also is the difference between the filtered and the nonfiltered results at the top of the distribution. The filtered results are closer to the analytical results. The application of the filters on the leading and trailing edge of the distribution has altered the numerical results everywhere. The net mass accummulation, however, has not changed very much (0.2465 kmol MU m^{-3} for the unfiltered case and 0.245498 kmol MU m^{-3} for the filtered case).

At longer times for this problem, the same trends get stronger. Figure 3.9 shows the results for the problem at later times where the analytical solution largely resides within one section. The DOF 2 1.2 spacing demonstrates adequate performance. The distribution of particle density is spread out over roughly 5 sections, while the analytic solution is roughly one third of a section in width. The Warren and Seinfeld method (DOF 1) with roughly the same density of degrees of freedom shows inferior performance. Figure 3.10 shows the average particle diameter versus time. The average particle diameter of the predicted distribution for the DOF 2 1.2 spacing case is indestingishable from the analytical result. The DOF 1 case, however, exhibits a net decrease in the average diameter from the analytical result, demonstrating that the large amount of numerical diffusivity is having a impact on integrated quantities.

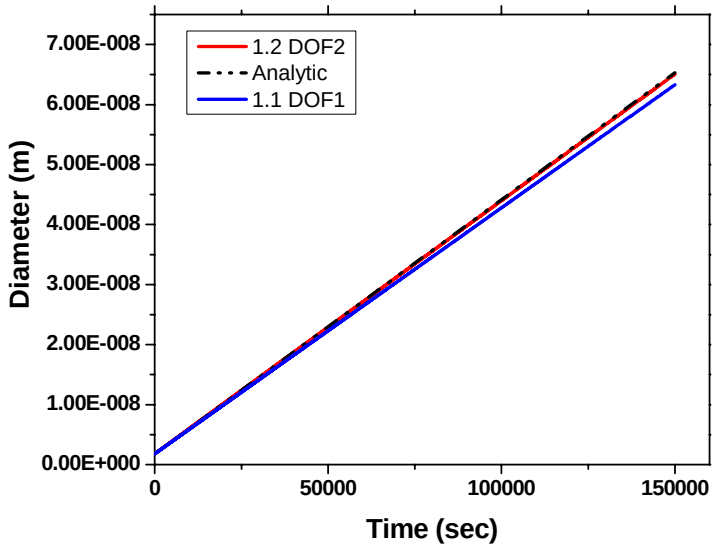

Figure 3.10 Average diameter versus time for various cases.

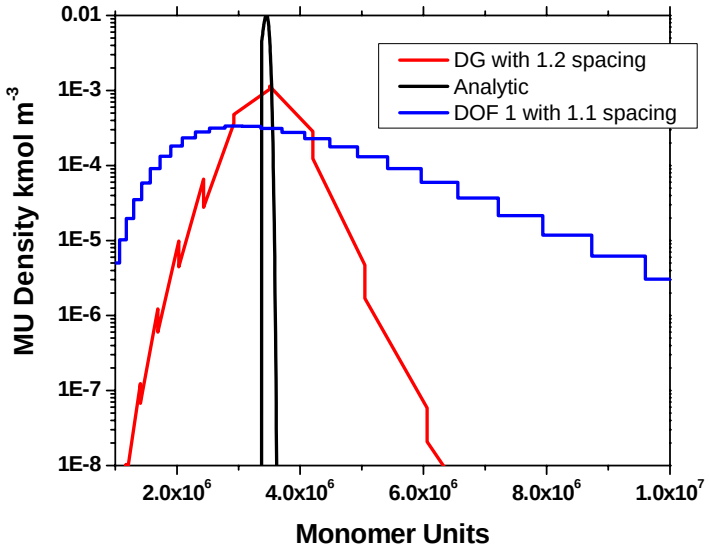

Figure 3.9 Distributions at a longer time, t = 150000 sec.

4. Coagulation

Coagulation is the process wherein two particles collide and create a single larger particle. A given size particle will grow to a larger size by an effective collision with another particle or monomer. An effective collision is one which results in the formation of a new particle composed of the two colliding particles. All collisions are not necessarily effective. The effectiveness will depend upon the size and composition of the particles, as well as other environmental factors. The coagulation operator is distinct from the condensation operator in the sense that the coagulation operator involves two particle reactants. The condensation operator involves just one particle reactant and zero, one or possibly more molecular reactants. The coagulation operator is very similar to bimolecular reactions. It will be shown that the for small particle sizes of 10 nm or so, the coagulation kernel is the rate limiting step for evolution of the particle distribution and therefore controls the time step that may be taken for accurately integrating the particle distribution equations.

The evolution of a distribution of particles, which has concentrations $n(\mu, t)$ kmol particle per m^3 per monomer unit (MU) of size μ monomer units at time *t*, due to coagulation, may be described by retaining only three of the terms in Eqn. (1.1) to give an integral-differential equation, Eqn. (4.1).

$$\frac{d}{dt}n(\mu) = \frac{1}{2}\int_0^{\mu} \beta(\tilde{\mu}, \mu - \tilde{\mu})n(\tilde{\mu})n(\mu - \tilde{\mu})d\tilde{\mu} - \int_0^{\infty} \beta(\mu, \tilde{\mu})n(\tilde{\mu})n(\mu)d\tilde{\mu} \qquad (4.1)$$

$\beta(\mu, \mu)$ is the coagulation kernel giving the collision frequency (units of m^3 (kmol particles)$^{-1}$ sec^{-1}), between two particles of MU size μ and size μ. The first term on the RHS of Eqn. (4.1) stems from two particles of lower size forming a particle of size μ. The one-half is necessary in order to not over count collisions. The second term on the RHS of Eqn. (4.1) is due to collision of particles of size μ colliding with other particles to form particles of a larger size.

The coagulation kernel has one form for the free molecule particle regime, where particles “see” the gas as individual molecules, and a different form for the continuum particle regime, where the particles “see” the gas as a continuum. In our application, a model that interpolates between free molecule and continuum and exhibiting the correct limits in both cases has been implemented.

Integrating Eqn. (4.1) over all particle total monomer unit sizes, μ, we obtain an equation for the total particle concentration, N_T, having units of kmol particle m^{-3}, as a function of time,

$$\frac{dN_T}{dt} = -\frac{1}{2}\int_0^{\infty}\int_0^{\infty} \beta(\mu, \tilde{\mu})n(\mu)n(\tilde{\mu})\, d\mu d\tilde{\mu}\,. \qquad (4.2)$$

At the small sizes of newly formed soot, our first target application, it appears that particles are sufficiently fluid-like that clusters, formed via coagulation, will flow into a spherical particle [30]. At larger sizes (and cooler temperatures than flamelet temperatures) this is not true. Particles that form at larger sizes may appear to be a fractal agglomeration of smaller primary particles. It has been widely observed that soot particles in fire consist of agglomerates of primary spherical particles. Collision frequencies of fractal particles are different from those for spherical particles of the same total volume. The model for coalescence initially used here does not account for the morphology of soot particles; particles are assumed to be spherical. However, a fractal dimension could very easily be added to the program, by adding an extra degree of freedom per particle section, to account for the change in the collision properties due to a change in the morphology of a soot particle. This has been done in some other particle modeling efforts.

In this chapter, the coagulation kernel $\beta(\mu, \mu)$ is presented. Then, the method for evaluating the integrals within CADS is described. In the last section, a couple of verification problems are solved.

4.1 Coagulation Kernel due to Brownian and/or Molecular Motion

CADS has adopted Fuch's expression [31] for the collision frequency between two particles, because it is applicable for particles ranging from millimeter to molecular sizes. This expression is interpolates between the free molecular and continuum-limiting regimes.

$$\beta(\mu_i, \mu_j) = 2A_v \frac{\pi(d_i + d_j)(D_i^b + D_j^b)}{\dfrac{(d_i + d_j)}{(d_i + d_j) + 2\sqrt{\delta_i^2 + \delta_j^2}} + \dfrac{8(D_i^b + D_j^b)}{(d_i + d_j)\sqrt{G_i^2 + G_j^2}}} \tag{4.3}$$

where

$\beta(\mu_i, \mu_j)$ = Collision frequency function between particles in MKS units of m^3 $kmol^{-1}$ sec^{-1}

A_v = Avogadro's number

d_i = Diameter of particle *i*

D_i^b = Brownian diffusivity of particle *i*

δ_i = Characteristic separation distance for particle *i*

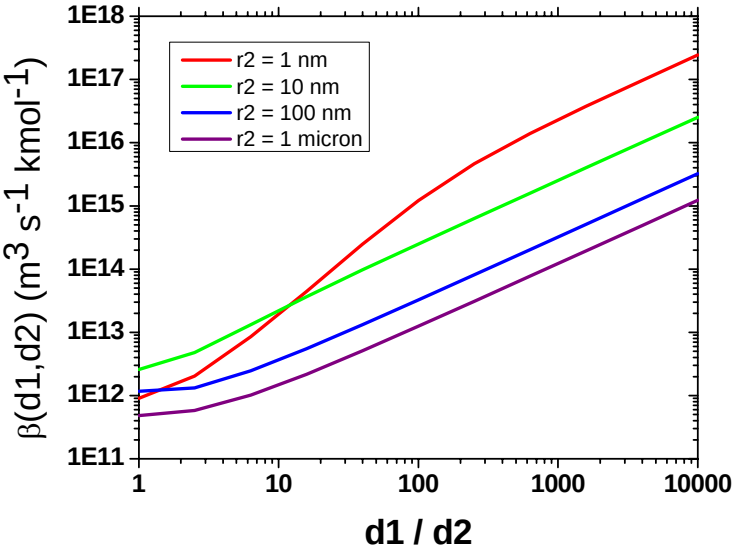

Figure 4.1 Variation of the collision frequency function, $\beta(d_1, d_2)$, with the particle size ratio, d_1 / d_2, for N_2 at 400 C and 1 atm based on the Fuchs formula. The value of $\beta(d_1, d_2)$ is largest for particles of very differing sizes.

$G_i = \left(\frac{8kT}{\pi m_i}\right)^{1/2}$ = Mean thermal velocity for particle *i* with mass m_i

$\beta(d_1, d_2)$ has units of $m^3\ s^{-1}\ kmol^{-1}$. If multiplied by two particle concentrations of units kmol m^{-3}, it produces a rate of production of kmol $m^{-3}\ s^{-1}$, which is the nominal units of gas-phase rates of production from gas phase reactions. The Fuch's expression accounts for particle size, and it may be used with distinct multicomponent particles, if a relation for the size of the particle as a function of its composition is provided. Collisions may be thought of as a chemical reaction with the collision frequency function as the reaction rate. The rate of formation of $i + j$ by reaction of *i* with *j* by collision is then given by the rate of their collision. The efficiency of collision is assumed to be equal to one, in this initial CADS implementation, even though there is some evidence that collisions between very small particles with near-molecular diameters have non-unity collision efficiencies. Implementation of a non-unity collision efficiency model in future editions of CADS is a relatively small modular project, due to the object-oriented approach.

The Brownian diffusivity of particle *i* may be given by the following relation, Eqn. (4.4) [32].

$$D_i^b = \frac{kT}{f}, \text{ where } f = \frac{3\pi\mu d_i}{C_i} \text{ is the Knudsen-Weber interpolating friction factor} \qquad (4.4)$$

C_i is the Cunningham correction factor to Stokes law, used to account for slip-flow on the particle surfaces, and is dependent on the particle Knudsen number. C_i is given by the following formula [2, p. 34], obtained originally by a fit to experimental data:

$$C_i = 1 + (Kn)\left(A_1 + A_2 \exp\left[-\frac{A_3}{Kn}\right]\right). \qquad (4.5)$$

$$A_1 = 1.257,\ A_2 = 0.400, \text{ and } A_3 = 1.10$$

where

$$Kn = 2\frac{\lambda}{d_i} \qquad (4.6)$$

$$\lambda = \frac{1}{\sqrt{2}\pi n \sigma^2} = \frac{2\mu}{\rho}\sqrt{\frac{\pi \overline{M}}{8RT}} \qquad (4.7)$$

$\overline{M}$ is the mean molecular weight. The equation for C_i provides an interpolation formula for the entire range of Knudsen number. λ is the mean free path in the gas. In Eqn. (4.7) n is the gas concentration and σ is the collisional cross section of gas phase molecules.Note the hard sphere result from Bird, Stewart, and Lightfoot [51], for example, has a 3 instead of the 2 next to μ in Eqn. (4.7). However, the result for real gases is much more complicated. The actual answer is that the number has to be consistent with the accommodation coefficient, α, (a quantity appearing in the molecular limit which represents the fraction of molecules which leaves the particle surface in equilibrium) such that the drag force on the particle in the free molecular and continuum limits is given by the correct limiting expression. Using Eqn. (4.7), and the expression for the friction coefficient that is obtained in the molecular limit, [2, p. 33],

$$f = \frac{2}{3}d^2\rho\left(\frac{2\pi RT}{\overline{M}}\right)^{\frac{1}{2}}\left[1 + \frac{\pi\alpha}{8}\right], \qquad (4.8)$$

Eqns. (4.4), (4.5), (4.7), and (4.8) may be solved for in the molecular limit ($Kn \to \infty$) to determine the consistent value of α It turns out to be $\alpha = 0.907$. However, no adequate theories exist for specification of α. Therefore, α is assumed to be the constant 0.907 within the CADS package.

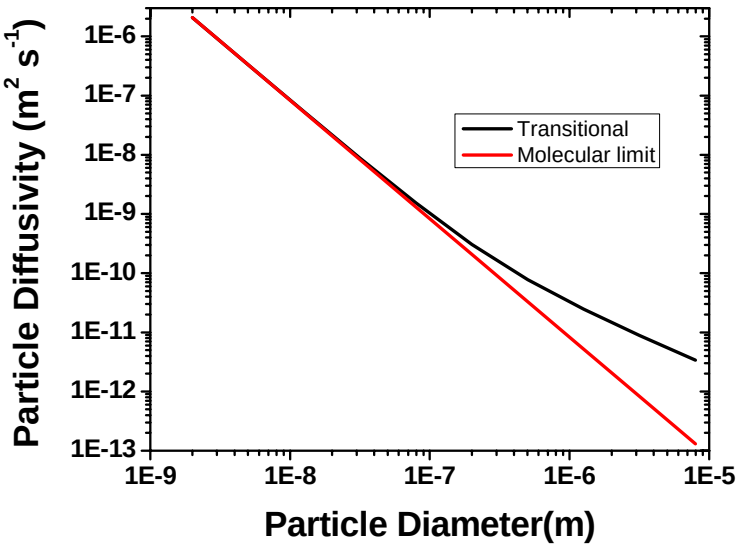

Figure 4.2 Particle Brownian diffusion coefficients as a function of particle diameter. 400 K, 1 atm, in an N_2 ambient. Black line is the full formula; red line is the formula from the molecular limiting case, assuming an accommodation coefficient of 0.91.

Figure (4.2) displays the particle diffusivity vs. particle diameter result for the 2 cases of the full expression, Eqn. (4.4), vs. just the molecular kinetic theory limit, Eqn. (4.8). The kinetic theory expression was calculated assuming a 0.91 accommodation coefficient. The bath gas was assumed to be 400 K N_2 at 1 atm. The full expression, Eqn. (4.4), is seen to be consistent with the molecular limiting theory at small particle sizes. Moreover, if the curve is extrapolated to molecular dimensions, then predicted diffusivities are roughly equal to the gas-phase molecular counterparts (0.5×10^{-4} m^2 s^{-1}). At large particle sizes, Eqn. (4.4) yields a much larger value of the diffusion coefficient than the molecular limiting expression.

Eqn. (4.3) may be simplified and compared to other formulations in the case of its two limits, molecular collisions and continuum collisions. In the continuum limit, the expression on the LHS of the denominator in Eqn. (4.3) dominates, and the expression reduces to Eqn. (4.9), where the expression for the particle radius, $r_i = d_i/2$, has been used instead of the particle diameter.

$$\beta(\mu_i, \mu_j) = 4\pi A_v((r_i + r_j) + \sqrt{\delta_i^2 + \delta_j^2})(D_i^b + D_j^b) \tag{4.9}$$

Without the $\delta_{ij} = \sqrt{\delta_i^2 + \delta_j^2}$ term, this expression is exactly the expression that may be derived for the continuum limit, which assumes Brownian-motion diffusion-limited coagulation [2, p. 190].

The characteristic separation distance for particle i, δ_i, is a concept originated by Fuchs to marry the continuum coagulation description with a molecular description. Essentially, around each particle, Fuchs envisioned a distance within which, particles would move according to the kinetic theory of gasses as if in a vaccum.Outside that region, which for particle-particle interactions is equal to $(r_i + r_j) + \delta_{ij}$, particle-particle interactions behave in the continuum limit. Equating the fluxes at the boundaries of the two regions results in a transition law rule, Eqn. (4.3).

The derivation for δ_i is reported in [32, p. 162] and the results are conveyed below. Define an apparent mean free path, L_i, for the particle as

$$L_i = \frac{8D_i^b}{\pi G_i}.$$

Then, Fuchs derives the following relation, Eqn. (4.10), for the separation distance.

$$\delta_i = \frac{1}{3d_iL_i}\left((d_i + L_i)^3 - (d_i^2 + L_i^2)^{\frac{3}{2}}\right) - d_i \tag{4.10}$$

For T = 400 K and 1 atm, Fig. (4.3) plots the ratio of the separation distance to the diameter as a function of the particle diameter. Also shown on the plot is the Knudsen number. It's readily seen

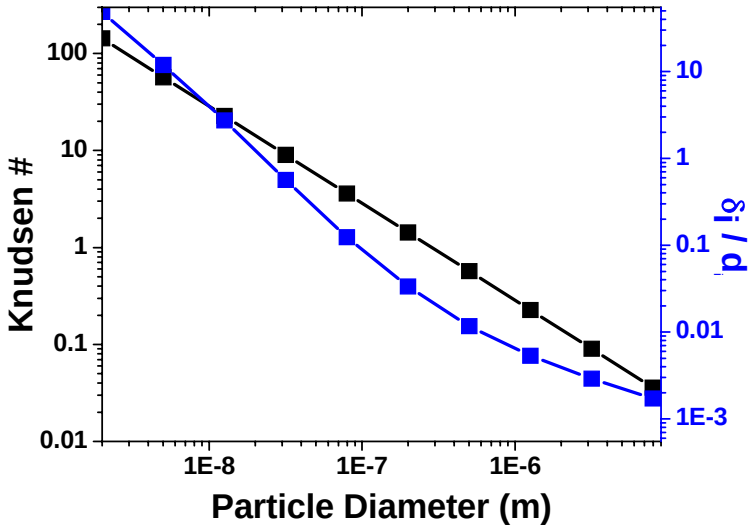

Figure 4.3 Ratio of the separation distance, δ_i, to the diameters as a function of the particle diameter, at 400 K and 1 atm, is in blue. Black line is *Kn* vs. particle diameter.

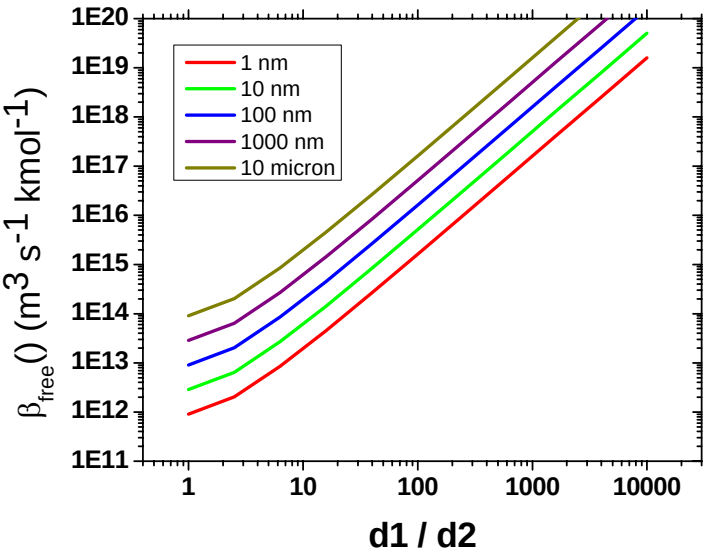

Figure 4.4 Variation of the collision frequency function, $\beta(\mu_i, \mu_j)$, with the particle size ratio d_1/d_2 for N_2 at 400 C and 1 atm based on the free molecular formula.

that the separation distance is a small fraction of the particle diameter except for small particles of roughly 20 nm or less. The separation distance itself is fairly constant with respect to the particle diameter having a value of about 10 nm. Therefore, in the continuum limit, the separation distance, δ_{ij}, in Eqn. (4.9) is effectively zero.

The molecular limit for Eqn. (4.3) may derived by dropping the LHS of the denominator. This results in Eqn. (4.11).

$$\beta(\mu_i, \mu_j) = \frac{\pi}{4} A_v (d_i + d_j)^2 \sqrt{G_i^{\,2} + G_j^{\,2}} \tag{4.11}$$

Eqn. (4.11) may be rewritten in terms of the volume of each particle, V_i and V_j, assuming spherical particles, and the particle density, $\rho_{\text{bulk\_particle}}$, as Eqn. (4.12).

$$\beta_{\text{free}}(\mu_i, \mu_j) = A_v \left(\frac{3}{4\pi}\right)^{\frac{1}{6}} \left(\frac{6kT}{\rho_{\text{bulk\_particle}}}\right)^{\frac{1}{2}} \left(V_i^{\frac{1}{3}} + V_j^{\frac{1}{3}}\right)^2 \sqrt{\frac{1}{V_i} + \frac{1}{V_j}} \tag{4.12}$$

Eqn. (4.12) agrees with Eqn. 7.17 on ref. 2, p. 192, and Eqn. 4.36 from ref. 32, p. 160. The values of $\beta(\mu_i, \mu_j)$ for the free molecular flow collision integral are plotted in Figure (4.4). When both particles have low radii the free molecular formula is the same as the Fuch's formula. For larger

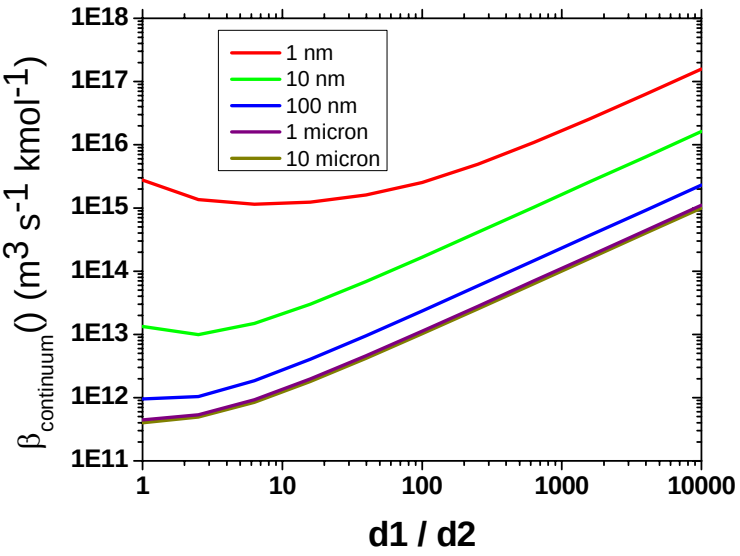

Figure 4.5 Variation of the collision frequency function, $\beta(\mu_i, \mu_j)$, with the particle size ratio d_1/d_2 for N_2 at 400 C and 1 atm based on the continuum molecular formula, Eqn. (4.9).

molecules the formula yields a higher collision rate than the Fuchs formula, because the free molecular formula doesn't take into account of the diffusional mass-transport resistance that occurs in the continuum regime. Figure (4.5) contains a plot of the collision frequency in the continuum limit, Eqn. (4.9).

4.2 Evaluating The Integrals in the Coagulation Kernel

In this section, the method by which CADS handles the calculation of Eqn. (4.1) after binning the particles into sections is described. The evaluation of Eqn. (4.1) turns into a double integral over particle source bins *A* and *B*, with complicated integral limits, for the production of particles in source bin *L*. Within each section, particle concentrations may have distributions of densities which must be accounted for.

The chief complication in the calculation of coagulation integrals is to codify all of the possible boundary condition cases that arise after the particle size distributions are binned into sections. The approach is based on cataloging all of the section bin to section bin interactions that produce an agglomerated particle in product section *L*. Particles in section *L* have MU values between μ_L and μ_{L+1}. The loop over section *L* will be the outer loop in the total agglomeration source term, the RHS of Eqn. (4.1). Section *A* is the section containing the larger of the two particles involved in

the agglomeration reaction producing a section *L* particle. Section *B* is the section containing the lower MU particle. Both *A* and *B* must be equal to or lower than *L*.

First, the lowest and highest sections for particle *A* which could possibly contribute an agglomeration source term for section *L* will be found. The lowest *A* section bound, section *SA_LB* (section *A* - lower bound), contributing to the *L* source term is the section for which coagulation with itself would yield a section L particle. This section is determined from the relation.

$$\mu_{SA\_LB} \leq \frac{\mu_L}{2} < \mu_{SA\_LB+1} \tag{4.13}$$

The highest *A* section that contributes an agglomeration source term, *SA_UB* (section *A* - upper bound) is section *L* in most cases,

$$SA\_UB = L.$$

Then, a loop is formulated over the section *A*, the section containing the bigger particle, from *SA_LB* to *SA_UB*. We look for interactions between section *A* particles and another section containing the smaller section *B* particle such that the resulting particle lies within section *L*. First, for a given section *A* with lower and upper *MU* section bounds of μ_{SA} and μ_{SA+1}, we find *SB_LB* and *SB_UB*, the lowest and highest values of section *B* such that an agglomeration reaction producing a section *L* particle can occur.

Let's first calculate *SB_LB*. μ_{B_LB} is the lower bound for the section *B* particle. Section *SB_LB* may be found from the following relation,

$$\mu_{SB\_LB} \leq \mu_L - \mu_{SA\_1} < \mu_{SB\_LB+1}. \tag{4.14}$$

Then, we find the upper section *B* bound by using the relation,

$$\mu_{SB\_UB} < \mu_{L+1} - \mu_{SA\_LB} \leq \mu_{SB\_UB+1}. \tag{4.15}$$

to find *SB_UB*. Note that *SB_UB* may well be equal to *L* in some coarse mesh cases where $2\mu_L < \mu_{L+1}$. This would involve *L*-*L* agglomeration cases where the resulting particle is also put into bin *L*. The mass within bin *L* remains the same. However, the total number of particles is reduced by one. This loss of particles must also be accounted for in the discretization scheme; this topic will be addressed later in the chapter.

Now, for a given section *A*, we may then loop over each section *B* from sections *SB_LB* to *SB_UB* in order to calculate the collision integrals. For each section *A* - section *B* interaction producing a

particle in section *L*, there are three coagulation integrals that conceptually need to be evaluated, corresponding to the rate of particle loss from each section, the rate of section *A* loss of MU's and the rate of section *B* loss of *MU*'s. The three integrals, which represent different weighted averages of the collision rate between a particles of size μ_A and μ_B are presented in Eqns. (4.16 - 4.19).

$$S_{A,B}^{L,\,p\_\text{loss}} = \int_{\mu_{B\_L}}^{\mu_{B\_U}} \int_{\mu_{A\_L}(\mu_B)}^{\mu_{A\_U}(\mu_B)} \frac{q(\mu_A)}{\mu_A}\frac{q(\mu_B)}{\mu_B}\beta(\mu_A, \mu_B)\; d\mu_A\; d\mu_B \tag{4.16}$$

$$S_{A,B}^{L,\,A\_\text{loss}} = \int_{\mu_{B\_L}}^{\mu_{B\_U}} \int_{\mu_{A\_L}(\mu_B)}^{\mu_{A\_U}(\mu_B)} q(\mu_A)\frac{q(\mu_B)}{\mu_B}\beta(\mu_A, \mu_B)\; d\mu_A\; d\mu_B \tag{4.17}$$

$$S_{A,B}^{L,\,B\_\text{loss}} = \int_{\mu_{B\_L}}^{\mu_{B\_U}} \int_{\mu_{A\_L}(\mu_B)}^{\mu_{A\_U}(\mu_B)} \frac{q(\mu_A)}{\mu_A} q(\mu_B)\beta(\mu_A, \mu_B)\; d\mu_A\; d\mu_B \tag{4.18}$$

$$S_{A,B}^{L,\,L\_\text{gain}} = \beta_{SA,SB}^{L,\,A\_\text{loss}} + \beta_{SA,SB}^{L,\,B\_\text{loss}} \tag{4.19}$$

Note, the outer integral is carried out over section *B.* This creates fewer cases where the integral must be broken up into 2 sections. The types of functions to be used in the inner integral over section *A* can now be cataloged.

The lower value of the inner integrand is equal to $\mu_{A\_L}(\mu_B) = MAX(\mu_L - \mu_B, \mu_{SA})$. If the MAX function switch modes because of the integrand, then the integrand must be broken up in order to avoid severe numerical inaccuracies. The upper value of the inner integrand is equal to $\mu_{A\_U}(\mu_B) = MIN(\mu_{L+1} - \mu_B, \mu_{SA+1})$.

The differences in the inner integrand boundary conditions are cataloged via the simple format represented in Figure (4.6) The types clearly depend on where the integration bounds intersect the corners of the section *A* - section *B* limiting rectangle. Thus, there are 16 possible interaction types. Not all types are possible. For example, it's not possible to have an upper bound of type 3 combined with a lower bound of type 2 or 3. Integrals involving type 1 will have to be broken up into separate integrals in order maintain a simple trapezoidal shape (i.e., one trapezoid with parallel μ_B = constant sides). If both the upper limit and the lower limit are both of type 1, then the integral is broken up into three integrals. This is the most complicated case.

Note, the factor of 1/2 in Eqn. (4.1) is taken care of automatically when section *A* is not the same as section *B*. In other words, because section A is always larger than section *B*, we are explicitly

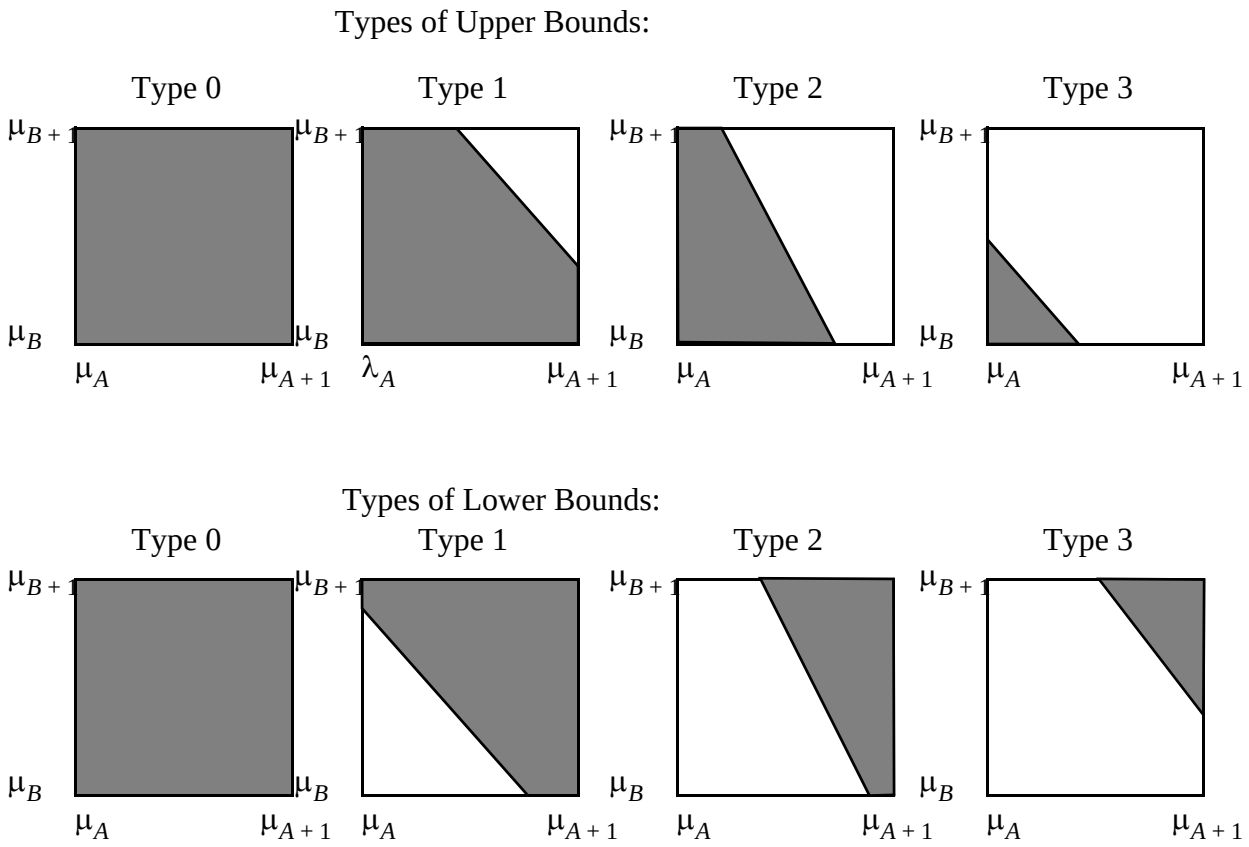

Figure 4.6 Catalog of all of the types of lower and upper bounds possible for the section *A* - section *B* integral.

not double counting collisions between section *A* - section *B* particles. When section *B* is in fact section *A*, we must maintain the principle that there is one particle which is always smaller than the other one within the collision integrals, Eqn. (4.16) to (4.18). This could be done by having the integral area along the $\mu_A = \mu_B$ line. However, I chose to accomplish the same thing by just multiplying the section *A* - section *A* integrals flux terms by 1/2.

Now that the complicated boundary condition cases are broken up into simpler boundary condition cases, where the loop over the bounds for μ_A involves only a single straight line, the double integral in Eqn. (4.16) - (4.18) may be solved by a general strategy. Romberg integration formulas have been used [33], that use Richardson extrapolation to generate limiting convergence bounds.

The entire coagulation procedure is kept within a dynamically allocated tree structure. With the current implementation of coagulation, there is no geometric constraint on section sizes. Therefore,

understanding and codifying the effects of numerical diffusion on particle size predictions may be done using a "mesh convergence" procedure. Mesh convergence here means that the size of sections get reduced to approximately the continuum limit.

4.3 Precalculation of Coagulation Integrals

All of the integrals presented in the previous section can be precomputed, and their values stored at the cost of a moderate amount of storage. This is true even for discretization schemes with multiple degrees of freedom in each section. We may formulate an algorithm for precalculation of the coagulation integrals, leading to a dramatic savings in computer time. As long as $\beta(\mu_A, \mu_B)$ stays constant or at least weakly varying, precalculation is a winning strategy.

The first step is to expand the basis functions in the integrals in Eqns. (4.16) to (4.18).

$$S_{SA,\,SB}^{L,\,p\_loss} = \int_{\mu_{B\_L}}^{\mu_{B\_U}} \int_{\mu_{A\_L}(\mu_B)}^{\mu_{A\_U}(\mu_B)} \frac{q(\mu_A)}{\mu_A}\frac{q(\mu_B)}{\mu_B}\beta(\mu_A, \mu_B)\, d\mu_A\, d\mu_B$$

$$= \int_{\mu_{B\_L}}^{\mu_{B\_U}} \int_{\mu_{A\_L}(\mu_B)}^{\mu_{A\_U}(\mu_B)} \frac{(q_L^A \phi_L^A(\mu_A) + q_H^A \phi_H^A(\mu_A))}{\mu_A}\frac{(q_L^B \phi_L^B(\mu_B) + q_H^B \phi_H^B(\mu_B))}{\mu_B}\beta(\mu_A, \mu_B)\, d\mu_A\, d\mu_B \quad (4.20)$$

$$= \beta_{A\_L,\,B\_L}^{L,\,p\_loss} b_A^L b_B^L + \beta_{A\_H,\,B\_L}^{L,\,p\_loss} b_A^H b_B^L +$$
$$\beta_{A\_L,\,B\_H}^{L,\,p\_loss} b_A^L b_B^H + \beta_{A\_H,\,B\_H}^{L,\,p\_loss} b_A^H b_B^H$$

The 4 coefficients, $\beta_{A\_L,\,B\_L}^{L,\,p\_loss}, \ldots, \beta_{A\_H,\,B\_H}^{L,\,p\_loss}$, for each section *A*-*B* interaction, for the particle loss term, may be precomputed and stored in the dynamically allocated linked tree. Then, the coagulation integrals may be cheaply determined via a 4 term quadratic expression described in Eqn. (4.20). For each section *A*- section *B* to section *L* interaction, there are 12 interaction coefficients for DOF 2 expansions, 4 for each of the $\beta_{A\_L,\,B\_L}^{L,\,p\_loss}$, $\beta_{A\_L,\,B\_L}^{L,\,A\_loss}$, $\beta_{A\_L,\,B\_L}^{L,\,B\_loss}$ terms. For DOF 1 expansions, there are 3 interaction coefficients, one for each of the $\beta_{A,\,B}^{L,\,p\_loss}$, $\beta_{A,\,B}^{L,\,A\_loss}$, $\beta_{A,\,B}^{L,\,B\_loss}$ terms.

4.3.1 Temperature and Bulk Composition parameterization

The cost of computing $\beta_{A\_L,\,B\_L}^{L,\,p\_loss}$ and the other coefficients at each coagulation rate of progress evaluation is prohibitively expensive. For example, if there are *N* sections then in general there will be on the order of 1/4 N^3 section *L*-*A*-*B* evaluations to determine. Therefore, the integrals are evaluated once, and the coefficients, $\beta_{A\_L,\,B\_L}^{L,\,p\_loss}, \ldots \beta_{A\_H,\,B\_H}^{L,\,p\_loss}$, for the biquadratic basis functions

reactions are stored dynamically via a linked list tree. However, the coagulation kernels, $\beta(\mu_A, \mu_B)$, vary as a function of the temperature, pressure, gas composition (through the evaluation of the viscosity) and potentially as a function of the bulk composition within the reactant sections *A* and *B*. Therefore, $\beta^{L,\,p\_\text{loss}}_{\text{A\_L, B\_L}}$, $\beta^{L,\,A\_\text{loss}}_{\text{A\_L, B\_L}}$, and $\beta^{L,\,B\_\text{loss}}_{\text{A\_L, B\_L}}$ will be a constantly varying function of the surrounding gas conditions and local particle composition.

One answer to how to modify the coefficients is to create a multiplicative factor for each section *A* - section *B* interaction which takes into account the change in $\beta(\mu_A, \mu_B)$ from the value used when the section A - section B interaction coefficients were precalculated value to the current value. This may be done for midsection particle sizes for both section A and section B. The final expression is Eqns. (4.21-4.23)

$$S^{p\_\text{loss}}_{BAL} = fac\left[\begin{array}{l}\beta^{L,\,p\_\text{loss}}_{\text{A\_L, B\_L}} b^L_A b^L_B + \beta^{L,\,p\_\text{loss}}_{\text{A\_H, B\_L}} b^H_A b^L_B + \\ \beta^{L,\,p\_\text{loss}}_{\text{A\_L, B\_H}} b^L_A b^H_B + \beta^{L,\,p\_\text{loss}}_{\text{A\_H, B\_H}} b^H_A b^H_B\end{array}\right] \tag{4.21}$$

$$S^{A\_\text{loss}}_{BAL} = fac\left[\begin{array}{l}\beta^{L,\,A\_\text{loss}}_{\text{A\_L, B\_L}} b^L_A b^L_B + \beta^{L,\,A\_\text{loss}}_{\text{A\_H, B\_L}} b^H_A b^L_B + \\ \beta^{L,\,A\_\text{loss}}_{\text{A\_L, B\_H}} b^L_A b^H_B + \beta^{L,\,A\_\text{loss}}_{\text{A\_H, B\_H}} b^H_A b^H_B\end{array}\right] \tag{4.22}$$

$$S^{B\_\text{loss}}_{BAL} = fac\left[\begin{array}{l}\beta^{L,\,B\_\text{loss}}_{\text{A\_L, B\_L}} b^L_A b^L_B + \beta^{L,\,B\_\text{loss}}_{\text{A\_H, B\_L}} b^H_A b^L_B + \\ \beta^{L,\,B\_\text{loss}}_{\text{A\_L, B\_H}} b^L_A b^H_B + \beta^{L,\,B\_\text{loss}}_{\text{SA\_H, B\_H}} b^H_A b^H_B\end{array}\right] \tag{4.23}$$

where

$$fac = \frac{\beta\left(\left(\frac{\mu_A + \mu_{A+1}}{2}\right), \left(\frac{\mu_B + \mu_{B+1}}{2}\right), T\right)}{\beta\left(\left(\frac{\mu_A + \mu_{A+1}}{2}\right), \left(\frac{\mu_B + \mu_{B+1}}{2}\right), T^o\right)} \tag{4.24}$$

fac is the ratio of the coagulation kernel, for particles at the midsections of section *A* and section *B*, evaluated at the current environmental state to that evaluated at the state of the previous precalculation. T^o here refers to the previous precalculated state, which may include multiple state variables, not just *T*. *T* in the numerator refers to multiple state variables, not just the temperature, in the current gas-particle environment.

Note, the temperature dependence of $\beta(\mu_A, \mu_B)$ may be further analyzed. In the molecular limit, $\beta(\mu_A, \mu_B)$ can be seen (see Eqn. (4.12)) to have a $\sqrt{T}$ dependence. In the continuum limit, Eqn. (4.9), the collision integrals temperature dependence is determined by the temperature dependence of the sum of the particle diffusion coefficients. The temperature dependence of D_i^b is given by T/μ. The temperature dependence of μ varies in a complex fashion with T and the gas phase composition. However, it roughly follows a $\sqrt{T}$ dependence. Therefore, the collision frequency in the continuum limit roughly follows a $\sqrt{T}$ dependence as well. Therefore, in the initial implementation fac has been parameterized with a $\sqrt{T/T^o}$ dependence. This is expected to be changed in later implementations to the more accurate formulation in Eqn. (4.24) that handles the effects of transitional Knudsen numbers accurately.

4.4 Implementation of Numerical filters for the Coagulation Kernel

The numerical issues associated with having 2 degrees of freedom in a section for the coagulation kernel turn out to be very similar to the condensation operator. Without filters, ahead of the evolving distribution from a coagulation kernel, negative coefficients develop, because the distribution in the leading section can't accommodate an MU value outside the allowed MU band determined by the basis functions within the section. Behind the distribution, sections don't drain uniformly, leaving a skewed distribution in their wake, with near net zero particle amounts. The solution that was developed for the condensation problem for the draining problem involved changing the MU per particle value for the particles leaving a section to a value determined by the diagonalization of the 2x2 matrix problem. The same approach is used here for the coagulation operator. The math is simpler, since there is no growth term in the coagulation kernel. The details are provided below.

The draining problem is broken up into individual Section *B*- Section *A* -> Section *L* interactions. If all individual draining interactions result in stable non-negative coefficient-producing results, then the sum is guaranteed to result in the same. The problem is further subdivided into first considering section *A* and then considering section *B.* The particle loss rate, Eqn. (4.21), is never changed however during this entire process. First the effective MU per particle loss rate from section *A* is modified. Then, the same is done for section *B.* The gain MU per particle rate for section *L* is then the sum of the section *A* and *B* loss rates.

For the single draining reaction involving one reaction term, consider the follow small matrix problem for the change in coefficient values, b_i^L and b_i^H, for the drain of either Section *A* or Section *B* (we will assume *i* is section *A* below, but it could be either). The matrix problem is essentially a restatement of Eqns. (4.21) and (4.22 or 4.23). The top reaction in Eqn. (4.25) is the particle loss reaction, while the bottom reaction is the MU loss equation from the section.

$$\begin{bmatrix} c_A^L & c_A^H \\ \frac{\Delta\mu}{2} & \frac{\Delta\mu}{2} \end{bmatrix} \begin{bmatrix} \frac{\partial b_A^L}{\partial t} \\ \frac{\partial b_A^H}{\partial t} \end{bmatrix} = \begin{bmatrix} -k_{BAL}^{A,\,coag} N_A \\ -(k_{BAL}^{A,\,coag} \mathrm{MU}_{BAL}^{A,\,coag}) N_A \end{bmatrix} \tag{4.25}$$

$$c_A^H = \int_{\mu_A}^{\mu_{A+1}} \frac{q_A^H}{\mu} d\mu = 1 - \frac{\mu_A}{\Delta\mu} \ln\left(\frac{\mu_{A+1}}{\mu_A}\right) \text{ where } q_A^H = \frac{\mu - \mu_A}{\mu_{A+1} - \mu_A} \tag{4.26}$$

$$c_A^L = \int_{\mu_A}^{\mu_{A+1}} \frac{q_A^L}{\mu} d\mu = \frac{\mu_{A+1}}{\Delta\mu} \ln\left(\frac{\mu_{A+1}}{\mu_A}\right) - 1 \text{ where } q_A^L = \frac{\mu_{A+1} - \mu}{\mu_{A+1} - \mu_A} \tag{4.27}$$

$$\mathrm{MU}_A^L = \frac{\int_{\mu_A}^{\mu_{A+1}} q_A^L d\mu}{\int_{\mu_A}^{\mu_{A+1}} \frac{q_A^L}{\mu} d\mu} = \frac{\Delta\mu}{2c_A^L} \text{ and } \mathrm{MU}_A^H = \frac{\int_{\mu_A}^{\mu_{A+1}} q_A^H d\mu}{\int_{\mu_A}^{\mu_{A+1}} \frac{q_A^H}{\mu} d\mu} = \frac{\Delta\mu}{2c_A^H}$$

$$N_A = c_A^H b_A^H + c_A^L b_A^L \tag{4.28}$$

c_A^H and c_A^L are the particle number coefficients for the lower and upper DOFs in section *A*. q_A^H and q_A^L are the upper and lower basis functions for Section *A*. MU_A^H and MU_A^L are the upper and lower monomer units per particle in the upper and lower basis function modes in section *A*. N_A is the number of particles in Section *A*. $k_{BAL}^{A,\,coag}$ is the loss rate per number of particles in section *A* due to the particular section *B*- section *A*-> section *L* interaction (*BAL*) under consideration. Because the focus is on one section, section *A*, the sectional coefficients for section *B* are included in the term, $k_{BAL}^{A,\,coag}$. $\mathrm{MU}_{BAL}^{A,\,coag}$, the importance of which will be described below, isthe loss rate of total monomer units per loss rate of particles in section *A.*

One necessary condition for proper draining of a section with a skewed distribution is the requirement that $\frac{\partial}{\partial t} b_A^L = 0$ when $b_A^L = 0$. Solving Eqn. (4.25), when $b_A^L = 0$ and thus $N_A = c_A^H b_A^H$:

$$DET\begin{bmatrix}\frac{\partial b_A^L}{\partial t}\\ \frac{\partial b_A^H}{\partial t}\end{bmatrix} = \begin{bmatrix}\frac{\Delta\mu}{2} & -c_A^H\\ -\frac{\Delta\mu}{2} & c_A^L\end{bmatrix}\begin{bmatrix}-k_{BAL}^{A,\,coag}N_A\\ -(k_{BAL}^{A,\,coag}\mathrm{MU}_{BAL}^{A,\,coag})N_A\end{bmatrix} \tag{4.29}$$

$$\text{with } DET = -\Delta\mu + \frac{\mu_A + \mu_{A+1}}{2}\ln\left(\frac{\mu_{A+1}}{\mu_A}\right)$$

$$DET\frac{\partial b_A^L}{\partial t} = k_{BAL}^{A,\,coag}N_A\left(-\frac{\Delta\mu}{2} + c_A^H\mathrm{MU}_{BAL}^{A,\,coag}\right) = 0 \tag{4.30}$$

Thus, for $\frac{\partial}{\partial t}b_A^L = 0$ to be true,

$$\mathrm{MU}_{BAL}^{A,\,coag} = \frac{\Delta\mu}{2c_A^H} = \mathrm{MU}_H^A \tag{4.31}$$

$$DET\frac{\partial b_A^H}{\partial t} = k_{BAL}^{A,\,coag}N_A\frac{\Delta\mu}{2} - (k_{BAL}^{A,\,coag}\mathrm{MU}_{BAL}^{A,\,coag})c_A^L N_A \text{ or } \frac{\partial b_A^H}{\partial t} = -k_{BAL}^{A,\,coag}b_A^H$$

Therefore, we should require that as $b_A^L \to 0$, $\mathrm{MU}_{BAL}^{A,\,coag} \to \mathrm{MU}_A^H$. This requirement is almost met by the coagulation kernel by itself. However, other influences in the coagulation integral, such as a variable collision rate and variations in other particle distributions in other sections cause this not to be the case rigorously. Therefore, we require a condition on the kernel that all collisions involving a “draining” section have the requirement:

$$\mathrm{MU}_{BAL}^{A,\,coag} \geq \mathrm{MU}_H^A, \text{ when } b_A^L \text{ is below a threshold and } b_A^H \text{ is above.} \tag{4.32}$$

$$\mathrm{MU}_{BAL}^{A,\,coag} \leq \mathrm{MU}_L^A, \text{ when } b_A^H \text{ is below a threshold and } b_A^L \text{ is above.} \tag{4.33}$$

The inequality comes about from looking at Eqn. (4.30) and requiring that, when $b_A^L = 0$, $\frac{\partial}{\partial t}b_A^L$ should be greater or equal to zero, in order for the skewed condition not to occur. In words, when $b_A^L = 0$, and thus all of the particles are in the top mode of the section, particles which are heavier than the average MU per particle of the top mode should be taken out of the section, in order for b_A^L to stay positive. The condition in Eqn. (4.33) is the exact compliment of the Eqn. (4.32) condition for the opposite case where $b_A^H \to 0$.

When both b_A^L and b_A^H are positive but below a tolerance, the following condition for the coagulation reaction rate is required:

$$\mathrm{MU}_{BAL}^{A,\,coag} = \overline{\mathrm{MU}^A} = \frac{Q_A}{N_A} \text{ when } b_A^L \text{ and } b_A^H \text{ are below a threshold} \tag{4.34}$$

Eqn. (4.34) forces Section *A* to drain to zero as a single entity, with both modes draining at the same relative rate constant. Eqns. (4.32), (4.33), and (4.34) are the basic strategy used in the method `BimolecularRxnLowSections` in the input deck described in Section 4.6. This is the default drain stabilization strategy used in `CADS`.

An alternative strategy within `CADS` is available. The origin of the different method stems from the differences between Eqn. (4.21) and Eqn. (4.25)'s description of the drain reaction in Section *A*. Eqn. (4.21) suggests that the high and low modes of Section *A* can drain independently at different rate constants. Rewriting Eqn. (4.25) to reflect this fact leads to Eqn. (4.35).

$$\begin{bmatrix} c_A^L & c_A^H \\ \frac{\Delta\mu}{2} & \frac{\Delta\mu}{2} \end{bmatrix} \begin{bmatrix} \frac{\partial b_A^L}{\partial t} \\ \frac{\partial b_A^H}{\partial t} \end{bmatrix} = -\begin{bmatrix} k_{BAL}^{AL,\,coag} c_A^L b_A^L + k_{BAL}^{AH,\,coag} c_A^H b_A^H \\ k_{BAL}^{AL,\,coag} \mathrm{MU}_{BAL}^{AL,\,coag} c_A^L b_A^L + k_{BAL}^{AH,\,coag} \mathrm{MU}_{BAL}^{AH,\,coag} c_A^H b_A^H \end{bmatrix} \tag{4.35}$$

where the equations,

$$\begin{aligned} k_{BAL}^{AL,\,coag} c_A^L &= (\beta_{\mathrm{A\_L,\,B\_L}}^{L,\,p\_\mathrm{loss}} b_B^L + \beta_{\mathrm{A\_L,\,B\_H}}^{L,\,p\_\mathrm{loss}} b_B^H) fac \\ k_{BAL}^{AH,\,coag} c_A^H &= (\beta_{\mathrm{A\_H,\,B\_L}}^{L,\,p\_\mathrm{loss}} b_B^L + \beta_{\mathrm{A\_H,\,B\_H}}^{L,\,p\_\mathrm{loss}} b_B^H) fac \\ k_{BAL}^{AL,\,coag} \mathrm{MU}_{BAL}^{AL,\,coag} c_A^L &= (\beta_{\mathrm{A\_L,\,B\_L}}^{L,\,A\_\mathrm{loss}} b_B^L + \beta_{\mathrm{A\_L,\,B\_H}}^{L,\,A\_\mathrm{loss}} b_B^H) fac \\ k_{BAL}^{AH,\,coag} \mathrm{MU}_{BAL}^{AH,\,coag} c_A^L &= (\beta_{\mathrm{A\_H,\,B\_L}}^{L,\,A\_\mathrm{loss}} b_B^L + \beta_{\mathrm{A\_H,\,B\_H}}^{L,\,A\_\mathrm{loss}} b_B^H) fac \end{aligned}, \tag{4.36}$$

defines the new terms, $k_{BAL}^{AL,\,coag}$, $k_{BAL}^{AH,\,coag}$, $\mathrm{MU}_{BAL}^{AL,\,coag}$, $\mathrm{MU}_{BAL}^{AH,\,coag}$, from previously defined quantities.

Continuing as before, the necessary condition for proper draining with a developing a skewed distribution is that when $b_A^L = 0$, then $\frac{\partial}{\partial t} b_A^L = 0$. Solving Eqn. (4.35), when $b_A^L = 0$ and thus $N_A = c_A^H b_A^H$ yields:

$$\mathrm{MU}_{BAL}^{AH,\,coag} = \frac{\Delta\mu}{2c_A^H} \rightarrow \mathrm{MU}_A^H \tag{4.37}$$

$$\frac{\partial b_A^H}{\partial t} = -k_{BAL}^{AH,\,coag} b_A^H \tag{4.38}$$

The other necessary condition that when $b_A^H = 0$, $\frac{\partial}{\partial t} b_A^H = 0$ leads to

$$\mathrm{MU}_{BAL}^{AL,\,coag} = \frac{\Delta\mu}{2c_A^L} \rightarrow \mathrm{MU}_A^L \tag{4.39}$$

$$\frac{\partial b_A^L}{\partial t} = -k_{BAL}^{AL,\,coag} b_A^L \tag{4.40}$$

Therefore, the following alternative stabilizing strategy described by Eqns. (4.41), (4.42), and (4.43) may be used.

$$\begin{aligned} \mathrm{MU}_{BAL}^{AL,\,coag} &\rightarrow \mathrm{MU}_A^L \\ \mathrm{MU}_{BAL}^{AH,\,coag} &\rightarrow \mathrm{MU}_A^H \end{aligned} \quad \text{when } b_A^L \text{ and } b_A^H \text{ are both below a threshold} \tag{4.41}$$

$$\begin{aligned} \mathrm{MU}_{BAL}^{AL,\,coag} &\rightarrow \mathrm{MU}_A^L \\ \mathrm{MU}_{BAL}^{AH,\,coag} &\rightarrow \mathrm{MU}_A^H \end{aligned} \quad \text{when } b_A^L \text{ is below and } b_A^H \text{ is above a threshold} \tag{4.42}$$

$$\begin{aligned} \mathrm{MU}_{BAL}^{AL,\,coag} &\rightarrow \mathrm{MU}_A^L \\ \mathrm{MU}_{BAL}^{AH,\,coag} &\rightarrow \mathrm{MU}_A^H \end{aligned} \quad \text{when } b_A^H \text{ is below and } b_A^L \text{ is above a threshold} \tag{4.43}$$

This strategy is called the "`BimolecularRxn`" strategy in the input deck described in Section 4.6. The name `BimolecularRxn` is used, because the implementation closely resembles treating the two modes of a section with separate drain reactions near conditions that would cause the section to undergo non-negative conditions. Away from problem conditions, however, the full 12 coefficient, accurate implementation (Eqns. (4.21, 4.22, and 4.23) is maintained to describe the *BAL* interaction.

Now that we have described the strategy for section *A*, what about section *B*? Because the particle loss terms haven't been touched in the treatment for section *A*, the section *B* treatment can be handled independently of the section *A* treatment. Thus, the section *B* treatment is the same as has

been described for section *A*. The average MU loss for section *B* is adjusted, when necessary to maintain numerical stability. The final result is Eqn. (4.44).

$$S_{BAL}^{B\_loss\_adj} = (1 - ramp_B)S_{BAL}^{B\_loss} + (ramp_B)S_{BAL}^{B\_loss\_star} \tag{4.44}$$

where

$$S_{BAL}^{B\_loss\_star} = \begin{array}{l} \beta_{A\_L,\,B\_L}^{L,\,p\_loss}\mathrm{MU}_B^L b_A^L b_B^L + \beta_{A\_H,\,B\_L}^{L,\,p\_loss}\mathrm{MU}_B^L b_A^H b_B^L + \\ \beta_{A\_L,\,B\_H}^{L,\,p\_loss}\mathrm{MU}_B^H b_A^L b_B^H + \beta_{A\_H,\,B\_H}^{L,\,p\_loss}\mathrm{MU}_B^H b_A^H b_B^H \end{array} \tag{4.45}$$

After the drain operations have been carried out, the source term vector for the single section *B* - section *A* -> section *L* interaction may be described by the following equation.

$$\boldsymbol{S}_{BAL}^{coag} = (1 - ramp_A)\boldsymbol{S}_{BAL}^{acc} + (ramp_A)(\boldsymbol{S}_{BAL}^{AstarL} + \boldsymbol{S}_{BAL}^{AstarH}) \tag{4.46}$$

$\boldsymbol{S}_{BAL}^{acc}$ is the accurate source term, containing all 12 coefficients, with $S_{BAL}^{p\_loss}$, $S_{BAL}^{A\_loss}$, $S_{BAL}^{B\_loss\_adj}$ given by Eqns. (4.21), (4.22), and (4.45), and $S_{BAL}^{L\_gain} = S_{BAL}^{A\_loss} + S_{BAL}^{B\_loss\_adj}$.

$$\boldsymbol{S}_{BAL}^{acc} = \begin{bmatrix} -S_{BAL}^{p\_loss} \\ -S_{BAL}^{A\_loss} \\ -S_{BAL}^{B\_loss\_adj} \\ S_{BAL}^{L\_gain} \end{bmatrix} \tag{4.47}$$

$\boldsymbol{S}_{BAL}^{AstarL}$ is the stable source term for the *BAL* interaction that is attributable to the lower mode *L* of section *A*. $\boldsymbol{S}_{BAL}^{AstarH}$ is the stable source term for the *BAL* interaction that is attributable to the higher mode *H* of section *A*.

$$\boldsymbol{S}_{BAL}^{AstarL} = \begin{bmatrix} -S_{BAL}^{AstarL,\,p\_loss} \\ -S_{BAL}^{AstarL,\,A\_loss} \\ -S_{BAL}^{AstarL,\,B\_loss\_adj} \\ S_{BAL}^{AstarL,\,L\_gain} \end{bmatrix} \text{ and } \boldsymbol{S}_{BAL}^{AstarH} = \begin{bmatrix} -S_{BAL}^{AstarH,\,p\_loss} \\ -S_{BAL}^{AstarH,\,A\_loss} \\ -S_{BAL}^{AstarH,\,B\_loss\_adj} \\ S_{BAL}^{AstarH,\,L\_gain} \end{bmatrix} \tag{4.48}$$

The following relations hold

$$S_{BAL}^{AstarL,\,p\_\text{loss}} = k_{BAL}^{AL,\,coag} c_A^L b_A^L \tag{4.49}$$

$$S_{BAL}^{AstarH,\,p\_\text{loss}} = k_{BAL}^{AH,\,coag} c_A^H b_A^H$$

$$S_{BAL}^{AstarL,\,A\_\text{loss}} = S_{BAL}^{AstarL,\,p\_\text{loss}} \text{MU}_A^L$$

$$S_{BAL}^{AstarL,\,A\_\text{loss}} = S_{BAL}^{AstarH,\,p\_\text{loss}} \text{MU}_A^H$$

$$S_{BAL}^{AstarL,\,B\_\text{loss\_adj}} = (1 - ramp_B) S_{BAL}^{AstarL,\,Bacc\_\text{loss}} + (ramp_B) S_{BAL}^{AstarL,\,Bstab\_\text{loss}}$$

$$S_{BAL}^{AstarH,\,B\_\text{loss\_adj}} = (1 - ramp_B) S_{BAL}^{AstarH,\,Bacc\_\text{loss}} + (ramp_B) S_{BAL}^{AstarH,\,Bstab\_\text{loss}}$$

$S_{BAL}^{AstarL,\,p\_\text{loss}}$ and $S_{BAL}^{AstarH,\,p\_\text{loss}}$ are the parts of the particle loss source term that can be attributed to the lower and upper modes of section *A*. $S_{BAL}^{AstarL,\,Bacc\_\text{loss}}$ is the part of the MU loss from section B calculated from the accurate formula, that can be attributed to the lower mode of section *A*. $S_{BAL}^{AstarL,\,Bstab\_\text{loss}}$ is the part of the MU loss from section B calculated from the stable formula (which is $S_{BAL}^{AstarL,\,p\_\text{loss}} \text{MU}_B^L$), that can be attributed to the lower mode of section *A*. $S_{BAL}^{AstarH,\,Bacc\_\text{loss}}$ and $S_{BAL}^{AstarH,\,Bstab\_\text{loss}}$ are defined analogously. $S_{BAL}^{L\_\text{gain}}$, $S_{BAL}^{AstarL,\,L\_\text{gain}}$, and $S_{BAL}^{AstarH,\,L\_\text{gain}}$ are defined so that each source term conserves mass and elements.

4.5 A Filter For Source Term Additions to a Section

Without any filter, it is observed that the an advancing coagulation distribution will create sections ahead of the main distribution where $b_L^H < 0$. The reason for this is that the coagulation operator is trying to create a mean distribution within the section where, $\overline{\text{MU}}_L < \text{MU}_L^L$, by adding particles with a small MU to particle ratio relative to the bounds of the section. In order to accommodate this request, b_L^H is driven to negative values. This problem may be solved by splitting the incoming particle stream into two parts, one with $\overline{\text{MU}}_L = \text{MU}_L^L$, and one with particles diverted into the next lower section with $\overline{\text{MU}} = \mu_L$. Here is the math involved with this tie-line calculation.

Let $S_{B+A\rightarrow L}^{\text{coag, part}}$ be the source term for particles incoming into section *L* from the section *A* - section *B* interaction; it's name will be shortened to S^{part} below. Let $S_{B+A\rightarrow L}^{\text{coag, TMU}}$ be the source term for total monomer units; it's name will be shortened to S^{TMU} below. Let $\text{MU}_{B+A\rightarrow L}^{\text{coag}}$ be the average MU per particle in the source term,

$$\mathrm{MU}^{\mathrm{coag}} = \frac{S^{\mathrm{TMU}}}{S^{part}} \text{ with the condition that } \mathrm{MU}^{\mathrm{coag}} < \mathrm{MU}_L^L .$$

Then we will distribute this source term partly into the lower section, Section *L-1*, which is alternatively called *Lalt*, under some conditions to preserve positivity of the coefficients. These conditions are when b_L^H is below and b_L^L above a threshold, and also, it turns out, when both b_L^H and b_L^L are below a threshold. Changing the names of S^{part} and S^{TMU} to $S_L^{part,\,\mathrm{orig}}$ and $S_L^{\mathrm{TMU,\,orig}}$, Eqn. (4.50) expresses the condition for conservation of particles and TMU in the partition of the original source term.

$$\begin{aligned} S_L^{part,\,\mathrm{orig}} &= S_L^{part,\,\mathrm{new}} + S_{L-1}^{part,\,\mathrm{new}} \\ S_L^{\mathrm{TMU,\,orig}} &= S_L^{\mathrm{TMU,\,new}} + S_{L-1}^{\mathrm{TMU,\,new}} = \mathrm{MU}_L^L S_L^{part,\,\mathrm{new}} + \mu_L S_{L-1}^{part,\,\mathrm{new}} \end{aligned} \tag{4.50}$$

Solving the 2x2 system:

$$\begin{aligned} S_L^{part,\,\mathrm{new}} &= \frac{S_L^{\mathrm{TMU,\,orig}} - S_L^{part,\,\mathrm{orig}}\mu_L}{\mathrm{MU}_L^L - \mu_L} & S_L^{\mathrm{TMU,\,new}} &= S_L^{part,\,\mathrm{new}}\mathrm{MU}_L^L \\ S_{L-1}^{part,\,\mathrm{new}} &= \frac{S_L^{part,\,\mathrm{orig}}\mathrm{MU}_L^L - S_L^{\mathrm{TMU,\,orig}}}{\mathrm{MU}_L^L - \mu_L} & S_{L-1}^{\mathrm{TMU,\,new}} &= S_{L-1}^{part,\,\mathrm{new}}\mu_L \end{aligned} \tag{4.51}$$

Eqn. (4.51) is a simple tie-line condition. Note, an alternative implementation could spot the average monomer units per particle entering section *L-1* at the value MU_{L-1}^H. This would be appropriate if we were also concerned with skewness in Section *L-1*. Then, the final result would be Eqn. (4.52).

$$\begin{aligned} S_L^{part,\,\mathrm{new}} &= \frac{S_L^{\mathrm{TMU,\,orig}} - S_L^{part,\,\mathrm{orig}}\mathrm{MU}_{L-1}^H}{\mathrm{MU}_L^L - \mathrm{MU}_{L-1}^H} & S_L^{\mathrm{TMU,\,new}} &= S_L^{part,\,\mathrm{new}}\mathrm{MU}_L^L \\ S_{L-1}^{part,\,\mathrm{new}} &= \frac{S_L^{part,\,\mathrm{orig}}\mathrm{MU}_L^L - S_L^{\mathrm{TMU,\,orig}}}{\mathrm{MU}_L^L - \mathrm{MU}_{L-1}^H} & S_{L-1}^{\mathrm{TMU,\,new}} &= S_{L-1}^{part,\,\mathrm{new}}\mathrm{MU}_{L-1}^H \end{aligned} \tag{4.52}$$

Both of these filters, Eqn.(4.51) and Eqn. (4.52), require a ramping condition in order to turn them on, if continuity in the overall source term is to be maintained. The same criteria as was used for the condensation operator, Eqn. (3.40), is used for the ramping condition.

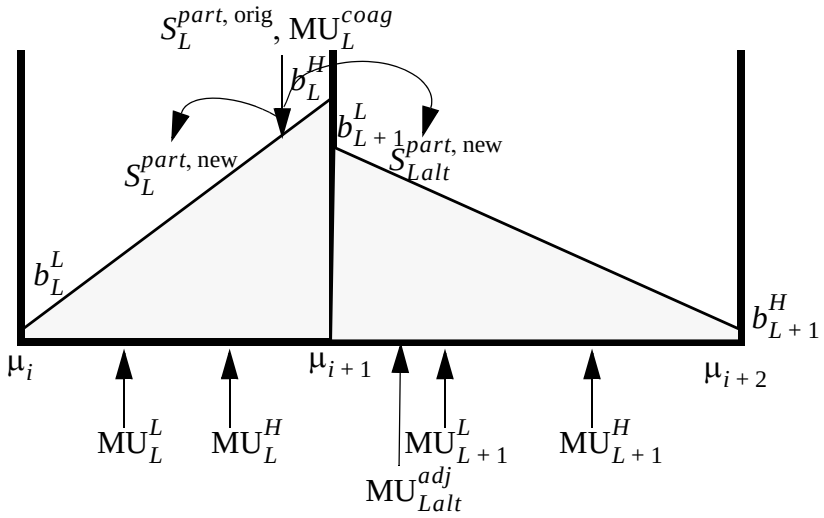

Figure 4.7 Pictorial description for when an influx of particles into section *L* must be split adjusted to flow into section *L*and section *L+1*, when the MU value of the influx is unsupportable by the section.

In Figure (4.7) we present a schematic of the filtering scenario described by Eqn. (4.52), except that the total MU's in the flux are now too high for section *L* and the section *L* coefficients have the property that b_L^L is below and b_L^H is above a threshold. Then, the alternate section that will be receiving some of the particles originally destined for section *L* is section *L+1*. Section *L+1* is called *Lalt*.

The total source term for section *L* consists of a flux of particles $S_L^{part,\,orig}$ with total mass flux of $S_L^{TMU,\,orig}$ such that the average mass per incoming particle is $S_L^{TMU,\,orig} / S_L^{part,\,orig} = MU_L^{coag}$. If $MU_L^L < MU_L^{coag} < MU_L^H$, then the section can in all cases accept the flux as is. However, if $MU_L^H < MU_L^{coag}$, then the ability of the section to absorb the source of particles without b_L^L perhaps extending into negative values depends on the relative values of b_L^H and b_L^L, the magnitude of the particle flux, $S_L^{part,\,orig}$ and the time step. When b_L^H is much greater than b_L^L, some of the particle flux $S_{Lalt}^{part,\,new}$ must be moved to a higher zone, $L+1$, with a greater MU per particle value, so that the remaining portion of the particle flux, $S_L^{part,\,new}$, can have its MU per particle value adjusted downwards to MU_L^H, the highest MU per particle value supported by nonnegative coefficients within section *L*.

In choosing the MU value per particle for section $L+1$, an additional complication arises. At first we may want to select an MU value of μ_{L+1}, the lowest in section $L+1$, in order to minimize the effect of this operation on the accuracy of the coagulation kernel. However, we may risk creating a skewed distribution within section $L+1$ with our efforts to avoid a skewed distribution within

section L. To avoid this possibility, we adjust the MU value per particle to be assigned to the $S_{L+1}^{part,\,\mathrm{new}}$ particle flux, MU_{L+1}^{adj}:

$$\mathrm{MU}_{L+1}^{adj} = \mu_{i+1} + Ramp^{L+1}(\mathrm{MU}_{L+1}^{L} - \mu_{i+1}). \tag{4.53}$$

$Ramp^{L+1}$ is a function of b_{L+1}^{L} and b_{L+1}^{H}, varying between 0 and 1. When $b_{L+1}^{H} \ll b_{L+1}^{L}$, $Ramp^{L+1} = 1$. When this is not the case, $Ramp^{i+1}$ decreases to zero in a continuous manner such that residual evaluation retains its smoothness as a function of b_{L+1}^{L} and b_{L+1}^{H}.

A tie line approach is used to distribute the particle fluxes between sections L and $L+1$, such that the total particle and mass fluxes are preserved.

$$S_{L}^{part,\,new} = \frac{\mathrm{MU}_{L+1}^{adj} S_{L}^{part,\,\mathrm{orig}} - S_{L+1}^{\mathrm{TMU,\,orig}}}{\mathrm{MU}_{L+1}^{adj} - \mathrm{MU}_{L}^{H}} \quad S_{L+1}^{part,\,new} = \frac{S_{L+1}^{\mathrm{TMU,\,orig}} - \mathrm{MU}_{L}^{H} S_{L}^{part,\,\mathrm{orig}}}{\mathrm{MU}_{L+1}^{adj} - \mathrm{MU}_{L}^{H}} \tag{4.54}$$

The mass flux into section L, $S_{L}^{\mathrm{TMU},\,new}$ and section $L+1$, $S_{L+1}^{\mathrm{TMU},\,new}$, is adjusted as

$$S_{L}^{\mathrm{TMU},\,new} = \mathrm{MU}_{L}^{H} S_{L}^{part,\,new} \qquad S_{L+1}^{\mathrm{TMU},\,new} = \mathrm{MU}_{L+1}^{adj} S_{L+1}^{part,\,new}. \tag{4.55}$$

Finally, the entire procedure involving split adjusting the particle flux P_f must be ramped as a function of the relative values of b_{L}^{H} and b_{L}^{L} in the same way we adjusted MU_{L+1}^{adj}, in order to maintain smoothness in the residual evaluation.

$$\begin{aligned} \boldsymbol{S}^{part,\,\mathrm{final}} &= \boldsymbol{S}^{part,\,\mathrm{orig}} + Ramp^{i}(\boldsymbol{S}^{part,\,new} - \boldsymbol{S}^{part,\,\mathrm{orig}}) \\ \boldsymbol{S}^{\mathrm{TMU,\,final}} &= \boldsymbol{S}^{\mathrm{TMU,\,orig}} + Ramp^{i}(\boldsymbol{S}^{\mathrm{TMU},\,new} - \boldsymbol{S}^{\mathrm{TMU,\,orig}}) \end{aligned} \tag{4.56}$$

$\boldsymbol{S}^{part,\,\mathrm{orig}}$ denotes the original particle flux with an MU per particle value of MU_{L}^{coag}. $\boldsymbol{S}^{part,\,new}$ is the particle flux to sections L and $L+1$ that is the final result to the entire split adjustment procedure described above. $\boldsymbol{S}^{part,\,\mathrm{final}}$ represents the total particle flux that is employed as the final source term for the single Section *B*- Section *A*-> Section *L* coagulation interaction.

4.5.1 Adjustment to the Source Term Addition Algorithm

Eqn. (4.46) has described the case where the original source term for the *BAL* interaction, $\boldsymbol{S}_{BAL}^{coag}$, has been broken up into three parts, $\boldsymbol{S}_{BAL}^{acc}$, $\boldsymbol{S}_{BAL}^{AstarL}$, $\boldsymbol{S}_{BAL}^{AstarH}$, in order to ensure non-negativity of the sectional coefficients. In practice, for evolving coagulation distributions along the high side of the distribution, it was observed that when $ramp_A = 1$, the split-adjust procedure described in the previous section frequently lead to the case where none of the original flux into the destination

section *L*, actually ended up in section *L*. Instead it ended up in the alternate section, *L-1*. To correct this and obtain a smoother and more accurate distribution as a function of time, the split adjust procedure was modified. The modified procedure split-adjusted the three individual source terms, $\boldsymbol{S}_{BAL}^{acc}$, $\boldsymbol{S}_{BAL}^{AstarL}$, $\boldsymbol{S}_{BAL}^{AstarH}$, separately before combining the total into the final result. With this modification, there was always some of the initial source term that ended up in section *L* (from the split-adjustment of $\boldsymbol{S}_{BAL}^{AstarH}$).

4.5.2 DOF 1 Split-Adjust Methods

Up to now, the discussion has been exclusively concerned with stabilizing DOF 2 methods. However, the DOF 1 case needs split adjustments as well, in order to account for the particle number and to conserve mass/elements at the same time. The reason for the need to split adjust is simply explained. A DOF1 section contains a fixed monomer unit to particle ratio, MU_i, where for us the following relation holds, $\mathrm{MU}_i = 0.5(\mu_i + \mu_{i+1})$, while a DOF 2 section contains a variable ratio (where $\mathrm{MU}_i^L < \mathrm{MU} < \mathrm{MU}_i^H$). Therefore, in order to conserve mass, an incoming particle source stream with an incoming monomer unit to particle ratio, MU^{coag}, must be divided into 2 sections in order to conserve particles and mass in the source stream. This is the same tie-line calculation as was previously described in Section 4.5 in Figure. 4.7 and Eqn. (4.46).

4.6 Input Deck Options

The preceding sections have described the theory behind the options available to `CADS` users. Figure (4.8) shows the section block of the input file for `CADS` that controls the coagulation kernel.

The first option “Method” toggles on and off the coagulation kernel in its entirety. Setting it to “`NONE`”, turns the kernel off. Setting it to “`ADS`”, turns the kernel on.

The next optional keyword, “Collision Rate Formula”, sets the formulation for the collision rates between particles. There are four options: `Mixed`, `Free Molecular`, `Continuum Limit`, and `Zero`. The default, `Mixed`, implements the Fuchs formula, Eqn. (4.3). `Free Molecular` implements the low-pressure limiting collision rate formula, Eqn. (4.12). `Continuum Limit` implements the continuum-limit formula, Eqn. (4.9). `Zero` sets the coagulation kernel to zero.

The “`DOF2 Interaction Method`” keyword broadly determines how the different degrees of freedom in an a DOF 2 interpolation of a section interact with each other in the coagulation kernel. The default method is `CombinedHiLow`. If that is used, then each section *A* - section *B* $\rightarrow$ section *L* interaction produces one source vector that then gets distributed into section *L* and surrounding sections if necessary to preserve non negativity.

```
START BLOCK ADS MODEL DEFINITION
START BLOCK COAGULATION MODEL
   ! Method = [NONE | ADS]
   !     (required) (default = ADS)
   Method = ADS
   !-------------------------------------------------------------
   !   Collision Rate Formula = {"Mixed", "Free Molecular",
   !                             "Continuum limit", zero }
   !        (Default="Mixed")   (optional)
   Collision Rate Formula = Mixed
   !-------------------------------------------------------------
   ! DOF2 Interaction Method = [ CombinedHighLow |
   !                              SeparateHighLow]
   !        Method for how the different degrees of freedom in a
   !        DOF 2 interpolation of a section interact with each
   !        other in the coagulation kernel.
   !             (Default = CombinedHighLow) (optional)
   DOF2 Interaction Method = CombinedHighLow
   !-------------------------------------------------------------
   ! Drain MU Adjust Method = [NONE | Maximal | Minimal |
   !                           BimolecularRxnLowSections |
   !                           BimolecularRxn | AlwaysSeparateHighLow]
   !   (optional) (Default = BimolecularRxnLowSections)
   !           Method for specifying the coagulation method with CADS.
   Drain MU Adjust Method = BimolecularRxnLowSections
   !-------------------------------------------------------------
   ! Section Addition Adjust Method = [Conservative | Ramp |
   !                                   Always Accurate]
   !       Method for filling in a section. This sets algorithm
   !       for determining the MU to particle ratios entering a
   !       section as a function of the distribution of
   !       particle ratios already in the section.
   !            (default = ramp) (optional)
   Section Addition Adjust Method = Ramp
   !-------------------------------------------------------------
END BLOCK COAGULATION MODEL
END BLOCK ADS MODEL DEFINITION
```

Figure 4.8 Block of the input file that controls Coagulation

The other option is "`SeparateHiLow`". In this treatment each mode of the DOF 2 section, i.e., the low mode and the high mode, is handled relatively separately. Thus, the section A - section $B \rightarrow$ section L interaction produces 4 source vectors. Each of those source vectors are then distributed into section L and surrounding sections (if necessary) separately. The result of this is that the DOF2 coagulation kernel behaves similar to the DOF 1 kernel as each mode in a section interacts separately and also unfortunately has similar convergence properties. Each interaction has a fixed final MU ratio for the product particle. However, because the high and low mode in a draining section is treated separately, the skew problem described in the previous sections is completed eliminated.

At the source code level, the "`DOF2 Interaction Method`" causes different child objects of the parent coagulation kernel object to be employed. The `CombinedHiLow` object causes the `PartCoagDOF2Combined` object to be used. The "`SeparateHiLow`" object causes the `PartCoagDOF2Bimolecular` object to be used.

The "`Drain MU Adjust Method`" modifies the way the coagulation kernel handles the draining of a section. Currently, there are 6 of these methods. The default is called `BimolecularRxnLowSections`, which was described in Section 4.4. An alternative to the default is `BimolecularRxn`, which was also described in Section 4.4. The method `NONE` may be described, as well, which turns off all drain adjustments. The methods `Maximal` and `Minimal` are first-generation methods, which are discouraged and won't be described here.

The line, "`Section Addition Adjust Method`" sets the algorithm for determining the MU to particle ratios entering a section as a function of the distribution of particle ratios already in the section. There is a balance between accuracy and numerical stability (numerical stability both in terms of the non negativity of the distribution and the smoothness of the Jacobian terms. The theory behind this was discussed in Section 4.5. The `Conservative` method always injects an MU ratio, MU^{coag},into section *i* such that $\mathrm{MU}_i^L < \mathrm{MU}^{coag} < \mathrm{MU}_i^H$. The `Ramp` method only changes the incoming MU^{coag} ratio if the change is required due to the existing distribution in the section. The `Always Accurate` option never changes the incoming MU^{coag} distribution. Negative coefficients are to be expected.

Note, all of the above options only modify the DOF 2 case. Except for the collision rate formula, the DOF 1 case is not affected by any of these values. Note, in the DOF 1 case, split adjustments are always done in order to account for both particle numbers and to conserve mass/elements.

4.7 Sample Problem for Particle Growth via Coagulation

A particle distribution undergoing pure coagulation driven by either the free-molecule coagulation kernel or the continuum-regime coagulation kernel will develop a self-similar distribution for each kernel [4]. Any numerical description of coagulation employing one of these kernels should generate the respective self similar distribution. Friedlander [4] defines the form and requirements of the self-similar or self preserving particle size distribution. The size distribution function, $n(\mu, t)$, is defined as the differential particle number concentration, *dN*, that lies between particle sizes, μ and $\mu + d\mu$, at time *t*. Eqn. (4.57) represents the differential equation.

$$n(\mu, t) = \frac{dN}{d\mu} \tag{4.57}$$

N is the cumulative size distribution function, i.e., the number of particles having size μ or lower. The integral relations for the total particle number, N_T,

$$N_T = \int_0^\infty n \, d\mu \ , \tag{4.58}$$

and the total volume of particles, ϕ,

$$\phi = \int_0^\infty nV(\mu)d\mu \, , \tag{4.59}$$

where $V(\mu)$ is the volume of a particle of size μ, hold for the distribution function with the additional condition that $n(\mu, t)$ approaches zero at both extremes of μ approaching zero and infinity.

Coagulation of particles is described by the integral-differential equation Eqn. (4.60).

$$\frac{d}{dt}n(\mu, t) = \frac{1}{2}\int_0^{\nu} \beta(\tilde{\mu}, \mu - \tilde{\mu})n(\tilde{\mu})n(\mu - \tilde{\mu})d\tilde{\mu} - \int_0^\infty \beta(\mu, \tilde{\mu})n(\tilde{\mu})n(\mu)d\tilde{\mu} \tag{4.60}$$

Integrating Eqn. (4.60) over all particle MU sizes, μ, we obtain an equation for the total number of particles as a function of time,

$$\frac{dN_T}{dt} = -\frac{1}{2}\int_0^\infty\int_0^\infty \beta(\mu, \tilde{\mu})n(\tilde{\mu})n(\mu)d\tilde{\mu}d\mu \, , \tag{4.61}$$

where $\beta(\mu, \mu)$ is the coagulation kernel giving the collision frequency (units of m^3 (kmol particles)$^{-1}$ sec^{-1}), between two particles of size μ and size μ. The coagulation kernel has one form for the free molecule particle regime, where particles "see" the gas as individual molecules, and a different form for the continuum particle regime, where the particles "see" the gas as a continuum. In our application, we have a form that interpolates between free molecule and continuum and goes to the correct limits in each case (see Section 4.1).

The size distribution function can be made dimensionless. The average volume of a particle, $\bar{V}$, is defined as the total particle volume concentration ϕ divided by the total particle number concentration, N_T.

$$\bar{V} = \frac{\phi}{N_T} = \frac{\int_0^\infty nVd\mu}{\int_0^\infty n \, d\mu} \text{ where } V = v_o\mu \tag{4.62}$$

v_o is the volume of the particle per monomer unit. The normalized particle size, η, is defined as the particle volume, V, divided by the average volume of a particle in the distribution, $\bar{V}$.

$$\eta = \frac{V}{\bar{V}} = \frac{V \cdot N_T}{\phi} \tag{4.63}$$

The dimensionless distribution function, $\psi(\eta, t)$, is defined as

$$\psi(\eta, t) = \frac{n(\mu, t)\phi}{N_T^2 v_o} \ . \tag{4.64}$$

Integral relations,

$$\int_0^\infty \psi(\eta, t) d\eta = 1 \tag{4.65}$$

and

$$\int_0^\infty \eta\psi(\eta, t) d\eta = 1 \ , \tag{4.66}$$

demonstrate that the integral of ψ and the first integral moment of ψ are independent of time.

In the free molecule regime, an analytical solution, Eqn. (4.68), exists (by solving Eqn. 8.18, p. 234 [14]) for the total particle number evolution over time in the self preserving distribution limit. If Eqn. (4.12) is plugged into Eqn. (4.61), the following relation is found:

$$\frac{dN_T}{dt} = -\frac{1}{2}\left(A_v\left(\frac{3}{4\pi}\right)^{\frac{1}{6}}\left(\frac{6kT}{\rho_b}\right)^{\frac{1}{2}}\right)(a)N_T^{\left(2-\frac{1}{6}\right)}\phi^{\frac{1}{6}} \tag{4.67}$$

where

$$a = \int_0^\infty\int_0^\infty \left(\frac{1}{\eta_i} + \frac{1}{\eta_J}\right)^{\frac{1}{2}}\left(\eta_i^{\frac{1}{3}} + \eta_j^{\frac{1}{3}}\right)^2 \Psi_i\Psi_j d\eta_i d\eta_j$$

is a dimensionless constant, whose value must be calculated once the dimensionless self-preserving distribution, Ψ_i has been determined. Wu and Friedlander [35] have published a value

for a of 6.552 for spherical particles in the free molecular regime. Eqn. (4.67) may be integrated and N_T solved for to obtain Eqn. (4.68).

$$N_T(t) = \frac{N_T(0)}{\left[1 + \frac{A_v 5}{6}\left(\frac{a}{2}\right)\left(\frac{3\phi}{4\pi}\right)^{1/6}\left(\frac{6kT}{\rho_p}\right)^{1/2} N_T(0)^{5/6}\, t\right]^{6/5}} \tag{4.68}$$

Eqn. (4.68) is often expressed in terms of a dimensionless time, (for example Ref. 29 uses this expression)

$$\frac{1}{\tau_{FM}} = A_v\left(\frac{3\phi}{4\pi}\right)^{1/6}\left(\frac{6kT}{\rho_p}\right)^{1/2} N_T(0)^{5/6}, \tag{4.69}$$

to yield

$$\frac{N_T(t)}{N_T(0)} = \frac{1}{\left[1 + \frac{5}{6}\left(\frac{a}{2}\right)\frac{t}{\tau_{FM}}\right]^{6/5}}. \tag{4.70}$$

In the continuum regime, an analytical solution, Eqn. (4.71), also exists for the total particle number, $N_T(t)$.

$$N_T(t) = \frac{N_T(0)}{\left[1 + \frac{2kT}{3\mu}(1 + a \cdot b) N_T(0)\, t\right]} \tag{4.71}$$

a is 0.9046 and b is 1.248. In the above equations, k is Boltzmann's constant, T is absolute temperature, μ is gas viscosity, and ρ_p is particle material density. Note that in both expressions, when the right-hand term in the denominator is large compared to 1, the expression is independent

of the initial particle concentration. Tabulations of the self preserving distribution functions maybe found in ref [29].The distributions are plotted here in Figure 4.9.

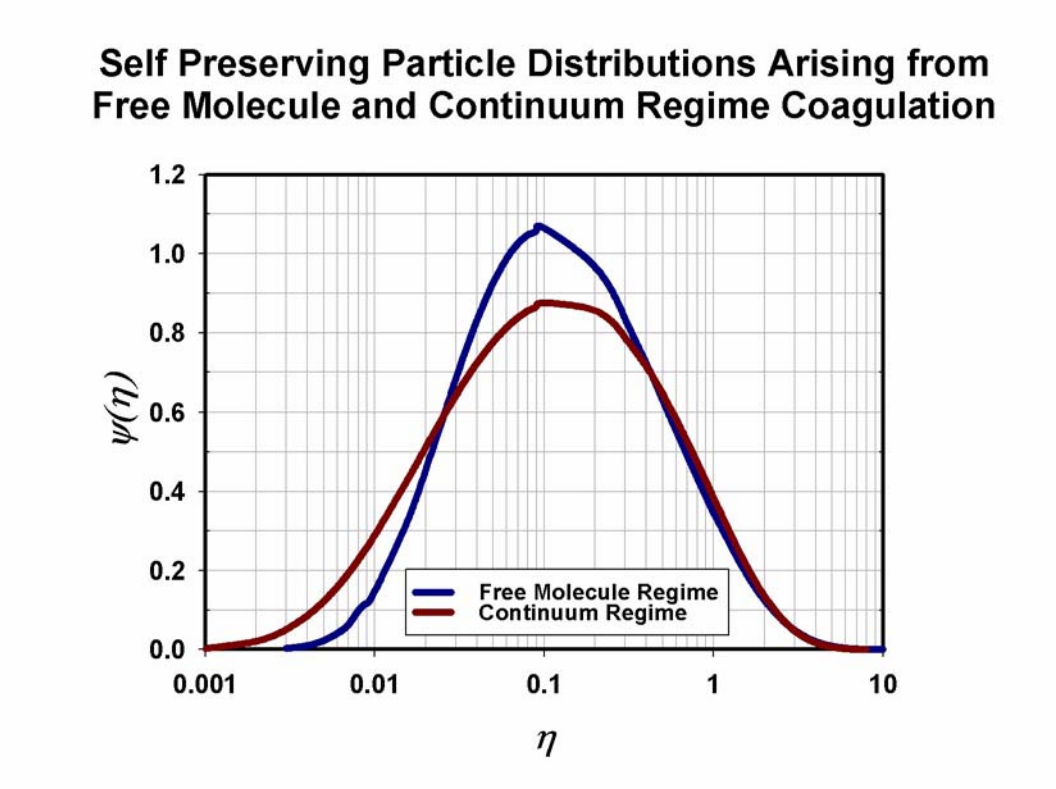

Figure 4.9 Self-preserving particle number distributions [29].

There are two time scales of interest for the test problem. The first is how long it takes to get to the self-similar distribution and the second is how long it takes the expressions for the number concentration of the self-similar distribution to become independent of the initial number concentration. The first time scale, the time to get to a self-similar distribution from an initially mono disperse distribution is reported by Friedlander [2] for the free molecule regime as

$$\tau_{SPF} = \frac{5}{A_v\left(\frac{3\phi}{4\pi}\right)^{1/6}\sqrt{\frac{6kT}{\rho_p}}\,N_T(0)^{5/6}} = 5\tau_{FM}, \tag{4.72}$$

and for the continuum regime as

$$\tau_{SPC} = \frac{39\mu}{2kT\ N_T(0)} \quad . \tag{4.73}$$

Going from an initially poly-disperse distribution will take greater or lesser time depending on how different the initial distribution is from the self-similar distribution. Sometimes the time can be much greater. For a free-molecular self-preserving distribution, the geometric standard deviation of a self-preserving distribution is 1.46 [29]. Starting with an initial lognormal distribution of 2.5 will take about 3 orders of magnitude longer to get to a self preserving distribution than from an initially mono disperse distribution. Similar results are expected for the continuum regime.

Time for the concentration decline to become independent of the initial concentration (to within 1%) for the free molecule regime is

$$\Delta t_F = \frac{100}{\frac{5}{6}\left(\frac{a}{2}\right)\left(\frac{3\phi}{4\pi}\right)^{1/6}\sqrt{\frac{6kT}{\rho_p}}\,N_T(0)^{5/6}} = \frac{100}{\frac{5}{6}\left(\frac{a}{2}\right)}\tau_{FM}, \tag{4.74}$$

and for the continuum regime is

$$\Delta t_C = \frac{100}{\frac{2kT}{3\mu}(1 + a \cdot b)N_T(0)} \quad . \tag{4.75}$$

4.7.1 Free-Molecular Regime Test Problem

In the free molecule regime, consider an initial distribution of mono-disperse 1 nm diameter particles with initial concentration of 1×10^{16} per cm^3, a bulk material density of 4.2 grams per cm^3, at a temperature of 1800 K. These conditions are used in Vemury and Pratsinis [29]. Assuming a molecular weight of 26 kg kmol^{-1}, and bulk-particle molar volume of 0.00619 m^3 kmol^{-1}, the volume fraction of particles in the gas, ϕ, is 5.24x10^{-6} (volume concentration of 5.24 cm^3 per m^3). For these number, the characteristic time constant, τ_{FM}, Eqn. (4.69) is equal to 0.7508×10^{-6} sec. The time to get to a self-preserving distribution is 3.5×10^{-6} seconds. The time for the number concentration decline to become independent of the initial concentration is 27×10^{-6} seconds. In less than 30×10^{-6} seconds, this problem should give a self preserving distribution with a number concentration decline independent of the initial concentration. In 10×10^{-3} seconds, the concentration will be 3.3×10^{10} particles per cm^3.

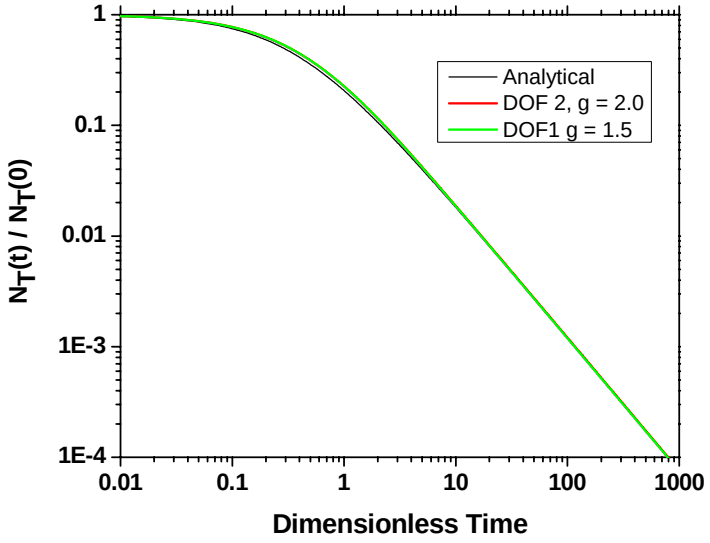

Figure 4.10 Self-preserving particle distribution in the free-molecular regime. Dimensionless particle number vs. the dimensionless time. DOF 2 and DOF 1 cases vs. the analytical solution.

Figure 4.10 contains the results for the dimensionless particle number vs. the dimensionless time, t/τ_{FM}. Agreement is excellent between the calculation and the analytical result, Eqn. (4.70), for both the DOF1 and DOF2 cases.

Results for the nondimensional particle distributions are presented in Figure 4.11. Even on a relatively coarse grid with a geometric spacing of 2.0 the DOF 2 implementation exhibits very nice agreement with the analytical results (black) given in [29]. The degree-of-freedom 1 (DOF 1) method is also shown. Results for geometric spacings of 1.5 and 1.2 indicate that there is seems to be convergence towards the analytical results. However, a much finer resolution is needed to get good agreement with the DOF-1 method, obviously. The 1.2 DOF 1 method had about 35 bins encompassing the distribution in Figure 4.11with 150 bins in the entire simulation, while the 1.5 DOF-1 method had 15 bins in the distribution with 100 bins in the entire simulation. The 2.0 DOF 2 method had 10 bins with 20 degrees of freedom in the distribution in Figure 4.11 with 40 bins in the entire simulation.

4.7.2 Continuum Regime

In the continuum regime, consider an initial distribution of mono-disperse 0.5 μm diameter particles with an initial concentration of 10^{10} per cm^3 in a gas at 300 K with a viscosity of

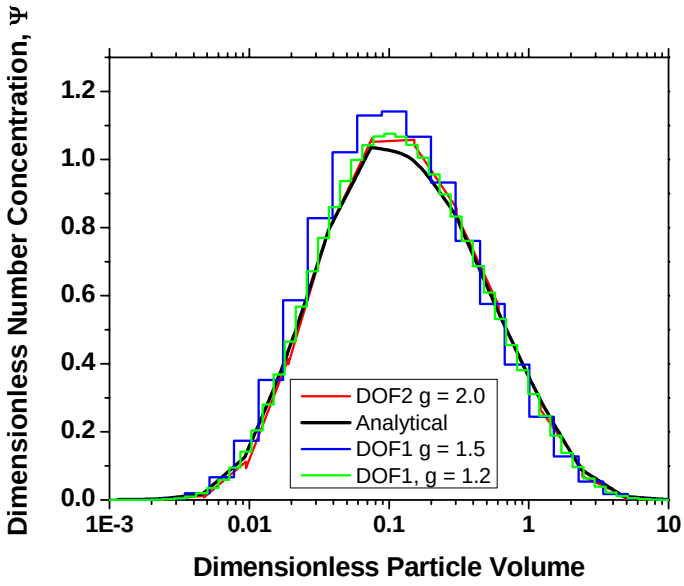

Figure 4.11 Results for self-preserving distribution in the free-molecular regime. Distributions of particle number concentration are shown at t = 0.01.

1.81×10^{-4} poise. The volume fraction ϕ is 6.55×10^{-4} (volume concentration of 655 cm^3 per m^3). This is a very high volume concentration but for the mathematical purposes here it is suitable. For this situation, the time to get to self preserving is 8.5 seconds. The time for the number concentration decline to become independent of the initial concentration is 31 seconds. In less than 40 seconds, this problem should give a self-preserving distribution with a number concentration decline independent of the initial concentration. One sees that this takes much longer than the free molecule case above because the initial concentrations are lower by six orders of magnitude in order to maintain realistic conditions. In 300 seconds, the concentration will be 1.03×10^7 particles per cm^3 and the average diameter will be 4.96 μm.

5. Nucleation

5.1 Definition of Nucleation

Nucleation refers to particles that are formed via condensation of gas phase species. Nucleation begins when thermodynamics favors the condensed phase in particle form over the gas phase. Classically, this is generally associated with a concentration of the nucleating gas phase species in excess of its equilibrium vapor pressure, i.e., a saturation ratio greater than one. Nucleation occurs to relieve this supersaturation. The excess in concentration may arise from cooling of the gas or from gas phase production of the nucleating species from chemical reaction. At the same time monomer and cluster collision is taking place, monomer evaporation from the clusters is occurring. In classical nucleation theory, there exists an equilibrium cluster size distribution determined by the balance between condensation on and evaporation from clusters at some reference condition. Nucleation occurs when there is net growth of some critical cluster size in this distribution. The formation rate of these critical clusters is the nucleation rate and the size of the critical cluster is the nuclei size. Classical nucleation theory and closely related formulations provide a prediction of the size of the critical clusters and the particle "current" through that size which constitutes the nucleation rate. Within combustion problems, however, due to the absence of Kelvin Effect type influences, critical cluster sizes occur at the molecular sized level, i.e., where the speciation is handled via direct inclusion of species (i.e., polyaromatic hydrocarbons) in the molecular mechanism. Thus, an alternative approach to the classical nucleation formulation within `CADS` has been implemented wherein bimolecular reactions between precursor PAH's lead to formation of the initial particles. Classical nucleation formulations may be added later in `CADS`, as the need warrants.

Within `CADS`, nucleation reactions are defined as reactions that occur between molecular species and particles that create and/or destroy particles. Typically, if all of the reaction participants on one side of the reaction sign are molecular species and one or more participants on the other side of the reaction sign is a particle, then that reaction is considered a nucleation reaction and is included in the nucleation mechanism. Note, this means that reactions which destroy particles are also considered nucleation reactions. Nucleation reactions should be contrasted with surface growth (i.e., condensation) reactions. Within `CADS`, all reactions are considered either to be nucleation, condensation or bulk-particle reactions. Condensation reactions typically don't lead to the change in the number density of particles, and the rates of condensation reactions are proportional to the surface area of the particles. Nucleation reactions necessarily lead to the change in the number density of particles, and may exhibit a variety of rate dependencies with respect to the particle number density and therefore the particle surface area. The chief dependency/feature is that nucleation reactions may be independent of the number density of particles.

There is one possibility where condensation reactions can also lead to the change in the number of particles. This case is the etching of particles from the zeroth sectional bin. An optional modification to the normal condensation algorithm has been added to allow for the destruction of particle number density due to etching of particles in the zeroth sectional bin. This is carried out while conserving elements, but under represents high molecular weight products of particle etching reactions. The alternative to this optional modification would involve adding in an additional "nucleation/etching" reaction to the nucleation mechanism involving particles in the zeroth order bin as a reactant that would create a single or multiple product molecular species (of a relatively high molecular weight) as the result of the etching reaction.

The Kelvin equation takes into account the effects of surface tension on the stability of small particles. Because the previous implementation is completely reversible, it should be possible to include the effects of the Kelvin equation in the nucleation rates of particles. The Kelvin equation necessarily includes a modification to all reversible reactions that change the size of a particle.

5.2 Particle Species

In order to represent the possibly reversible nature of nucleation reactions, within nucleation reactions, particles are represented by "particle species." Particle species are individual molecular species, whose concentrations exhibit a duality with the sectional bin particle unknowns. Because they are "species" from Cantera's point of view, Cantera's existing mechanism for calculating rate of progress for kinetic reactions and rates of formation of species may be used in the calculation of nucleation reactions, where one side of the reaction expression actually involves a distribution of molecular sizes represented by a distributed function of the total species size.

Figure 5.1. graphically depicts this duality. Particle species maintain two different ways for specifying its elemental composition. In the first way, a particle species maintains an elemental composition list just as an other molecular species does. In the other way, particle species maintain a stoichiometric composition vector based on the monomer units of the particle phase. As Figure 5.1 illustrates, their composition can be represented in terms of the total monomer unit number, ν ,and the monomer unit composition, for multi-species particle phases. Also, the rate of production of particle species can be directly related to the rate of production of total monomer units, monomer units types and total particles. This is done by assigning particles species to individual section bins. Particle species below the cutoff of the zeroth section bin are assigned to the zeroth sectional bin.

The end result of the nucleation operator are source terms for the gas-phase molecular species, $\boldsymbol{S}^{\text{nucl\_gas}}$ and for each section L of the particle distribution, $\boldsymbol{S}^{\text{nucl\_part},\,L}$. The sum of the gas and particle nucleation source terms conserves mass and is faithful to the prescribed change in the par-

ticle number density, detailed in the nucleation kinetics mechanism. This is true for the 2 DOF formulation and frequently true for the 1 DOF formulations. In the 1 DOF formulation, a tie-line approach is used to apportion the particle species source terms between 2 sectional bins. The tie-line approach solves a 2x2 equation system based on the total number of monomer units and the total desired change in particle numbers for each particle species.

For nucleation reactions that are reversible or that involve the irreversible destruction of particle species, a concentration of particle species must be determined in order to evaluate the production rate of the particle species from a reaction rate. This is achieved by constructing an algorithm for determining the concentration of a particular particle species from the monomer unit distribution function, $q(\mu, t)$ of the section L, corresponding to the particle species. The default behavior is to assign δ_{MU} of the available particle concentration around the total monomer unit value of the particle species, μ, to the concentration of a particular particle species. The exact equation for the concentration (kmol m^{-3}) of particle species k, $C_{ps}[k]$, is Eqn. (5.1).

$$C_{ps}[k] = \frac{q_L(\mu, t)\delta_{MU}^{L}(fit_{MU})}{\mu_{ps}} \tag{5.1}$$

$q_L(\mu, t)$ is the total monomer unit distribution function in section L, the section containing particle species k. fit_{MU} is limiting factor between 0 and 1 that takes into account of the elemental sto-

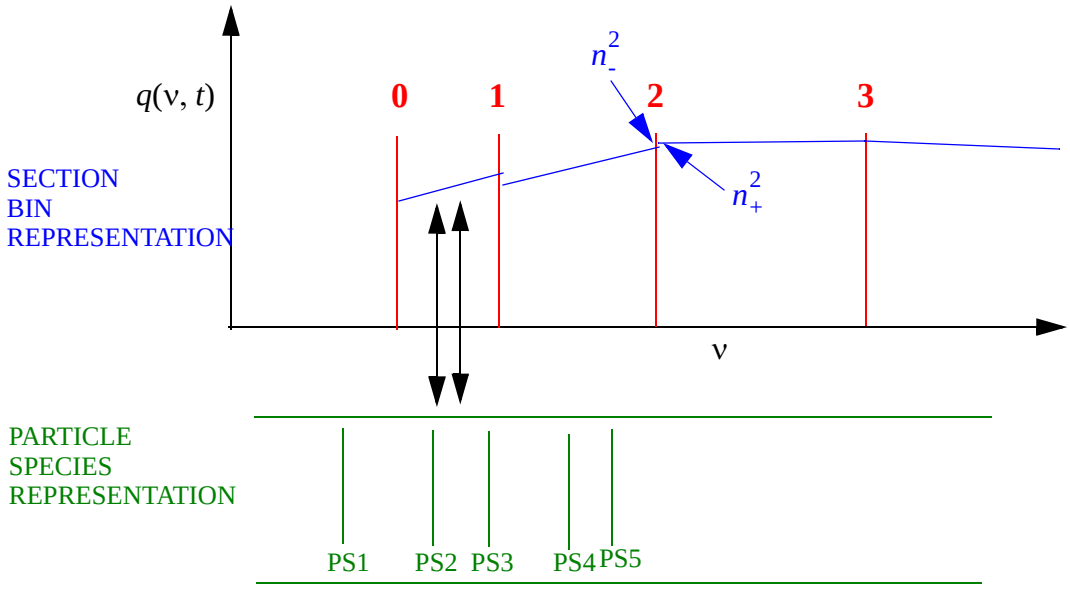

Figure 5.1 Duality between particle species and section bins. Nucleation reactions are written in terms of particle species. These, in turn can be represented by particle unknowns.

ichiometry between the particle species and the elemental composition of the section. For example, if the sectional bin is made up of particles consisting of SiO_2 and the particle species contains carbon, then $fit_{MU} = 0$. If the sectional bin is made up of purely carbon and the particle species is pure carbon, then $fit_{MU} = 1$. μ_{ps} is the total monomer units in the particle species. δ_{MU}^{L} is equal to the following formula:

$$\delta_{MU}^{L} = min\left(5, \frac{\nu_H^L - \nu_L^L}{num_{ps}^L}\right) \tag{5.2}$$

where num_{ps}^{L} is the number of particle species in section L. Though far from rigorous, Eqn. (5.2) attempts to create a representative range of μ values for which concentration of the particular particle species accounts for. In particular if there are many particle species in a single bin, then the concentration of any single particle species should be reduced to reflect this fact. The cap on 5 monomer units in Eqn. (5.2) reflects the following thought. If there is only one particle species in a very large sectional bin, encompassing a large range of total monomer units, then the concentration of the particle species should probably only reflect the concentration of particles around its own total monomer unit concentration and not the entire concentration of particles in the section.

5.3 Cantera Implementation

Much of the mechanics of calculating production rates and thermodynamic quantities for the nucleation process is carried out by the normal lower-level functions of Cantera. However, in order to use these functions, wrappers around Cantera's thermodynamics objects and kinetics objects that take into account the sectional bin representation must be constructed. Figure 5.2 pictorially describes these wrapper objects involved and their interrelationships within the `CADS` calculation of the particle nucleation rate.

A central role is played by the `PartSpecPhase` object. The `PartSpecPhase` object calculates the concentrations and thermodynamics for the particle species, i.e., the full range of features of the parent, Cantera's `ThermoPhase` object. Conceptually, all of the normal thermodynamics functions declared in the `ThermoPhase` object are implemented by the `PartSpecPhase` object. The concentration of particle species is needed by the kinetics algorithms for reversible nucleation reactions or for reactions involving particle break-up into molecular species. The evaluation of the concentration of particle species is handled by the `PartSpecPhase` object, as described in the previous section.

5.3.1 Calculation of Thermodynamic Species for Particle Species

Two basic options are provided for calculation of the thermodynamic functions for particle species. The first option uses the particle solid-phase thermodynamic values to calculate the thermodynamic functions for each particle species. It does this by referencing the thermodynamics object for the particle solid phase through a reference pointer, `m_ParticleSolidPhase`. First it calculates all of the reference thermodynamic functions for the monomer units of the solid phase at the current temperature using `m_ParticleSolidPhase.` Then, the values are combined with the stoichiometric composition of the particle species in terms of the monomer units of the solid phase to derive the molar thermodynamic functions for the particle species. The stoichiometric composition of each particle species is determined using a basis set of monomer unit species for the solid phase [34]. The basis set is defined as a minimal number of monomer units types whose elemental composition can span the range of all possible elemental compositions of all monomer unit types. Note, this number is frequently, but not always, equal to the number of elements defined in the monomer units of the solid particle phase. The default set of basis monomer unit species, as well

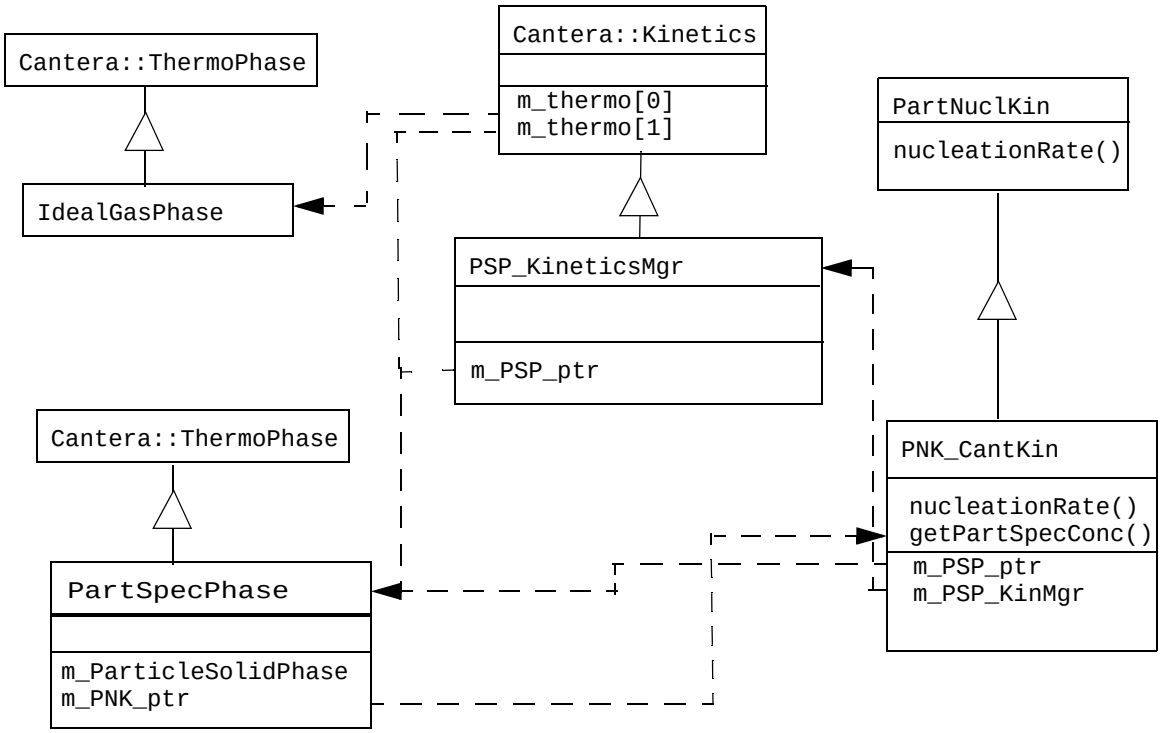

Figure 5.2 Glyph diagram of the implementation of nucleation within the Cantera framework. Triangles represent inheritance relationships with the pointy end directed at the parent object.Dotted arrows depict dependencies; one object maintains a pointer to another object in order to carry out a needed function.

as the stoichiometric composition of each particle species, are determined algorithmically (e.g., see ref [34]). However, since this affects the calculation of the thermodynamic properties, the default set can be specified in the input XML file using the optional `defaultSpeciesBasis` XML node option within the Particle Species `thermo` XML element specification. For example, suppose there were 2 species in the soot solid phase, `C_new` and `C_old`, each having an elemental composition of one carbon atom, but perhaps have different thermodynamic reference properties. Then, the following XML snippet, which sets up a `PartSpecPhase` object, could be added to the particle species XML element to resolve the ambiguity in specification of the default thermodynamic properties.

```
<!-- “phase” contains elements, thermo, species, transport and kinetics descriptions
      of the phase -->
<phase id=”PartSpecPhaseExample, dim=”3”
   <!-- “thermo” describes the thermodynamic model for the ThermoPhase object, and
         provides parameters for that model -->
   <thermo model=”PartSpecPhase”>
      <condensedPhaseThermo> sootPhase </condensedPhaseThermo>
      <defaultSpeciesBasis>
          C_new
      </defaultSpeciesBasis>
   </thermo>
</phase>
```

The model attribute of `PartSpecPhase` in the `thermo` XML node specifies that the phase will be a Particle Species phase. The `condensedPhaseThermo` node specifies the condensed phase that will be used by the Particle Species phases to extract the basis set of condensed phases species from. In this example, we have defined the default monomer units basis species for the solid phase to be `C_new`. Note, either `C_new` or `C_old` could have been chosen since they have the same elemental composition. Then, if a particle species, named C18, consists of 18 carbon atoms, the enthalpy for C18 would be determined by the following relation:

$$h_{\mathrm{C18}}(T, P_o) \;=\; 18 h_{\mathrm{C\_new}}(T) \tag{5.3}$$

Another option can be used to specify the stoichiometry of a particle species. Within the species section of the XML file, the species basis may be specified exactly, using the `speciesBasisArray` XML element. For example, the C18 particle species mentioned above could be specified to consist of a unique number of `C_old` and `C_new` species, using the following XML code snippet:

```
<!-- “species” describes a single species in a phase. It contains the element
   composition, standard state thermodynamic specification, and transport props -->
<species id=”Part_C18 dim=”3”>
   <!-- “atomArray” contains the atomic composition -->
   <atomArray> C:18 </atomArray>
   <speciesBasisArray>
      C_new:11
      C_old:7
```

```
    </speciesBasisArray>
  </species>
```

This species XML element does not need a thermo XML element, since the thermodynamic specification of the species will be generated from a linear combination of the `C_new` and `C_old` thermodynamic values.

Alternatively, specific thermodynamic polynomials may be supplied for each specified particle species. These uniquely define the thermodynamic functions for the particle species irrespective of the thermodynamics for the ideal solid solution phase. This is done in the species XML element in the normal way. For example the previous example could be modified to be the following:

```
<species id=”Part_C18 dim=”3”>
   <atomArray> C:18 </atomArray>
   <speciesBasisArray>
      C_new:11
      C_old:7
   </speciesBasisArray>
   <thermo>
      <NASA P0="100000.0" Tmax="1000.0" Tmin="200.0">
         <floatArray size="7" title="low"> 2.344331120E+000,
            7.980520750E-003,-1.947815100E-005, 2.015720940E-008,
            -7.376117610E-012, -9.179351730E+002, 6.830102380E-001
         </floatArray>
      </NASA>
      <NASA P0="100000.0" Tmax="3500.0" Tmin="1000.0">
         <floatArray size="7" title="high"> 3.337279200E+000,
            -4.940247310E-005, 4.994567780E-007, -1.795663940E-010,
            2.002553760E-014, -9.501589220E+002, -3.205023310E+000
         </floatArray>
      </NASA>
   </thermo>
</species>
```

The specification of the thermo XML element preempts alternative thermodynamic function calculations previously discussed.

In all cases, the pressure dependence of the thermodynamic polynomials then follows the standard ideal gas law form. For example, the Gibbs energy, *G*, has the following pressure dependence:

$$G_{\mathrm{C18}}(T, P) \;=\; G_{\mathrm{C18}}(T, P_o) + RT\ln(P/P_o)\,. \tag{5.4}$$

Currently, `PartSpecPhase` assumes that the particle species exist in an ideal gas, but are dilute in that medium. The total sum of the species in the `PartSpecPhase` is not constrained to sum up to the total concentration given by the ideal gas law. Though never explicitly checked, this sum is implicitly assumed to be much less than one.

5.3.2 Calculation of Kinetic Rates of Reaction

The kinetics object that calculates the net rate of production of particle species is called `PSP_KineticsMgr`. It is derived from the Cantera class `Kinetics` and refers to the `CADS` object class `PartSpecPhase` and the gas phase medium, `IdealGasPhase`, through pointers. The class, `PNK_CantKin`, derived from `PartNuclKin` handles the mapping of the particle species rates of formation into the rates of production of unknowns specific to the sectional bins representation. Some of the code that is not Cantera-specific is included in the class `PartNuclKin`. Code that is specific to Cantera is included in virtual functions defined in the class `PartNuclKin` and reimplemented in the derived class, `PNK_CantKin`.

For example, the core routine, `nucleationRate(),` is a virtual function defined in `PartNuclKin`. It is defined explicitly in the class `PNK_CantKin`. In order to calculate the nucleation rate, `PartNuclKin` first translates the current sectional bin concentrations into concentrations of particle species. `PNK_CantKin` handles the calculation of the concentration of the particle species via Eqn. (5.2). Then, it calls `PSP_KineticsMgr` to calculate the rates of production of gas-phase species and particle species using the normal Cantera kinetics procedure. It then reinterprets the rates of particle species production in terms of rates of production of particle section bin concentrations.Damping algorithms to maintain positivity in the DOF 2 implementation are also located in the `PartNuclKin` class. These are described in a later section.

5.4 Implementation in the Input File

The master input file contains a block pertaining to the nucleation operator. Figure 5.3 contains the nucleation operator section of the particle input file. The `METHOD` command line has two values, `Cantera` or `NONE`. This serves to toggle the nucleation operator. The next command line supplies the file name for the XML commands specifying the particle phase and the nucleation reactions. The last command line, `Section Addition Adjust Method`, specifies the algorithm to employ to stabilize the nucleation source term. This is discussed in a subsequent section.

The nucleation kinetics equations are contained in the particle species file. Figure 5.4 contains an example of a particle species XML file. Figure 5.5 is a continuation of that example showing sample nucleation reactions involving the particle species.

Figure 5.4 displays several different ways that particle species can be declared in the input file. First, they can be explicitly declared. The XML element, `speciesArray,` defines what explicit particle species is in the phase and where to get information about the species. The attribute

```
START BLOCK ADS MODEL DEFINITION
     ....
     START BLOCK NUCLEATION MODEL
        ! METHOD = {<Cantera>| NONE}
        METHOD = Cantera
        ! Cantera Particle Nucleation Phase File = {string}
        !                Default name = PartNuclPhaseFile.xml
        Cantera Particle Nucleation Phase File = partNucl.xml
        ! Section Addition Adjust Method = [ Conservative | <Ramp> |
        !                Always Accurate ]
        Section Addition Adjust Method = Ramp
     END BLOCK NUCLEATION MODEL
     ....
END BLOCK ADS MODEL DEFINITION
```

Figure 5.3 Nucleation block of the particle input file

`datasrc` defines where to find information about the species. It's value "`#species_PartSpec`" means that the data should be found in the current file, as children of the XML element named `speciesData`. Alternative, if the value "`file.xml#species_PartSpec`" were specified, then the file named, file.xml, would be searched for XML elements `speciesData` in order to find the species data.

The thermodynamic model for the particle species phase is defined to be `PartSpecPhase`, that causes the implementation of the `PartSpecPhase` class described in section 5.3. What this means is that thermodynamics for particle species that don't have explicitly defined thermodynamic functions is obtained from the particle solid phase. The ID of the particle solid phase to obtain the thermodynamic functions from is given in the child XML element, `condensedPhaseThermo`. "`sootPhase`" is the name of the condensed phase thermodynamics description for the particles. In that phase, the MU types, `C-new` and `C-old`, are defined, with thermodynamic functions assigned to these species. The condensed phase is currently always assumed to be an ideal, constant volume, ideal solution.

The `elementArray` XML element specifies what elements are defined to be in the particle phase and where to get information about them, i.e., the file "elements.xml". Elements don't have to be in any particular order.

The kinetics model is defined to be `ParticleNucleationKinetics`. This means that the `PSP_KineticsMgr` is used to calculate the reaction rates. The `phaseArray` XML element indicates what other phases are involved in the calculation of reaction rates within `PSP_KineticsMgr`. The phase, `ethene`, is the ID of an ideal gas phase containing the `H2`, `A1`, and `A3` species that appear in the kinetic reaction expressions. The XML element `reactionArray` contains the database

specification, i.e., where the reactions are to be found. The `datasrc` attribute, "`#reactions_Nucleation`" means that the nucleation reactions should be found in the XML element, `reactionData`, with id "`reactionsNucleation`" within the current file.

The species, `Part_12`, `Part_20`, and `Part_23` are declared explicitly in the phase. `Part_12` has explicit thermodynamic functions assigned to it. `Part_20` doesn't have thermodynamic functions

```
<phase id="PartSpecPhaseExample" dim="3">
  <thermo model="PartSpecPhase">
     <condensedPhaseThermo> sootPhase </condensedPhaseThermo>
     <defaultSpeciesBasis> C-new </defaultSpeciesBasis>
  </thermo>
  <elementArray datasrc="elements.xml"> H C O N Ar </elementArray>
  <speciesArray datasrc="#species_PartSpec" rule="create_on_demand">
     Part_12 Part_20 Part_23 </speciesArray>
  <reactionArray datasrc="#reactions_Nucleation" />
  <kinetics model="ParticleNucleationKinetics" />
  <phaseArray> ethene </phaseArray>
</phase>

<speciesData id="species_PartSpec">
  <!-- Particle species specified by its own input thermo -->
  <species name="Part_12">
     <atomArray> C:12 </atomArray>
     <speciesBasisArray> C-new:12 </speciesBasisArray>
     <thermo>
        <NASA P0="100000.0" Tmax="1000.0" Tmin="200.0">
           <floatArray size="7" title="low">2.344331120E+000, 7.980520750E-003,
           -1.947815100E-005, 2.015720940E-008, -7.376117610E-012,
           -9.179351730E+002, 6.830102380E-001</floatArray>
        </NASA>
        <NASA P0="100000.0" Tmax="3500.0" Tmin="1000.0">
           <floatArray size="7" title="high">3.337279200E+000, -4.940247310E-005,
           4.994567780E-007, -1.795663940E-010, 2.002553760E-014,
           -9.501589220E+002, -3.205023310E+000</floatArray>
        </NASA>
     </thermo>
  </species>
  <species name="Part_20">
     <atomArray> C:20 </atomArray>
     <speciesBasisArray> C-new:20 </speciesBasisArray>
  </species>
  <species name="Part_23">
     <atomArray> C:23 </atomArray>
  </species>
</speciesData>
```

Figure 5.4 Example of an XML particle-species file

assigned to it; it obtains its thermo from the particle solid phase. `Part_20` defines the MU species basis, however, (e.g., consisting of 20 `C_new` MU's) so that the thermodynamics functions are deliberately specified. `Part_23` only specifies its stoichiometry, leaving the specification of the MU species basis up to a default algorithm. In this case, the default MU species basis is defined to be `C-new` by the `defaultSpeciesBasis` XML element of the `thermo` XML element, see Figure 5.4.

However, another particle species is defined implicitly through the first nucleation reaction in Figure 5.5. In that reaction, the product "Particle" is created. This is a keyword indicating that the particle species should be created "on the fly", if an existing particle species doesn't have the required elemental composition. The required elemental composition in this case, noting that A1 has C6H6 and A3 has C14H10 stoichiometry, is one with 28 carbons. Thus, a new particle species is created in the phase with 28 carbons, with its thermodynamics specified by the default generat-

```
<reactionData id="reactions_Nucleation" model="ParticleNucleationKinetics">
  <!--
     The Particle species entry will automatically create a
     new species in the PartSpecPhase phase with the right stoichiometry.
  -->
  <reaction id="nucleation_rxn_1" reversible="no">
     <equation>A3 + A3 =] Particle + 10 H2</equation>
     <reactants>A3:2</reactants>
     <products>Particle:1 H2:10</products>
     <rateCoeff>
        <Arrhenius order="2">
           <A units="cm3/mol/s">1.2E23</A>
           <b>-2.92</b>
           <E units="cal/mol">15890.</E>
        </Arrhenius>
     </rateCoeff>
  </reaction>

  <reaction id="nucleation_rxn_2" reversible="yes">
     <equation> A1 + A3 [=] Part_20 + 8 H2</equation>
     <reactants> A1:1 A3:1 </reactants>
     <products> Part_20:1 H2:8 </products>
     <rateCoeff>
        <Arrhenius order="2">
           <A units="cm3/mol/s">1.2E23</A>
           <b>-2.92</b>
           <E units="cal/mol">15890.</E>
        </Arrhenius>
     </rateCoeff>
  </reaction>
</reactionData>
</ctml>
```

Figure 5.5 Example of an XML particle-species file (continued)

ed MU species basis of the particle solid phase. The XML element `speciesArray` contains an attribute, `rule`, which can be used to toggle this automatic particle species creation capability. If the attribute `rule` with the value "`create_on_demand`" isn't present then the auto-generation capability won't be enabled.

In Figure 5.5, all of the XML entries that specify the individual reaction kinetics are boiler-plate Cantera entries, except for the `create_on_demand` particle species, which auto generates species, as described previously. Note, however, the `equation` XML element doesn't specify any attributes to the reaction; it's only a string representation of the reaction and is ignored. The `reactants` and `products` XML elements and the `reversible` attribute specify the actual properties of the kinetic reaction.

5.5 Numerical Stability Considerations

Numerical testing has demonstrated that filters for the nucleation source term are necessary in order to maintain numerical robustness. Here and elsewhere within `CADS`, numerical robustness is defined as maintaining positivity of all coefficients in the sectional particle distribution unknowns for the DOF 2 case no matter how large a time step is taken, when an implicit time-stepping algorithm is used. If the nucleation source term has a monomer unit to particle ratio which is too close to one end of a bin, a skewed distribution with a negative sectional coefficient may result. The DOF 1 case doesn't exhibit similar problems. The same approach is used in the nucleation term as has been previously described elsewhere. Figure 5.6 depicts the filtering carried out on the nucleation source term.

$\boldsymbol{S}^{\text{nucl\_part},\,i}$ represents the source term vector for nucleation. Its form is given by Eqn. (5.5). The ratio of the total monomer unit (*TMU*) term to the particle source term yields the monomer unit per particle value of the source term, MU. Each sectional bin, *i*, in the DOF 2 implementation can afford a certain range of MU values in its intersectional distribution, while still maintaining positive basis coefficients. The high and low limit for this range is specified by the values MU_H^i and MU_L^i. Values of average number of monomer units in the particles of a section, MU_ν^i, outside of this range (i.e., either $\mathrm{MU}_\nu^i < \mathrm{MU}_L^i$ or $\mathrm{MU}_\nu^i > \mathrm{MU}_H^i$) for the section, may cause one of the sectional basis coefficients, b_i^L or b_i^H (where $q^i(\mu) = b_i^L \Theta_i^L(\mu) + b_i^H \Theta_i^H(\mu)$ for the DOF 2 representation), potentially to go negative.

$$\boldsymbol{S}^{\text{nucl\_part},\,i} = \begin{bmatrix} S_{part}^{\text{nucl\_part},\,i} \\ S_{TMU}^{\text{nucl\_part},\,i} \\ S_{MU1}^{\text{nucl\_part},\,i} \\ \cdots \\ S_{NMU-1}^{\text{nucl\_part},\,i} \end{bmatrix} \tag{5.5}$$

The nucleation source term's monomer unit per particle average, $\text{MU}^{i}_{S^{nucl}}$, is equal to $S_{TMU}^{\text{nucl\_part},\,i} / S_{part}^{\text{nucl\_part},\,i}$. When $\text{MU}^{i}_{S^{nucl}}$ is outside the range of MU^{i}_{L} to MU^{i}_{H}, difficulties with one of the sectional basis coefficients going negative may or may not occur depending on a range of factors associated with the other source terms, the existing values of the sectional basis coefficients, the value of the time step, and the time-stepping algorithm. For example, there may be other source terms to section *i* such that the net source terms has an acceptable overall monomer unit per particle ratio. If the existing sectional basis functions are positive, for time dependent calculations and for short enough times, the sectional basis functions will maintain their positivity.

However, Figure 5.6 shows a case where the nucleation source term must be reapportioned between the neighboring section in order to maintain the positivity of b_i^L no matter what the time step value. In this case, $q_L^i = b_i^L = 0$ and $\text{MU}^{i}_{S^{nucl}} > \text{MU}^{i}_{H}$. Barring other source terms, and as-

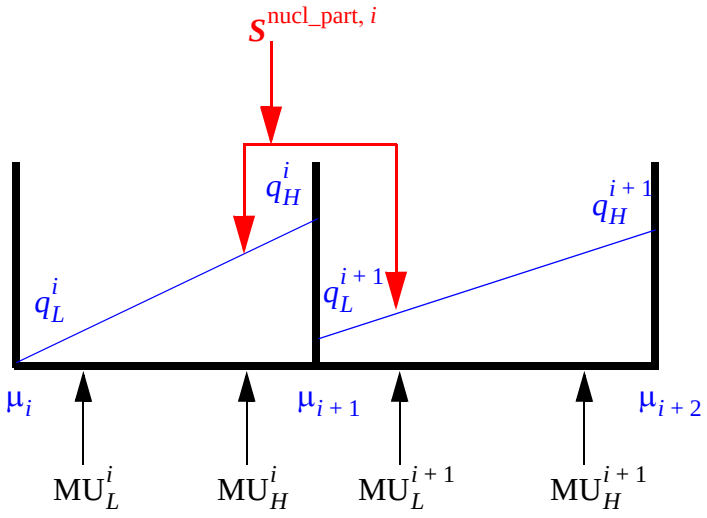

Figure 5.6 DOF 2 filtering of the nucleation source term vector. The source term is apportioned to neighboring section in order to maintain positivity of the basis coefficients.

suming that the nucleation source terms are positive, this will force:

$$\frac{db_i^L}{dt} < 0 \text{ and } \frac{db_i^H}{dt} > \frac{d(b_i^L + b_i^H)}{dt} > 0$$

The following algorithm may be used to avoid this instability. For the particular case where $\mathrm{MU}_{S^{nucl}}^{i} > \mathrm{MU}_{H}^{i}$, as shown in Figure 5.6 (other cases are handled in a similar fashion), a portion of the $\boldsymbol{S}^{\mathrm{nucl\_part},\,i}$ vector is redirected to the higher section, i+1. Thus, the source term, $\boldsymbol{S}^{\mathrm{nucl\_part},\,i}$ is split into two parts, each part having a different and more stable value for $\mathrm{MU}_{S^{nucl}}$ according to that section's parameters. The original section's adjusted monomer unit ratio is denoted as MU_{adj}^{i}, while the neighboring section's adjusted monomer unit ratio is denoted as MU_{adj}^{i+1}.The following value for these quantities are used:

$$\mathrm{MU}_{adj}^{i} = \mathrm{MU}_{H}^{i} - f_{\mathrm{mu\_adj}}^{safety}(\mathrm{MU}_{H}^{i} - \mathrm{MU}_{L}^{i}) \tag{5.6}$$

$$\mathrm{MU}_{adj}^{i+1} = \mathrm{MU}_{L}^{i+1} + f_{\mathrm{mu\_adj}}^{safety}(\mathrm{MU}_{H}^{i+1} - \mathrm{MU}_{L}^{i+1}) \tag{5.7}$$

The second term in Eqn. (5.6) and (5.7) is to ensure positivity of sectional basis coefficients even after roundoff errors. $f_{\mathrm{mu\_adj}}^{safety}$ is a constant with a current value of 2×10^{-4}. Then, the following 2 x 2 matrix system representing a partitioning of the original source term for section i that conserves the original total particle and TMU source terms, Eqn (5.8), is solved for the adjusted particle source terms for section i and $ialt = i + 1$, $S_{\mathrm{part},\,i}^{adj,\,i}$ and $S_{\mathrm{part},\,i+1}^{adj,\,i}$.

$$\begin{bmatrix} 1 & 1 \\ \mathrm{MU}_{adj}^{i} & \mathrm{MU}_{adj}^{i+1} \end{bmatrix} \begin{bmatrix} S_{\mathrm{part},\,i}^{adj,\,i} \\ S_{\mathrm{part},\,i+1}^{adj,\,i} \end{bmatrix} = \begin{bmatrix} S_{part}^{\mathrm{nucl\_part},\,i} \\ S_{TMU}^{\mathrm{nucl\_part},\,i} \end{bmatrix} \tag{5.8}$$

$$S_{\mathrm{TMU},\,i}^{adj,\,i} = \mathrm{MU}_{adj}^{i} S_{\mathrm{part},\,i}^{adj,\,i} \qquad S_{\mathrm{TMU},\,i+1}^{adj,\,i} = \mathrm{MU}_{adj}^{i+1} S_{\mathrm{part},\,i+1}^{adj,\,i} \tag{5.9}$$

$$S_{\mathrm{MU\_j},\,i}^{adj,\,i} = (X_{\mathrm{MU\_j}}^{\mathrm{nucl\_part},\,i}) S_{\mathrm{TMU},\,i}^{adj,\,i} \qquad S_{\mathrm{MU\_j},\,i+1}^{adj,\,i} = (X_{\mathrm{MU\_j}}^{\mathrm{nucl\_part},\,i}) S_{\mathrm{TMU},\,i+1}^{adj,\,i}$$

The above description applies to how the tie-line approach is applied to one section, with the alternate source term section being one section higher ($ialt = i + 1$). To formulate a stabilized nucleation source term, this approach is applied to all source terms and then collected:

$$\boldsymbol{S}^{\mathrm{nucl\_part},\,adj} = \sum_i (\boldsymbol{S}_i^{\mathrm{nucl\_part},\,adj,\,i} + \boldsymbol{S}_{ialt}^{\mathrm{nucl\_part},\,adj,\,i}) \tag{5.10}$$

where

$$\boldsymbol{S}_i^{\text{nucl\_part},\,adj,\,i} = \begin{bmatrix} S_{\text{part},\,i}^{adj,\,i} \\ S_{\text{TMU},\,i}^{adj,\,i} \\ S_{\text{MU\_j},\,i}^{adj,\,i} \\ \cdots \\ S_{\text{MU\_nmu-1},\,i}^{adj,\,i} \end{bmatrix} \text{ and } \boldsymbol{S}_{ialt}^{\text{nucl\_part},\,adj,\,i} = \begin{bmatrix} S_{\text{part},\,ialt}^{adj,\,i} \\ S_{\text{TMU},\,ialt}^{adj,\,i} \\ S_{\text{MU\_j},\,ialt}^{adj,\,i} \\ \cdots \\ S_{\text{MU\_nmu-1},\,ialt}^{adj,\,i} \end{bmatrix} \tag{5.11}$$

Here, $ialt = i-1$ or $ialt = i+1$ depending on value of $\text{MU}_{S^{nucl}}^{i}$.

There is one important special case to this procedure. The lowest section, section 0, can not handle $\text{MU}_{S^{nucl}}^{i} < \text{MU}_{L}^{0}$. For this case when the average MU's per particle in the lowest section is however than MU_{L}^{0}, particle conservation is dropped when the adjusted source term is needed for stability purposes:

$$S_{\text{part},\,0}^{adj,\,0} = \frac{S_{TMU}^{\text{nucl\_part},\,0}}{\text{MU}_{adj}^{0}} \tag{5.12}$$

$$S_{\text{TMU},\,0}^{adj,\,0} = S_{TMU}^{\text{nucl\_part},\,0} \tag{5.13}$$

$$S_{\text{MU\_j},\,0}^{adj,\,0} = S_{MU1}^{\text{nucl\_part},\,0}$$

This lowers the numbers of particles actually created, while maintaining elemental conservation. Because the particle coagulation rate is an inverse function of the particle size, this seems to be a fair approximation.

The DOF 1 case shares some similarities with the DOF 2 case. For the DOF 1 case, in essence, $\text{MU}_{L}^{i} = \text{MU}_{H}^{i}$. In other words, each section can only handle a single monomer unit number to particle ratio, not a range of monomer unit numbers. Therefore, a tie-line approach is always used to generate the correct monomer units to particle ratio given by the nucleation source term, $\text{MU}_{S^{nucl}}^{i}$.

5.5.1 Input File Options

There is a line command in the input file that modifies the stability behavior for the DOF 2 case called `Section Addition Adjust Method`. It can have three valid parameters, `Conservative`, `Ramp`, and `Always Accurate`.

The `Conservative` case always applies the stability operator, Eqn. (5.10), to the nucleation source term. The `Always Accurate` case always skips the application of the stability operator. The third case, `Ramp`, which is the default, attempts to apply the stability operator only when it is needed to maintain positivity of the section bin coefficients by applying the following formula:

$$\boldsymbol{S}^{\text{nucl\_part},\,adj} = \sum_i (\boldsymbol{S}_i^{\text{nucl\_part},\,adj,\,i} + \boldsymbol{S}_{ialt}^{\text{nucl\_part},\,adj,\,i})\Re_i + \boldsymbol{S}^{\text{nucl\_part},\,i}(1-\Re_i) \tag{5.14}$$

where $\Re_i$ is a ramp for section *i* that is 0 when no stability is needed and is 1 when full stability is needed. The following algorithm for determining $\Re_i$ is used.

$$\Re_i = max(\Re_i^{skew}, \Re_i^{sum}) \tag{5.15}$$

where

$$\Re_i^{skew} = Z^2(3-2Z) \text{ where } Z = min\left(1, \frac{ratio - r_L}{r_H - r_L}\right) \text{ and } ratio = max(r_L, \frac{q_{max}}{q_{min}}) \tag{5.16}$$

$$q_{max} = max(q_i^L, q_i^H) \text{ and } q_{max} = min(q_i^L, q_i^H)\,,\ r_L = 5\,, \text{and } r_H = 30$$

and

$$\Re_i^{sum} = Y^2(3-2Y) \text{ where } Y = min\left(1, \frac{y_H - y}{y_H - y_L}\right) \text{ and } y = max\left(y_L, \frac{q_{sum}}{q_i^{cutoff}}\right) \tag{5.17}$$

$$q_{sum} = q_i^L + q_i^H\,,\ y_H = 1 \text{ and } y_L = 0.1$$

$\Re_i^{skew}$ guards against a section becoming skewed, as depicted in Figure 5.6. $\Re_i^{sum}$ makes sure the algorithm reverts to a conservative implementation when there are few particles in the bin. q_i^{cutoff}, which does depend on *i*, is usually set to 4 or 5 magnitudes lower than the total monomer units in the particle phase.

5.6 Jacobian for Particle Nucleation Reactions

The nucleation operator is complicated. However, typically, relatively few gas phase species and relatively few particles section unknowns are involved. Therefore, the way to efficiently calculate it's Jacobian is to limit the number of variables which have to be numerically differenced in its derivation. Let's define the nucleation kinetics operator by the following Eqn.(5.18).

$$\boldsymbol{S}^{\text{nucl}}(b_i^L, b_i^H, Z_{i,m}, C_k^g, P, T) = \begin{bmatrix} \boldsymbol{S}_i^{\text{nucl\_part}} \\ \boldsymbol{S}_k^{\text{nucl\_gas}} \end{bmatrix} \tag{5.18}$$

$\boldsymbol{S}^{\text{nucl}}$ is the full source term for the nucleation operator. $\boldsymbol{S}_i^{\text{nucl\_part}}$ is the source term for the particle unknowns, while $\boldsymbol{S}_k^{\text{nucl\_gas}}$ is the source term for the gas unknowns. In general, particles may both be created and destroyed by this operator. Note, however, operations where the same number of particles appear on both sides of reaction are not included in this operator; they are included in the condensation operator. The dependence on the unknowns is in general full. However, in practice, there is a great deal of sparsity in the Jacobian created by differentiating Eqn. (5.18) with respect to the unknowns.

There are several boolean flags and integer arrays that are used to delineate the sparsity. The values of these booleans are calculated within the program.

`m_reversibleNuclRxn` True if there at least one reversible reaction defined in the nucleation reaction mechanism.

`m_srcAdjMade` True if there at least one source term adjustment made to a nucleation source term, where part of the source term is redirected to another section bin in order to maintain numerical stability.

`m_MUMoleFractionDependent` True if at least one nucleation reaction depends on the mole fraction of MU's.

`m_highestParticipatingSection` Highest section number containing a particle species.

These are used to cut down the operation count when a straightforward numerical delta approach is used to calculate the jacobian terms, with respect to all of the variables that are listed within Eqn. (5.18). For gas-phase unknowns, an additional variable, `heldConstant`, is used to indicate what variables constitute the independent gas-phase unknowns (note mks units are assumed).

`heldconstant`		
	0	`numSpecies` gas phase concentrations and *T*
	1	`numSpecies - 1` gas phase concentrations, *P*, and *T*
	2	`numSpecies - 1` gas phase mass fractions, *P*, and *T*

For the `heldconstant`=0 case, the pressure is not an independent variable. For the other two cases, an additional parameter, *kspec*, the special gas phase species must be supplied such that all delta's of gas phase concentrations follow the following thermodynamically consistent form:

$$C_k = C_k^{base} + \Delta_k \text{ and } C_{kspec} = C_{kspec}^{base} - \Delta_k,$$

such that the mole fractions always sum to one.

6. Surface Growth

The net surface growth rate of a particle, $G(\mu, t)$, is needed in the calculation of the condensation operator (see Chapter 3). $G(\mu, t)$ represents the cumulative effects of all of the reactions between a particle and the surrounding gas, as long as the particle itself is preserved. If the particle is consumed, then the reaction is considered to be covered under the "nucleation" reaction kernel, presented in the previous chapter. Currently, fragmentation of particles due to etching is not covered.

As was mentioned previously the net growth rate $G(\mu, t)$ is assumed to be able to be broken up into the net growth rate per surface area, which we will define as gr, with units of kmol[1] m^{-2} s^{-1} and the surface area corresponding to a particle of size μ, $SA(\mu)$. Then, the net growth rate may be expressed by the following equation

$$G(\mu, t) = (gr)SA(\mu)A_v \tag{6.1}$$

Note, $G(\mu, t)$ is not on a kmol MU basis. $G(\mu, t)$ moves particles along the sectional basis coordinates axis, e.g., μ_0, μ_1, μ_2,..., which is on a straight "number of monomer units" basis. Therefore, $G(\mu, t)$ has units of MU per time. The surface area is currently calculated assuming the particle is a sphere. However in the future, this calculation will be determined via an expanded function involving the fractal dimension of the particle.

$$SA(\mu) = 4\pi\left(\frac{3v(\mu)}{4\pi}\right)^{\frac{2}{3}} \tag{6.2}$$

$v(\mu)$,the volume of a particle with μ total monomer units, is calculated via the following formula.

$$v(\mu) = \frac{\bar{V}_i \mu}{A_v} \tag{6.3}$$

$\bar{V}_i$ is the partial molar volume of particles in section i; A_v is Avogadro's number. The molar volume of the particle is in general a function of the bulk composition of the particle. Currently, a constant molar volume ideal solution is assumed for the particle, where the particle molar volume for section i, $\bar{V}_i$,is calculated from the following formula

$$\bar{V}_i = \sum_{k=1}^{N_{mu}} V_k^o X_{k,i}. \tag{6.4}$$

V_k^o are constant standard state molar volumes for bulk particle monomer unit type *k*, and $X_{k,i}$ is the mole fraction for bulk particle monomer unit type *k* in sectional bin *i*.

The net growth rate gr, Eqn. (6.5), is the some of the individual growth rates for each component of the bulk particle phase, gr_k, of which there are N_{mu}.

$$gr = \sum_{k=1}^{N_{mu}} gr_k . \tag{6.5}$$

gr, as well as the gr_k's are calculated from the surface-interface problem defined below. In general the surface-interface problem involves the two "bulk" phases, the gas phase and the bulk particle phase, on either side of the interface, plus a specification of the interface itself. CADS relies heavily on the preexisting tools for defining and handling two-dimensional interfaces between three-dimensional phases already in Cantera [38]. This capability has been used to solve CVD and solid fuel cell problems. Surface phases are treated as two-dimensional ideal solutions of multiple interfacial species. Each surface phase may have a fixed site density, n^o, and each site may be occupied by one of several adsorbates, or may be empty (an empty site is usually also defined as a surface species). The chemical potential of each species is computed using the expression for an ideal solution:

$$\mu_k = \mu_k^o + RT\log\theta_k . \tag{6.6}$$

where θ_k is the coverage of species *k* on the surface. The coverage is related to the surface concentration, c_k, by the following formula

$$\theta_k = \frac{c_k n_k}{n^o}, \tag{6.7}$$

where n_k is the number of sites covered or blocked by each surface species *k*.

The following also holds for the surface site fraction variables, θ_k.

$$\sum_{k=1}^{N_s} \theta_k = 1 \tag{6.8}$$

In general the surface-growth rate of particles will be a function of the state of the bulk phases on either side of the interface plus the state of the interface itself. Cantera contains an interfacial

kinetics object that defines reactions that occur on the interface, involving both interfacial species and the species defined on either side of the interface in the adjacent bulk phases. The surface interfacial flux, $\dot{\boldsymbol{s}}^{\mathrm{PSF}}$, will have source terms amongst the gas phase, the particle phase, and the surface phase, Eqn. (6.9)

$$\dot{\boldsymbol{s}}^{\mathrm{PSF}} = \begin{bmatrix} \dot{\boldsymbol{s}}^{\mathrm{PSF,surf}} \\ \dot{\boldsymbol{s}}^{\mathrm{PSF,part}} \\ \dot{\boldsymbol{s}}^{\mathrm{PSF,gas}} \end{bmatrix} \tag{6.9}$$

Each of the separate source terms, $\dot{\boldsymbol{s}}^{\mathrm{PSF,surf}}$, $\dot{\boldsymbol{s}}^{\mathrm{PSF,part}}$, and $\dot{\boldsymbol{s}}^{\mathrm{PSF,gas}}$ are vectors over the species in their respective phases. $\dot{\boldsymbol{s}}^{\mathrm{PSF}}$ is in general a function of the gas/particle temperature, T, the gas pressure, and the species concentrations in the gas, monomer unit concentrations in the particle phase, and the surface-site concentrations in the interface phase. In general the surface interfacial flux is specific to each sectional bin, due to the fact that the surface fluxes are in general a function of the bulk-particle phase composition. However, CADS recognizes the special cases where the surface flux problem is not a function of the bulk composition, and therefore, only needs to be calculated once.

One problem remains however. The interfacial species concentrations, or equivalently the surface-site fractions, are not known a priori. The surface-site fractions are not part of the CADS solution vector either. To get around this problem, CADS makes the assumption that their values are in pseudo-steady state equilibrium with the current bulk compositions, and then the pseudo steady-state problem is solved implicitly as part of the source-term calculation. The next section describes the solution of this sub-problem.

6.1 Solving for the Surface Site Fractions

Let c_k be the surface concentration of the k^{th} species in the n^{th} surface phase with units of kmol m^{-2}. Then, Eqn. (6.10) is the conservation equation expressing the continuity balance for the species.

$$\frac{d(Ac_k)}{dt} = A\dot{s}_k^{\mathrm{PSF,surf}}, \qquad k = 1, \ldots, K_s^l \quad , \tag{6.10}$$

$\dot{s}_k^{\mathrm{PSF,surf}}$ is the net production rate from all surface reaction for the k^{th} surface species; it is a subset of the bigger vector, $\dot{\boldsymbol{s}}^{\mathrm{PSF}}$, in Eqn. (6.9). A is the surface area.

One of the surface-site conservation equations, Eqns. (6.10), is redundant and must be replaced by Eqn. (6.8). The equation corresponding to the surface site fraction with the largest magnitude is replaced by Eqn. (6.8).

The equation system, Eqn. (6.8) and Eqn. (6.10), with N_s unknowns, θ_k, is then solved using a damped Newton algorithm, with a time step transient if needed, using the utility routine, named `Placid`, developed under a previous program [39], until $\dot{s}_k^{\mathrm{PSF,surf}} = 0$.

`Placid` has been shown to be remarkably stable. It very rarely fails, and if it does, it's due to a badly constructed surface mechanism. It contains two algorithms. The first conservative algorithm, a pseudo transient that uses a damped newton's method, is used when the initial guess is bad and is based on the ref. [40], which guarantees convergence for a pseudo transient algorithm as long as suitable criteria for the problem statement, such as the existence of a stable steady state, are met. The other algorithm, a simple damped newton's method, is used when the initial guess is very close to the actual solution; this is the norm during an overall residual evaluation, where a saved solution from a previous iteration is used to initialize `placid`. It's relatively fast and accurate, because it's part of a global residual evaluation, that itself may be used as part of evaluating a global Jacobian.

At the end of the procedure, gr_k is identified as $gr_k = \dot{s}_k^{\mathrm{PSF,part}}$.

6.2 Cantera Implementation

Much of the mechanics of calculating production rates and thermodynamic quantities for the surface-growth process is carried out by the normal lower level functions of Cantera. However, in order to use these functions, wrappers around Cantera's thermodynamics objects and kinetics objects that take into account the sectional-bin representation and the need to solve the implicit surface problem are formulated. Figure 6.1 pictorially describes these wrapper objects involved and their interrelationships within the `CADS` calculation of the particle surface growth rate.

The parent object that controls the calculation of $G(\mu, t)$ is the `ParticleSurfRxn` object. A call to the virtual function `surfaceGrowthRate()` returns the vector of values, gr. The calculation of surface area is carried out by calls within the condensation object, `partDiscGalerkin`.

The child of `ParticleSurfRxn` is `PSR_InterfaceKinetics`. This implements `ParticleSurfRxn` members using wrapper objects around Cantera. Note, other children of `ParticleSurfRxn` are also available. For example, the constant surface-growth object, `SurfRxn_Const`, was used in the condensation sample problem in Chapter 3. There is also a HACA surface-growth object, `hacaSurfRxn`, that hard-codes the HACA mechanism for a hard-coded bulk phases [41].

PSR_InterfaceKinetics also inherits from the main Cantera interfacial Kinetics object, InterfaceKinetics, to obtain surface-reaction rates, using the function getNetProductionRates(). InterfaceKinetics contains a vector of the ThermoPhase objects containing all of the phases present at the interface, the interface phase, and the two surrounding bulk phases consisting of the ideal gas phase, IdealReactingGasPhase, and the particle bulk phase, IdealSolidSolnPhase, which was previously discussed in the nucleation chapter. PSK_InterfaceKinetics also contains a contiguous vector for all of the species source terms from the surface reaction.

In order to solve the implicit surface problem, where the surface site fractions are in pseudo-steady state, PSR_InterfaceKinetics calls Placid. Placid implements a nonlinear relaxation algorithm, calling getNetProductionRates() from

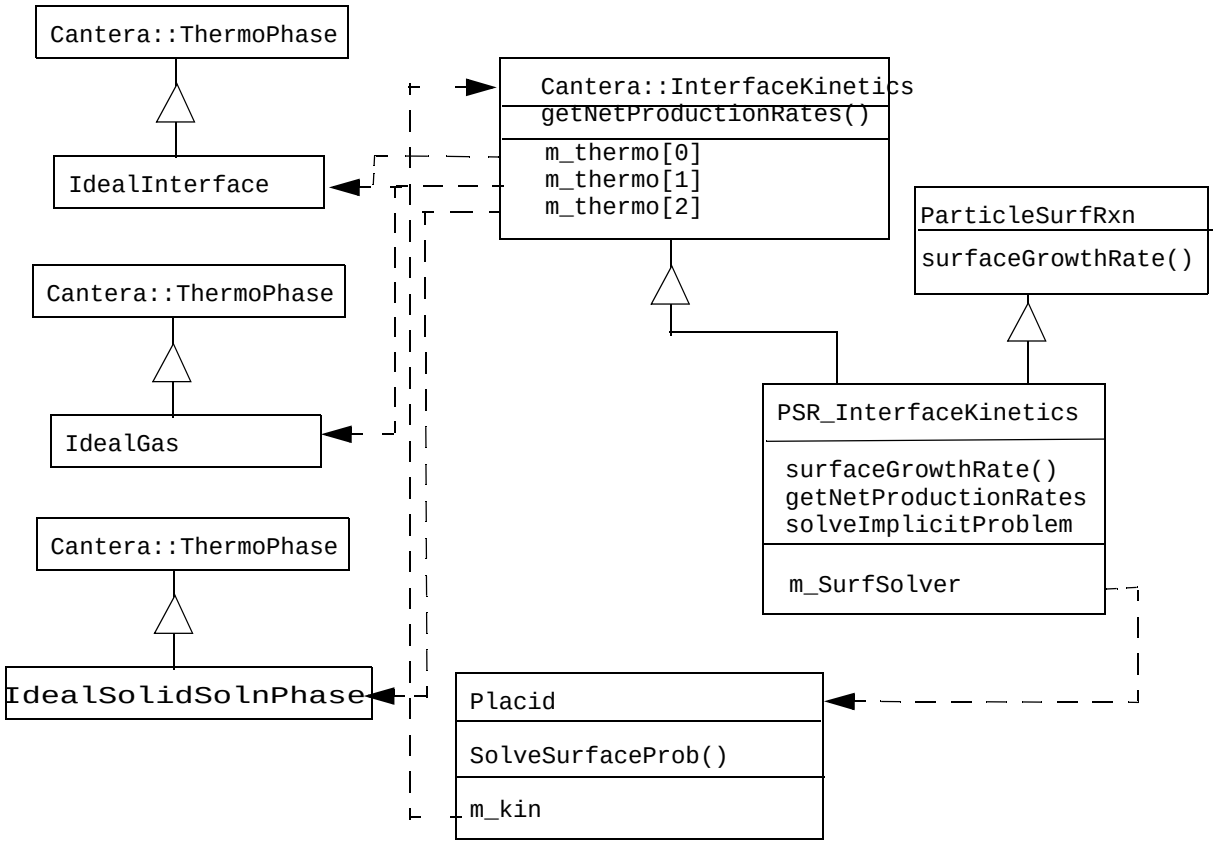

Figure 6.1 Glyph diagram of the implementation of surface kinetics within the Cantera framework. Triangles represent inheritance relationships with the pointy end directed at the parent object. Dotted arrows depict dependencies; one object maintains a pointer to another object in order to carry out a needed function.

```
START BLOCK ADS MODEL DEFINITION
START BLOCK PARTICLE SURFACE REACTION MODEL
   ! Method = [NONE | Cantera | Constant Surface Reaction Rate| haca hardcoded ]
   !     (required) (default = Cantera)
   !   Determines the method that the surface growth rate
   !   is determined.
   Method = Cantera
   !-------------------------------------------------------------
   !   Constant Surface Reaction Rate = { double} [units]
   !        (Default=00)   (conditionally required)
   !   default MKS units = kmol m-2 s-1
   ! Constant Surface Reaction Rate = 5.0E15 gmol/cm2/s
   !-------------------------------------------------------------
   ! Cantera Surface Reaction File = [ string ]
   ! (Default=SurfacePhaseFile.xml)   (optional)
   !   The name of the Cantera input file containing the
   !   description of the particle surface phase and surface
   !   phase reactions.
   Cantera Surface Reaction File = SurfacePhaseFile.xml
END BLOCK PARTICLE SURFACE REACTION MODEL
END BLOCK ADS MODEL DEFINITION
```

Figure 6.2 Block of the input file that controls Surface Growth

`PSR_InterfaceKinetics::InterfaceKinetics` repeatedly to relax the pseudo-steady state system involving the particle surface.

The `IdealSolidSolnPhase` object calculates the concentrations and thermodynamics for particle bulk species, i.e., the full range of features of the parent, Cantera's `ThermoPhase` object. Conceptually, all of the normal thermodynamics functions calculated by the `ThermoPhase` object are handled by the `IdealSolidSolnPhase` object. The `IdealSolidSolnPhase` object is the same object pointed to by the `m_ParticleSolidPhase` pointer in the `PartSpecPhase` object, mentioned in Section 5.3 of the previous chapter.

6.3 Input Deck Options

The preceding sections have described the theory behind the options available to `CADS` users. Figure 6.2 shows the section block of the input file for `CADS` that controls the surface growth calculation.

The first option "`Method`" determines how the surface growth for particles will be determined. The default is Cantera, which means that Cantera input files contain all of the details on how the calculation will be carried out. The method `NONE` means that the surface-growth rate will be set to zero. `Constant Surface Reaction Rate` means that a constant growth rate is applied to monomer unit type 0, and a zero growth rate is applied to all other monomer unit types. The method `HACA hardcoded` implements a hard-coded HACA mechanism for surface growth with hard-coded values.

The next optional keyword, "`Constant Surface Reaction Rate`", sets the constant surface reaction rate to a double; this option is only used when `Constant Surface Reaction Rate` is chosen for the Method.

Cantera Input File specifies the Cantera input file for the description of the surface phase and surface reactions that occur on that surface phase. The file to be input here should be in ctml format, and not Cantera's cti format.

6.4 Sample Problem

A sample problem that implements a HACA-type mechanism within the Cantera Aerosol Dynamics Simulator (i.e., `CADS`), is now presented. I will note up front that the mechanism included in this memo is for example-purposes only and has not been validated against data. The purpose is to generate an explicit example of how to set up a nontrivial surface-condensation mechanism and then to use `CADS` to analyze its behavior.

6.4.1 Description of the Mechanism

The starting point is the paper by Frenklach and Wang [36]. They provide several versions of the HACA (Hydrogen abstraction - carbon addition) mechanism depending upon the size of the aromatic molecule and the amount of detail to be included in the model; the first version is described in their paper on p. 170.

$$A_i + H \leftrightarrow A_i\bullet + H_2 \qquad \text{(P-1)}$$

$$A_i\bullet + C_2H_2 \leftrightarrow A_iC_2H_2\bullet \qquad \text{(P-2)}$$

$$A_iC_2H_2\bullet + C_2H_2 \rightarrow A_{i+1} + H \qquad \text{(P-3)}$$

Equation (P-1) represents the balance between abstraction and recombination of hydrogens on the surface of aromatics, A, of size *i*. Eqn. (P-1) is a reversible reaction, meaning that either forward and reverse rate constants or the species thermodynamics and forward rate constant must be supplied in order to completely specify the behavior of the reaction. Eqn. (P-2) consists of the reversible addition of acetylene to a radical (benzene-like) site on the aromatic. Eqn. (P-3) involves the irreversible addition of another acetylene to the aromatic, which completes the creation of another ring. (P-3) also involves the loss of a hydrogen atom, in order to close the stoichiometric balance on H atoms. Eqns. (P-1) to (P-3) taken together will increase the size of a PAH molecule by one aromatic ring.

There are a few subtle issues about the system. First, the overall rate for Eqn (P-1) is dependent on the number of species of type A_i and $A_i\bullet$ available for collisions with the gas phase molecules. Thus, the rate is proportional to the surface area of the particle. Also, note the following species replace one another on the surface during the mechanism: A_i, $A_i\bullet$, $A_iC_2H_2\bullet$, and A_{i+1}. A_{i+1} can be considered to be the same as A_i, except that one bulk aromatic unit has grown. Therefore, all of these species may be considered to be part of the same surface phase and to be in competition with each other in terms of their surface concentrations. The overall concentration of the union of these surface species may be expressed in terms of a surface site density of the surface phase and has units of kmol sites per m^2 of surface are. The relative concentration of the surface species on the surface may be expressed in terms of surface site fractions, θ_i, where θ_i is defined as the fractional coverage of the surface by surface species *i*.

During the cycle represented by Eqns (P-1) - (P-3) a total of 4 Carbons and 2 Hydrogens are incorporated into the molecule. These represent growth of the solid particle phase and the surface phase of the particle by this amount of elements. There are various choices for how to formulate the bulk phase species. One possibility is to assume this represents 2 bulk carbons PAH species and 2 bulk or surface C-H species that represent benzene-like carbons that have terminating hydrogen atoms. The advantage of this identification is that there are probably group contribution approaches that may be employed to calculate the thermodynamics of these species (e.g., p. 272 of ref 37 has thermodynamic group contribution values for these very species). And, thermodynamics is needed for various stages of the HACA mechanism. It also is better to keep each unit of the bulk phase, i.e., each monomer unit, roughly the same size, because values for the partial molar volume for each monomer unit are needed, and the code currently is more accurate if these are roughly all equal.

Because Eqns. (P-1) to (P-3) represent reactions involving species, it's not necessary to spell out what is in the bulk particle phase and what is in the surface phase of the bulk particle.

Frenklach and Wang go on (p. 172) to provide a more complicated mechanism, which they apply to the growth of larger PAH molecules via the linear kinetics lumping methodology, where they grow 2 units, i.e., grow from A_i to A_{i+2}.The net gain from growing 2 units is +8 C atoms and +2 H atoms. Thus, the unit of growth is seemingly in conflict. But, this is a real issue with underlying consequences for the structure/bonding of the emerging soot particles and needs a proper amount of attention. Note, if one can consider PAH's as a series of interlocked aromatics in a 2D hexagon pattern, then the fundamental unit in the solid particle phase is a single aromatic ring with an overall stoichiometry of 2 carbon atoms (6 carbon atoms each shared equally by 3 surrounding aromatics), with no hydrogen atoms at all. Therefore, the difference between the two PAH-growth mechanisms might involve a difference in the growth rate of the surface phase of the aromatics,

which do have hydrogen atom termination. If the growth process were carried out to grow more units, the number of hydrogen atoms abstracted from the surface might turn out to be increased relatively so that only carbon atoms would be pushed into the solid phase, and the limiting 2 carbon atom stoichiometry of the hexagon unit cell may be reached. The Frenklach and Wang paper doesn't have any kinetic mechanism for incorporating H atoms into the bulk soot phase. Thus, the first Frenklach mechanism may be interpreted as one that grows one "monomer unit of 2 PAH carbons and also grows the surface phase by 2 units of surface C-H. And, the second Frenklach mechanism may be similarly interpreted as one that grows 3 monomer units of 2 PAH carbons and 2 units of surface C-H units.

Indeed, this is the correct interpretation. Frenklach's soot condensation mechanism is described on p. 178 of their paper. This is the method they apply to larger PAH molecules, which they finally call soot. This mechanism is reproduced below:

$$\text{Csoot-H} + \text{H} \leftrightarrow \text{Csoot}\bullet + \text{H}_2 \qquad \text{(S1)}$$

$$\text{Csoot}\bullet + \text{H} \rightarrow \text{Csoot-H} \qquad \text{(S2)}$$

$$\text{Csoot}\bullet + \text{C}_2\text{H}_2 \rightarrow \text{Csoot-H} + \text{H} \qquad \text{(S3)}$$

Csoot-H represents an arm-chair site on the soot particle surface, with Csoot• being the corresponding radical. Rxn (S-1) represents the reversible hydrogen abstraction reaction involving that site. No thermodynamics is given for these sites in the paper. Rxn (S2) represents the recombination of Hydrogen atoms onto a radical site on the soot surface. Additional oxidation reactions are provided for the Csoot• site. However, these will not be discussed further.

In the mechanism (S1) to (S3), a total of 2 carbons and 0 hydrogens are introduced into the bulk particle phase during each cycle. Therefore, no hydrogen species have to be represented in the bulk particle phase; hydrogen species only exist on the surface.

6.4.2 Solution of the Surface Growth Equation Set

Eqns. (S-1) to (S-3) can be expressed in terms of a set of differential equations for the concentrations of the surface species and the formation rate of the bulk particle species:

$$\frac{d[\text{Csoot-H}]_s}{dt} = -R_{s1} + R_{s-1} + R_{s2} + R_{s3} \qquad (6.11)$$

$$\frac{d[\text{Csoot}\bullet]_s}{dt} = R_{s1} - R_{s-1} - R_{s2} - R_{s3} \tag{6.12}$$

$$\frac{d[C_b - 3C_b]_b}{dt} = 2R_{s3} \tag{6.13}$$

$[\ ]_s$ refers to a surface concentration, which has units of kmol m^{-2}. $[\ ]_b$ refers to the concentration of bulk particle species, which has units of kmol m^{-3}. $C_b - 3C_b$ refers to a bulk species consisting of a aromatic carbon bonded to 3 other aromatic carbon; it has a stoichiometry consisting of one carbon atom. In this mechanism, no hydrogen is buried into the bulk soot layer. However, if it were, one might include an additional bulk species named, $C_b - H$. $C_b - H$ consists of a benzene-like carbon atom with an attached Hydrogen atom; it would have a stoichiometry of one Carbon and one Hydrogen atom.

The rates of progress for the reactions are given by the following expressions:

$$R_{s1} = k_{s1}[\text{Csoot-H}]_s[\text{H}] \qquad R_{s-1} = k_{s-1}[\text{Csoot}\bullet]_s[\text{H}_2]$$

$$R_{s2} = k_{s2}[\text{Csoot}\bullet]_s[\text{H}]$$

$$R_{s3} = k_{s3}[\text{Csoot}\bullet]_s[\text{C}_2\text{H}_2]$$

The equation system may be solved by assuming steady state concentrations for the surface species, i.e., setting the Eqns. (6.11) and (6.12) to zero (equivalent to Eqn. (6.10)).

$$[\text{Csoot-H}]_s = \frac{k_{s-1}[\text{Csoot}\bullet]_s[\text{H}_2] + k_{s2}[\text{Csoot}\bullet]_s[\text{H}] + k_{s3}[\text{Csoot}\bullet]_s[\text{C}_2\text{H}_2]}{k_{s1}[\text{H}]} \tag{6.14}$$

$$[\text{Csoot}\bullet]_s = \frac{k_{s1}[\text{H}][\text{Csoot-H}]_s}{k_{s-1}[\text{H}_2] + k_{s2}[\text{H}] + k_{s3}[\text{C}_2\text{H}_2]} \tag{6.15}$$

Eqns. (6.14) and (6.15) are actually linearly dependent with respect to one another. To solve the system we need the additional piece of information that the two surface species sum up to the total surface site concentration, $[\text{Total}]_s$, Eqn. (6.16). The surface site concentration was estimated by Frenklach and Wang to be 2.3×10^{15} cm^{-2} [41].

$$[\text{Csoot-H}]_s + [\text{Csoot}\bullet]_s = [\text{Total}]_s \tag{6.16}$$

Eqn. (6.16) is the dimensionalized equivalent to Eqn. (6.8). Using Eqn. (6.16), the equations for the surface site fractions may be found:

$$[\text{Csoot}\bullet]_s = \frac{k_{s1}[\text{H}][\text{Total}]_s}{k_{s1}[\text{H}] + k_{s\text{-}1}[\text{H}_2] + k_{s2}[\text{H}] + k_{s3}[\text{C}_2\text{H}_2]} \tag{6.17}$$

$$[\text{Csoot-H}]_s = \frac{(k_{s\text{-}1}[\text{H}_2] + k_{s2}[\text{H}] + k_{s3}[\text{C}_2\text{H}_2])[\text{Total}]_s}{k_{s1}[\text{H}] + k_{s\text{-}1}[\text{H}_2] + k_{s2}[\text{H}] + k_{s3}[\text{C}_2\text{H}_2]} \tag{6.18}$$

And, the surface growth rate of particles due to the simplified HACA mechanism is found to be:

$$\frac{d[C_b - 3C_b]_b}{dt} = 2\left(k_{s3}[\text{C}_2\text{H}_2]\frac{k_{s1}[\text{H}][\text{Total}]_s}{k_{s1}[\text{H}] + k_{s\text{-}1}[\text{H}_2] + k_{s2}[\text{H}] + k_{s3}[\text{C}_2\text{H}_2]}\right) \tag{6.19}$$

The surface interface problem has been solved explicitly in this sample problem, because the mathematics is extremely simple. In general, within CADS, we solve the surface interface problem numerically, using placid to obtain Eqn. (6.17), (6.18), and (6.19) numerically. The numerical procedure has the advantage that it generalizes to more complex surface mechanisms automatically.

6.4.3 Implementation within CADS

Figure (6.3) contains the working lines of the cti file that describes the simplified HACA mechanism. The file format has been extensively documented in the manual for the cti data file structure and example files supplied on Goodwin's web site for Cantera [38].

This file may be postprocessed with the Cantera tool, `cti2ctml`, in order to create the file, `haca.xml`. The file `haca.xml` is the actual file for the surface condensation operator that is read into the `CADS` package (see section 6.3). `haca.xml` contains the same information as `haca.cti`, but in an unfriendly format. It also contains elements of the bulk-phase file, that would not normally

```
ideal_gas(name = 'gas', elements = 'O H C N Ar',
          species = 'gri30: H N2 CH3 CH4 C2H2 H2',
          initial_state = state(temperature = 1400.0, pressure = OneAtm,
              mole_fractions = 'H:0.002, N2:0.938, H2:0.04, CH4:0.01, C2H2:0.01'))
stoichiometric_solid(name = 'soot', elements = 'O H C N Ar',
                     density = (3.52, 'g/cm3'),
                     species = 'CB-CB3')
species(name = 'CB-CB3',
        atoms = 'C:1',
        thermo = const_cp(t0 = (1000., 'K'),
                          h0 = (9.22, 'kcal/mol'),
                          s0 = (-3.02, 'cal/mol/K'),
                          cp0 = (5.95, 'cal/mol/K')))
ideal_interface(name = 'soot_interface', elements = 'O H C N Ar ',
               species = 'Csoot-* Csoot-H', reactions = 'all',
                phases = 'gas soot', site_density = (3.8E-9, 'mol/cm2'),
                initial_state = state(temperature= 1000.0,
                                      coverages = 'Csoot-*:0.1, Csoot-H'))
species(name = 'Csoot-*',
        atoms = 'H:0 C:1',
        thermo = const_cp(t0 = (1000., 'K'), h0 = (51.7, 'kcal/mol'),
                          s0 = (19.5, 'cal/mol/K'), cp0 = (8.41, 'cal/mol/K')))
species(name = 'Csoot-H',
        atoms = 'H:1 C:1',
        thermo = const_cp(t0 = (1000., 'K'),
                          h0 = (11.4, 'kcal/mol'),
                          s0 = (21.0, 'cal/mol/K'),
                          cp0 = (8.41, 'cal/mol/K')))
surface_reaction('Csoot-H + H <=> Csoot-* + H2',   [2.5E14, 0.0, 16.0])
surface_reaction('Csoot-* + H => Csoot-H',        [4.0E11, 0.48, -0.072])
surface_reaction('Csoot-* + C2H2 => Csoot-H + H + 2 CB-CB3',
                                                    [4.0E13, 0.0, 10.11])
```

Figure 6.3 File `haca.cti`. This file contains all of the information necessary to specify the surface growth mechanism to `CADS`.

be in the surface-condensation file, specifying the identity, composition, and thermodynamics of the bulk particle phase species, also called “monomer units”. This additional information is used by the test program for `placid`, which is described below.

The equation system is solved within the `CADS` package using a submodel called `placid`, which was originally developed for the `MPSalsa` CVD code package [39]. `placid` solves the surface evolution equation set to the point where the surface species are in pseudo-steady state. In other words, it solves Eqns. (6.11) to (6.12) in this particular case, setting the surface-site time derivatives to zero. In this fashion, the surface-site fractions do not become part of the overall solution vector in the problem; their values are implicitly calculated as part of the overall residual calculation.

6.4.4 Testing the Solution For One Specific Case

Eqns. (6.11) to (6.12) may be solved for one specific case in order to validate the `placid` submodel.

First, let’s calculate the concentration of gas phase species, starting with C_{total}, the total concentration of species in the gas phase, and then applying mole fractions listed in Figure (6.3):

$$C_{total} = 8.705 \times 10^{-6} \frac{\text{gmol}}{\text{cm}^3} \tag{6.20}$$

$$[\text{H}] = 0.002 C_{total} = 1.74 \times 10^{-8} \frac{\text{gmol}}{\text{cm}^3}$$

$$[\text{H}_2] = 0.04 C_{total} = 3.48 \times 10^{-7} \frac{\text{gmol}}{\text{cm}^3}$$

$$[\text{C}_2\text{H}_2] = 0.01 C_{total} = 8.705 \times 10^{-7} \frac{\text{gmol}}{\text{cm}^3}$$

Next we will calculate the forward rate constants for reactions (S1) to (S3) at 1400 K using the reaction rate constants listed in Figure (6.3).

$$k_{s1} = \left(2.5 \times 10^{14} \frac{\text{cm}^3}{\text{gmol s}}\right) \exp\left[\frac{-16 \frac{\text{kcal}}{\text{gmol}}}{RT}\right] = 7.94 \times 10^{11} \frac{\text{cm}^3}{\text{gmol s}} \tag{6.21}$$

$$k_{s2} = \left(4.0 \times 10^{11} \ \frac{\text{cm}^3}{\text{gmol s}}\right) T^{0.48} \exp\left[\frac{0.072 \ \frac{\text{kcal}}{\text{gmol}}}{RT}\right] = 1.329 \times 10^{13} \ \frac{\text{cm}^3}{\text{gmol s}}$$

$$k_{s3} = \left(4.0 \times 10^{13} \ \frac{\text{cm}^3}{\text{gmol s}}\right) \exp\left[\frac{-10.11 \ \frac{\text{kcal}}{\text{gmol}}}{RT}\right] = 1.056 \times 10^{12} \ \frac{\text{cm}^3}{\text{gmol s}}$$

The evaluation of the reverse rate constant for (S1) involves more work, because we first have to evaluate the thermodynamics for all gas and surface species. From table printouts using `cttables`, [42], the standard state gibbs free energies for the gas species are obtained:

$$G°_{\text{H}}(1400K) - \Delta H°_{f,\,298}(\text{H}) = -43.669 \ \frac{\text{kcal}}{\text{gmol}} \text{ and } \Delta H°_{f,\,298}(\text{H}) = 52.103 \ \frac{\text{kcal}}{\text{gmol}} \qquad (6.22)$$

$$G°_{\text{H}}(1400K) = 8.434 \ \frac{\text{kcal}}{\text{gmol}}$$

$$G°_{\text{H}_2}(1400K) - \Delta H°_{f,\,298}(\text{H}_2) = -51.201 \ \frac{\text{kcal}}{\text{gmol}} \text{ and } \qquad \Delta H°_{f,\,298}(\text{H}_2) = 0 \qquad (6.23)$$

$$G°_{\text{H}_2}(1400K) = -51.201 \ \frac{\text{kcal}}{\text{gmol}}$$

While `cttables` is not yet able to handle interfacial reactions with surface species, the standard state thermodynamic functions for the surface phase species may be determined to be:

$$H°_{\text{Csoot}\bullet}(1400K) = 51.7 \times 10^3 + 8.41(400) \ \frac{\text{cal}}{\text{gmol}} = 55.064 \ \frac{\text{kcal}}{\text{gmol}} \qquad (6.24)$$

$$S°_{\text{Csoot}\bullet}(1400K) = 19.5 + 8.41\left(\ln\left(\frac{1400}{1000}\right)\right) = 22.330 \ \frac{\text{cal}}{\text{gmol}}$$

$$G°_{\text{Csoot}\bullet}(1400K) = 55.064 \times 10^3 - (1400)22.330 = 23.802 \ \frac{\text{kcal}}{\text{gmol}}$$

and:

$$H°_{\text{Csoot-H}}(1400K) = 11.4 \times 10^3 + 8.41(400) \ \frac{\text{cal}}{\text{gmol}} = 14.764 \ \frac{\text{kcal}}{\text{gmol}} \qquad (6.25)$$

$$S^{\circ}_{\text{Csoot-H}}(1400K) = 21.0 + 8.41\left(\ln\left(\frac{1400}{1000}\right)\right) = 23.83 \ \frac{\text{cal}}{\text{gmol}}$$

$$G^{\circ}_{\text{Csoot-H}}(1400K) = 14.764 \times 10^{3} - (1400)23.83 = -18.598 \ \frac{\text{kcal}}{\text{gmol}}$$

Therefore, the standard Gibbs free energy of reaction for S-1 is calculated as:

$$\Delta G^{\circ}_{S1} = 23.802 + (-51.201) - ((-18.598) + 8.434) = -17.235 \ \frac{\text{kcal}}{\text{gmol}} \tag{6.26}$$

And, the equilibrium constant for (S1) is calculated as:

$$K_{c,\,\text{S1}} = \exp\left[\frac{\Delta G^{\circ}_{\text{s-1}}}{RT}\right] = 2.0389 \times 10^{-3} \tag{6.27}$$

Therefore, the reverse reaction-rate constant for (S1) is calculated as:

$$k_{\text{S-1}} = k_{\text{S1}} K_{c,\,\text{S1}} = (7.94 \times 10^{11})(2.0389 \times 10^{-3}) = 1.62 \times 10^{9} \ \frac{\text{cm}^3}{\text{gmol s}} \tag{6.28}$$

Values for all of the terms in the denominator of Eqn. (6.17) can be formulated:

$$k_{\text{s1}}[\text{H}] = (7.94 \times 10^{11})(1.74 \times 10^{-8}) = 1.382 \times 10^{4} \ \text{s}^{-1} \tag{6.29}$$

$$k_{\text{s}-1}[\text{H}_2] = (1.618 \times 10^{9})(3.48 \times 10^{-7}) = 5.63 \times 10^{2} \ \text{s}^{-1}$$

$$k_{\text{s2}}[\text{H}] = (1.329 \times 10^{13})(1.74 \times 10^{-8}) = 2.312 \times 10^{5} \ \text{s}^{-1}$$

$$k_{\text{s3}}[\text{C}_2\text{H}_2] = (1.056 \times 10^{12})(8.71 \times 10^{-8}) = 9.198 \times 10^{4} \ \text{s}^{-1}$$

Now, the surface site fractions can be calculated from Eqns. (6.17) and (6.18).

$$\theta^{s}_{\text{Csoot}\bullet} = \frac{[\text{Csoot}\bullet]_s}{[\text{Total}]_s} = 0.0409 \tag{6.30}$$

$$\theta^{s}_{\text{Csoot-H}} = \frac{[\text{Csoot-H}]_s}{[\text{Total}]_s} = 0.9591 \tag{6.31}$$

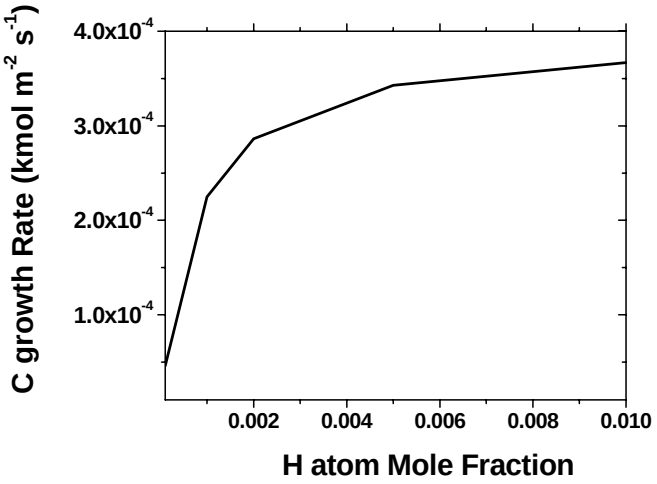

Figure 6.5 Dependence of the particle condensation rate on the mole fraction of H atoms in the gas phase.

$[\text{Total}]_s$ is the total concentration of surface sites. The growth rate may be calculated from Eqn. (6.19).

$$\begin{aligned}
\frac{d[C_b - 3C_b]_b}{dt} &= 2(k_{s3}[\mathrm{C_2H_2}](\theta^s_{\mathrm{Csoot}\bullet}[\text{Total}]_s)) \\
&= 2(1.056 \times 10^{12})(8.71 \times 10^{-8})(0.0409)(3.8 \times 10^{-9}) \\
&= 2.86 \times 10^{-5} \frac{\text{gmol}}{\text{cm}^2\ \text{s}} \\
&= 3.43 \times 10^{-4} \frac{\text{gm}}{\text{cm}^2\ \text{s}}
\end{aligned} \tag{6.32}$$

Figure 6.4 presents the results of the `PSR_PlacidTester2` program, located in the `cads/src` directory. The code itself actually took 23 iterations to solve the simple problem from a bad initial guess, due to the fact that it uses a very conservative pseudo-transient algorithm. The bottom line results are exactly the same as the analytical results presented above. However, all units in the printout are in MKS, the default for `Cantera`, instead of CGS units, which were used in the analysis above. Additional verification using a debugger proved that the thermochemistry involved with surface reaction (S1) agreed with the analytical results presented above.

Figure 6.5 contains the results of the growth rate, expressed as the rate of carbon incorporation into the particle phase per surface area, as a function of the H atom concentration in the gas phase. The

```
Printout of residual and jacobian
       Residual: weighted norm = 6.8907e-14
       Index       Species_Name      Residual     Resid/wtRes      wtRes
       0: soot_interface:Csoot-*  :   0.000e+00    0.000e+00    3.800e-11
       1: soot_interface:Csoot-H  :  -1.084e-19   -9.745e-14    1.113e-06
retn = 1
Gas Temperature = 1400
Gas Pressure    = 101325
Gas Phase:  gas   (0)
                      Name           Conc              MoleF       SrcRate
                                  (kmol/m^3)                    (kmol/m^2/s)
   0                     H      1.74094e-05          0.002   -7.204185e-04
   1                    N2       0.00816501          0.938    0.000000e+00
   2                   CH3                0              0    0.000000e+00
   3                   CH4       8.7047e-05           0.01    0.000000e+00
   4                  C2H2       8.7047e-05           0.01   -1.431975e-04
   5                    H2      0.000348188           0.04    5.034067e-04
Sum of gas mole fractions= 1

Bulk Phase:  soot   (6)
Bulk Temperature = 1400
Bulk Pressure    = 101325
                      Name           Conc              MoleF       SrcRate
                                  (kmol/m^3)                    (kmol/m^2/s)
   0                  CB-CB3          293.065              1    2.863949e-04
Bulk Weight Growth Rate = 0.00343989 kg/m^2/s
Bulk Growth Rate = 9.77241e-07 m/s
Bulk Growth Rate = 3518.07 microns / hour
Density of bulk phase = 3520 kg / m^3
                      = 3.52 gm / cm^3
Sum of bulk mole fractions= 1

Surface Phase:  soot_interface (7)
Surface Temperature = 1400
Surface Pressure    = 101325
                       Name         Coverage         SrcRate
   0                  Csoot-*        0.0409776    -1.084202e-19
   1                  Csoot-H         0.959022    -1.084202e-19
```

Figure 6.4 `PSR_PlacidTester2` output

dependence is linear for low H atom concentrations. However, for higher H atom ratios the linear dependence drops off. This is due to the rate limiting of the overall reaction due to the branching ratio effect created by the ratio of the (S1) and (S2) reaction rate. It's expected that the surface growth rate becomes independent of H atom concentration at high H atom concentrations. Note, all reaction rates and thermochemical input into this program should be double checked for accuracy and relevance. It was not the purpose of this memo to validate reaction rate information against theory or experiment, but merely to introduce and provide an example of the surface solver capability with the `CADS` package.

I have demonstrated that the surface solver portion of the CADS numerical code agrees with an analytical solution for a nontrivial surface mechanism involving implicit degrees of freedom. And, I have described roughly how surface site fractions are handled implicitly within CADS.

The methodology may look complex. However, it simplifies and generalizes the overall treatment of surface chemistry and alleviates the user from having to solve complicated algebraic expressions for effective surface fluxes when the surface chemistry mechanism gets more complicated than surface reaction treatments like (S1) to (S3). For example, a common additional complexity added to soot growth mechanisms is to add branches for the direct addition of PAH's to soot particles. This may be accommodated directly by the simple addition of more surface reactions to the file involving more gas phase species. These surface reactions can be made to depend upon the surface state of the soot surface, i.e., the current state of the surface site densities. Or, it can be made to be independent of the surface-state by not including any dependence on surface site concentrations within the surface mechanism.

7. Particle Bulk-Phase Reactions

CADS can also accommodate reactions in the particle bulk-phase. These consist of homogeneous reactions which take place in the bulk of the particle phase, between bulk phase species. These might include a liquid-to-solid (or crystal-structure) phase change, for example. Alternatively, the reactions might lead to a change in the chemical composition of soot that can occur internally within PAH structures, such as dehydrogenation reactions that transform the PAH further towards a graphitic structure. The latter would result in hydrogen leaving the particle for the gas phase.

The source term for bulk-phase reactions, $\boldsymbol{S}^{bulk}$, has been introduced in Chapter 2, Eqn. (2.20). It may be broken down into sectional components, $\boldsymbol{S}_i^{bulk}$ for section *i*, and then the vector into its components:

$$\boldsymbol{S}_L^{bulk} = \begin{bmatrix} S_i^{\text{bulk,part}} \\ S_i^{\text{bulk,TMU}} \\ S_{i,\,1}^{\text{bulk}} \\ \cdots \\ S_{i,\,nmu-1}^{\text{bulk}} \\ \mathbf{S}_i^{\text{bulk,gas}} \end{bmatrix} \tag{7.1}$$

Several implicit assumptions are employed when formulating the kinetics. The bulk phase is currently considered an ideal solution, and it is "well stirred", i.e., the phase has a homogeneous morphology. No particles are considered to be created or destroyed during this operation, so $S_i^{\text{bulk,part}} = 0$.

The calculation of $\boldsymbol{S}_i^{bulk}$ is carried out in two parts. First, the homogeneous source term $\dot{\boldsymbol{s}}^{\text{bulk}}$ is calculated from the solid kinetics operation. $\dot{\boldsymbol{s}}^{\text{bulk}}$, which spans all of the particle monomer unit types in the phase, has units of kmol m^{-3} s^{-1}. This is on a per volume of particle basis, as the kinetics object is treating the particle medium as a homogeneous volumetric phase. Currently, the source term for gas species from bulk particle reactions, $\mathbf{S}_i^{\text{bulk,gas}}$, is equal to zero; this may be changed in the future.

The source term, $\dot{\boldsymbol{s}}^{\text{bulk}}$, for species due to bulk particle reactions is calculated in the following standard Cantera methodology. Define N_r^{bulk} as the number of bulk phase reactions. The source term for kinetic species *k* (which spans all species in all phases present the gas, bulk, and surface

of the particles) is given by:

$$\dot{s}_k^{\text{bulk}} = \sum_{j=1}^{N_r^{bulk}} \alpha_{k,j} \dot{R}_j \tag{7.2}$$

The rate of progress, $\dot{R}_j$,of the j^{th} reaction is evaluated according to mass action kinetics, Eqn. (7.3).

$$\dot{R}_j = k_j^f \left(\prod_{k=1}^{N_{mu}} (c_k^g)^{\alpha_{k,j}^f} \right) - k_j^r \left(\prod_{k=1}^{N_{mu}} (c_k^g)^{\alpha_{k,j}^r} \right) \tag{7.3}$$

c_k^g are generalized concentrations of the bulk species. Generalized concentrations, Eqn. (7.4), are calculated from the activity of the bulk-phase species, a_k, multiplied by the standard concentration of the bulk species, $c_{o,k}$, which is an artifice that translates between activities and actual reaction-rate system used:

$$c_k^g = a_k c_{o,k} \tag{7.4}$$

In the discussion below, $c_{o,k}$ is further assumed to be independent of *k*. $\alpha_{k,j}^f$ and $\alpha_{k,j}^r$ are the forward and reverse species reaction order parameters for the j^{th} reaction. The stoichiometric coefficient, $\alpha_{k,j}$, for the k^{th} species in the j^{th} reaction is given by Eqn. (7.5).

$$\alpha_{k,j} = \alpha_{k,j}^r - \alpha_{k,j}^f \tag{7.5}$$

k_j^f and k_j^r are the temperature-dependent forward and reverse reaction rate constants for the j^{th} reaction. The reverse rate constant may be calculated from the forward rate constant from the following formula:

$$k_j^r = k_j^f (c_o)^{\Delta\alpha} \exp\left(\frac{-\Delta G_{o,j}}{RT}\right),$$

where

$$\Delta\alpha = \sum_{k=1}^{N_{mn}} \alpha_{k,j}^f - \alpha_{k,j}^r. \tag{7.6}$$

The standard-state Gibbs free energy, $\Delta G_{o,j}$, for reaction *j* is given by

$$\Delta G_{o,j} = \sum_{k=1}^{N_{mn}} \alpha_{k,j} \mu_k^o(T,P), \tag{7.7}$$

where $\mu_k^o(T,P)$ is the standard chemical potential of the k^{th} bulk phase species.

Then, $\dot{\boldsymbol{s}}^{\text{bulk}}$ is translated into $\boldsymbol{S}_i^{bulk}$ by multiplication of the volume of particles in the section per volume of gas, Eqn. (7.8).

$$S_{i,k}^{bulk} = \dot{s}_k^{\text{bulk}} (\bar{V}_i) Q_i \tag{7.8}$$

$\bar{V}_i$ is the molar volume for particles in section i (m^3 $kmol^{-1}$); Q_i is the total monomer density for section i (kmol m^{-3}). The molar volume of the particle is in general a function of the bulk composition of the particle. Currently, a constant molar volume ideal solution is assumed for the particle, where the particle molar volume for section i, $\bar{V}_i$,is calculated from the following formula:

$$\bar{V}_i = \sum_{k=1}^{N_{mu}} V_k^o X_{k,i}. \tag{7.9}$$

V_k^o are constant standard state molar volumes for bulk particle monomer unit type k, and $X_{k,i}$ is the mole fraction for bulk particle monomer unit type k in sectional bin i. Note, the formulation for $\bar{V}_i$ is used extensively within CADS to calculate particle volumes and particle surface areas assuming spherical particles.

Within CADS, the conservation equation for the first monomer unit type is discarded, and a conservation equation for the total number of monomer units is used in its place. The source term for the total monomer units is given by Eqn. (7.10).

$$S_i^{\text{TMU,bulk}} = \left(\sum_{k=1}^{N_{mu}} \dot{s}_k^{\text{bulk}} \right) (\bar{V}_i) Q_i \tag{7.10}$$

The source term for particle creation from bulk kinetics is currently zero:

$$S_i^{\text{part,bulk}} = 0 . \tag{7.11}$$

Nothing is currently done in the DOF-2 implementation to prevent skew numerical instabilities due to the non-zero values for $S_i^{\text{TMU,bulk}}$. For the DOF-1 implementation a tie-line calculation is

carried out to maintain particle conservation, so that the source term for section *i* is partly distributed into neighboring sections whenever $S_i^{\text{TMU,bulk}} \neq 0$ (see section 4.5.2).

Extensions towards inclusion of a more complicated bulk-phase reaction term may be made in the future. It's not inconceivable to add bulk gas-phase species as reactants or products to `CADS` solid kinetics treatment. However, this isn't done now, because extending the number of entries in the species vector and the bulk kinetics source term vector associated with this operation would be inefficient, and a use case for this capability has not arisen. Additionally, multiple bulk phases, at least based on an equilibrium partition of the available type of bulk monomer units in each sectional bin, may be easily achieved within this framework. Recently the Pitzer model for strong electrolyte thermodynamics has been implemented within Cantera [43], providing the necessary fundamental support for aerosol modeling involving the effluorescence and deliquescence of atmospheric salt particles.

7.1 Cantera Implementation

Much of the mechanics of calculating production rates and thermodynamic quantities for the bulk kinetics is carried out by the normal lower level functions of Cantera. However, in order to use these functions, wrappers around Cantera's thermodynamics objects and kinetics objects that take into account the sectional bin representation are formulated. Figure 7.1 pictorially describes these wrapper objects involved and their interrelationships within the `CADS` calculation of the particle bulk reactions.

The parent object that controls the calculation of $\boldsymbol{S}^{bulk}$ is the `PartSolidKin` object; this object contains the man API for bulk kinetics with respect to the other parts of `CADS`. A call to the virtual function `solidKinRate()` returns the bulk particle source term vector, $\boldsymbol{S}_i^{bulk}$, discussed in the previous section.

The child of `PartSolidKin` is `PSK_SolidKinetics`. This implements `PartSolidKin` members using wrapper objects around Cantera.

`PSK_SolidKinetics` also inherits from a child object of the main Cantera Kinetics object, `Kinetics`, called `SolidKinetics`. `SolidKinetics` is a variant of the `Kinetics` object, with support to handle solid-phase reactions efficiently. `SolidKinetics` currently assumes that there is one and only one bulk phase present, and that phase is the Cantera `ThermoPhase` model, `IdealSolidSolnPhase`. `IdealSolidSolnPhase` implements an ideal solution model with constant partial molar volumes and with a few choices for the formulation of generalized concentrations. Generalized concentrations, which translate the activity of a species into the

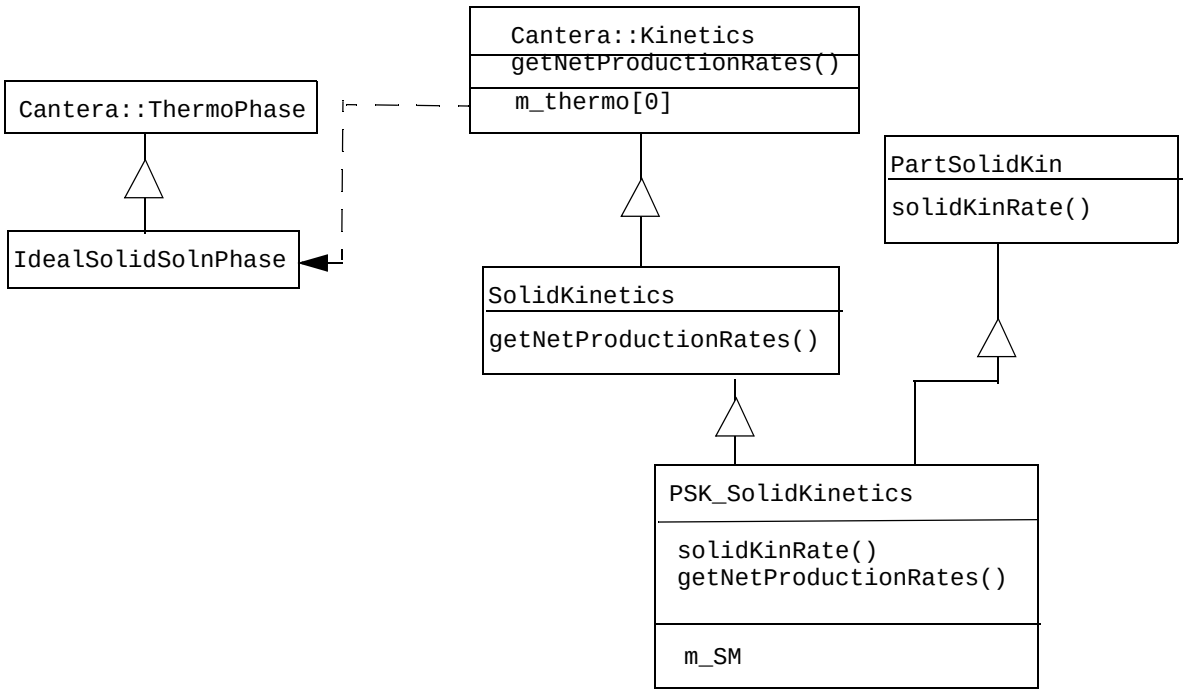

Figure 7.1 Glyph diagram of the implementation of bulk particle reactions within the Cantera framework. Triangles represent inheritance relationships with the pointy end directed at the parent object. Dotted arrows depict dependencies.

quantity used in kinetics rate expressions, may either be unity or the inverse of the species molar volume. So far, unity has been used within `CADS`. This means that reaction expressions involving bulk-phase species involve the activity, and therefore, in the case of ideal solutions, the mole fractions of bulk species.

For each section, `PSK_SolidKinetics` calculates the net production rates of bulk particle monomer units on a "per volume of particle" basis, due to the bulk homogeneous context of Cantera's kinetics operation. Then, it multiplies that with the particle volume of section *i* per volume of gas, which it obtains via queries to the main `SectionModel` object with `CADS`, `m_SM`, to get the correct values for $\boldsymbol{S}_i^{bulk}$.

7.2 Input Deck Options

The preceding sections have described the theory behind the options available to `CADS` users. Figure 7.2 shows the section block of the input file for `CADS` that controls the bulk-phase kinetics calculation.

```
START BLOCK ADS MODEL DEFINITION
START BLOCK PARTICLE PHASE MODEL
   ! Method = [NONE | ADS Ideal Solution | Cantera ]
   !     (required) (default = Cantera)
   !   Determines the method that is used internally for calculating the thermodynamics
   !   of the particle phase.
   Thermodynamics Method = Cantera
   !-----------------------------------------------------------
   ! Cantera Particle Condensed Phase File = [ string ]
   ! (Default=ParticlePhase.xml)   (conditionally required)
   !   The name of the Cantera input file containing the
   !   description of the particle condensed phase.
   !
   Cantera Particle Condensed Phase File= ParticlePhase.xml
END BLOCK PARTICLE SURFACE REACTION MODEL

START BLOCK PARTICLE PHASE KINETICS
   ! Kinetics Method = [ NONE | Cantera ]
   !    (required) (default = Cantera)
   !   Method for specifying the bulk phase kinetics. Note,
   !   if Cantera is chosen here, Cantera must be chosen for
   !   the Thermodynamics Method.
   !
   Kinetics Method = Cantera
END BLOCK PARTICLE PHASE KINETICS
END BLOCK ADS MODEL DEFINITION
```

Figure 7.2 Block of the input file that controls the specification of the bulk-phase thermodynamics and bulk-phase kinetics mechanism.

There are two blocks. The first block `Particle Phase Model` controls how the bulk phase thermodynamics is calculated within the program. The Method keyline controls the basic functionality. There are three options. `NONE` means that there is no thermodynamics or thermodynamics input file for the bulk particle phase. Very limited functionality within `CADS` is then available. The number, names, stoichiometry, and molar volumes of the species in the particle phase are then input via the `SECTION MODEL DEFINITION` block described in Section 2.11 and Figure 2.1. The next option `ADS Ideal Solution` is currently unimplemented, and shouldn't be used. The last option, `Cantera`, is used and should be used to obtain the full `CADS` functionality.

`Cantera Particle Condensed Phase File` specifies the Cantera input file for the description of the bulk particle phase and bulk particle reactions that occur homogeneous within that phase. The file to be input here should be in ctml format, and not Cantera's cti format, and is described in the following section.

The next block, `PARTICLE PHASE KINETICS,` has options that apply specifically to the bulk phase kinetics operator. There is one keyline in the block, called `Kinetics Method`, with two possible values: `NONE`, and `Cantera`. This option serves as a toggle switch for turning on and off the bulk particle kinetics source term.

7.3 Sample File

Figure 7.3 contains a sample bulk-phase thermodynamics file, with bulk phase chemistry defined. The XML element, `speciesArray`, defines the number and names of the monomer units defined in the bulk particle phase. Two species are defined: `C-new` and `C-graph`. The number and names of the species must match input that has been specified in other areas of the `CADS` input; if not, an error is thrown.

Each species has a section of the file devoted to specification of its standard state properties. This is in the `speciesData` XML element, in sub XML elements with names `species` and the `name` attribute equal to the name of the species.

In `species` sub element, the element stoichiometry of the species is given. NASA polynomials giving the polynomial forms for the reference state thermodynamic properties at 1 bar are provided, and the standard state molar volumes are also provided.

The thermodynamics model, `IdealSolidSolution`, to be used is listed in the `thermo` XML element. Parameters for the model, `standardConc model = "unity"`, are also provided in the `thermo` block. Setting the standard concentration to unity means that $c_{o,k} = 1$ in Eqn. (7.4); this is a standard treatment for solid phase kinetics.

A single bulk phase reaction is defined at the bottom of the file.

$$\texttt{C-new} \rightarrow \texttt{C-graph} \tag{B-1}$$

The reaction is defined as irreversible, and has the following formula for its rate of progress, Eqn. (7.3).

$$R_1 = (10000) X_{\texttt{C-new}} \tag{7.12}$$

Note, the Enthalpy and Gibbs free energy of the solid phase is defined in the bulk XML file, through the specification of the standard-state NASA polynomials and bulk-mixture thermodynamics properties. In particular, the sensible heat of formation of the particle phase is given by the difference in absolute enthalpies between the gas-phase precursor species and the absolute enthalpies of the particle phase, specified via the file presented in Figure 7.3. Overall consistency in the thermodynamics treatment within `CADS` means that attention must be spent on specifying the correct numbers for the bulk-phase thermodynamics. The importance of this issue has been addressed in [44]. The NASA polynomial form used for both the `C-new` and `C-graph` species in

```
<ctml>
   <phase id="sootPhase" dim="3">
    <state>
      <temperature units="K">500</temperature>
    </state>
   <thermo model="IdealSolidSolution">
       <standardConc model="unity" />
    </thermo>
    <elementArray datasrc="elements.xml"> H  C  O  N  Ar </elementArray>
    <speciesArray datasrc="#species_solidSolution">
        C-new  C-graph
    </speciesArray>
    <reactionArray datasrc="#reactions_solidSolution">
    </reactionArray>
    <kinetics model="SolidKinetics" />
    <standardConc model="unity" />
  </phase>
  <speciesData id="species_solidSolution">
    <species name="C-graph">
      <note>This corresponds to old soot</note>
      <atomArray> C:1 </atomArray>
      <thermo>
       <NASA P0="100000.0" Tmax="1000.0" Tmin="200.0">
         <floatArray size="7" title="low">
           2.344331120E+000,  7.980520750E-003, -1.947815100E-005,
           2.015720940E-008, -7.376117610E-012, -9.179351730E+002,
           6.830102380E-001
         </floatArray>
        </NASA>
        <NASA P0="100000.0" Tmax="3500.0" Tmin="1000.0">
          <floatArray size="7" title="high">
           3.337279200E+000, -4.940247310E-005,  4.994567780E-007,
          -1.795663940E-010,  2.002553760E-014, -9.501589220E+002,
          -3.205023310E+000
          </floatArray>
        </NASA>
      </thermo>
      <standardState model="constant_incompressible">
         <molarVolume> 0.0255 </molarVolume>
      </standardState>
    </species>
    <species name="C-new">
      <note>This corresponds to new soot</note>
      <atomArray> C:1 </atomArray>
      <thermo>
       <NASA P0="100000.0" Tmax="1000.0" Tmin="200.0">
         <floatArray size="7" title="low">
           2.344331120E+000,  7.980520750E-003, -1.947815100E-005,
           2.015720940E-008, -7.376117610E-012, -9.179351730E+002,
           6.830102380E-001
         </floatArray>
        </NASA>
        <NASA P0="100000.0" Tmax="3500.0" Tmin="1000.0">
          <floatArray size="7" title="high">
           3.337279200E+000, -4.940247310E-005,  4.994567780E-007,
          -1.795663940E-010,  2.002553760E-014, -9.501589220E+002,
          -3.205023310E+000
          </floatArray>
        </NASA>
      </thermo>
      <standardState model="constant_incompressible">
         <molarVolume> 0.0255 </molarVolume>
      </standardState>
    </species>
  </speciesData>
  <!-- reaction data -->
  <reactionData id="reactions_solidSolution" model="SolidKinetics" submodel="SolidKinetics_0">
   <reaction id="bulk_rxn_1" reversible="no">
      <equation>C-new [=>] C-graph</equation>
      <reactants> C-new:1 </reactants>
      <products> C-graph:1 </products>
      <rateCoeff>
        <Arrhenius order="1">
          <A> 10000. </A> <b> 0.0    </b> <E units="cal/mol"> 00000. </E>
        </Arrhenius>
      </rateCoeff>
    </reaction>
  </reactionData>
</ctml>
```

Figure 7.3 Sample bulk particle phase thermodynamics file - xml format

Figure 7.3 are derived assuming bulk graphite, with the absolute scale consistent with the NIST database scale used in the online databases [45].

8. Transport Properties of Particles

Transport properties for particles are needed for conservation equations for particles in flows. Transport properties are also needed for zero-dimensional simulations of particle concentrations, because particle source and sink terms often involve transport properties in their constitutive models.

In all cases, particles are assumed to be in inertial equilibrium with respect to the gas. This, of course, places a large restriction on the problems that CADS may handle, and the restriction may be relaxed in the future. Equations for justification of this restriction, in the initial CADS implementation aimed at sooting flames, are presented in Section 8.2. The transport properties of particles are then broken up into two cases: the calculation of the terminal velocity of particles, which though assumed to be in inertial equilibrium, may not be equal to the mass-averaged gas velocity, and the calculation of the diffusivities of particles. Particle terminal velocities from the thermophoretic and diffusiophoretic forces are described. Currently, the only diffusion kernel implemented within CADS is Brownian diffusion. A turbulent diffusivity kernel may be implemented at a later date.

8.1 Brownian Particle Diffusion Coefficients

Much of the theory for Brownian diffusion of particles has already been developed in Chapter 4, when introducing the Coagulation operator. The results will briefly repeated here. The Brownian diffusivity of particle *i* may be given by the following relation, Eqn. (8.1) [32].

$$D_i^b = \frac{kT}{f}, \text{ where } f = \frac{3\pi\mu d_i}{C_i} \text{ is the friction factor} \tag{8.1}$$

C_i is the Cunningham correction factor to Stokes law, used to account for slip-flow on the particle surfaces, and is dependent on the particle Knudsen number. C_i is given by the following formula [2, p. 34], obtained originally by a fit to experimental data:

$$C_i = 1 + (Kn)\left(A_1 + A_2 \exp\left[-\frac{A_3}{Kn}\right]\right). \tag{8.2}$$

$$A_1 = 1.207,\ A_2 = 0.400, \text{ and } A_3 = 1.10$$

where

$$Kn = 2\frac{\lambda}{d_i} \tag{8.3}$$

$$\lambda = \frac{1}{\sqrt{2}\pi n\sigma^2} \cong \frac{2\mu}{\rho\bar{u}} = \frac{2\mu}{\rho}\sqrt{\frac{\pi\overline{M}}{8RT}} \tag{8.4}$$

The equation for C_i provides an interpolation formula for the entire range of Knudsen number. λ is the mean free path in the gas. In Eqn. (8.4) n is the gas concentration and σ is the collisional cross section of gas phase molecules. Note the hard sphere result from BSL [51], for example, has a 3 instead of the 2 next to μ in Eqn. (8.4). However, the result for a real gas is much more complicated. The actual answer is that the number has to be consistent with the accommodation coefficient, α, (a quantity appearing in the molecular limit which represents the fraction of molecules which leaves the particle surface in equilibrium) such that the drag force on the particle in the free molecular and continuum limits is given by the correct, consistent, limiting expression. Using Eqn. (8.4), and the expression for the friction coefficient in the molecular limit, [2, p. 33],

$$f = \frac{2}{3}d^2\rho\left(\frac{2\pi RT}{\overline{M}}\right)^{\frac{1}{2}}\left[1 + \frac{\pi\alpha}{8}\right], \tag{8.5}$$

Eqns. (8.1), (8.2), (8.4), and (8.5) may be solved for in the molecular limit ($Kn \to \infty$) to determine the consistent value of α It turns out to be $\alpha = 0.907$. However, no adequate theories exist for specification of α. Therefore, α is assumed to be a given constant within the CADS package. Figure (8.1) displays the particle diffusivity vs. particle diameter result for the 2 cases of the full expression, Eqn. (8.1), vs. just the molecular kinetic theory limit, Eqn. (8.5). The kinetic theory expression is calculated assuming a 0.91 accommodation coefficient. The bath gas is assumed to be 400 K N_2 at 1 atm. The full expression, Eqn. (8.1), is seen to be consistent with the molecular limiting theory at small particle sizes. If the curve is extrapolated to “particles” of molecular dimensions, then predicted diffusivities are roughly equal to their gas-phase molecular counterparts (0.5×10^{-4} m^2 s^{-1}). At large particle sizes, Eqn. (8.1) yields a much larger value of the diffusion coefficient than the molecular limiting expression.

8.2 Particle Terminal Velocities

The premise used within CADS is that an aerosol particle moves with its local terminal velocity, calculated from the local gas conditions, i.e., it is not subject to inertial forces. This statement is expressed mathematically by Eqn. (8.6).

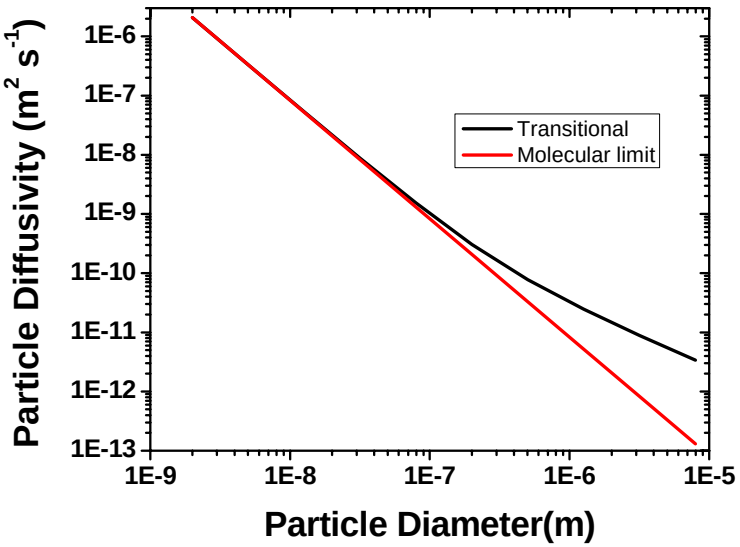

Figure 8.1 Particle Brownian diffusion coefficients as a function of particle diameter. 400 K, 1 atm, in an N_2 ambient. Black line is the full formula; red line is the formula from the molecular limiting case, assuming an accommodation coefficient of 0.91.

$$m_{part}\frac{d\boldsymbol{v}_{part}}{dt} = \boldsymbol{F}_{diffusiophoresis} + \boldsymbol{F}_{thermophoresis} + \boldsymbol{F}_{other} = \boldsymbol{0} \tag{8.6}$$

$\boldsymbol{F}_{diffusiophoresis}$ and $\boldsymbol{F}_{thermophoresis}$ are the force on a particle due to diffusiophoresis, i.e., gas concentration gradients, and thermophoresis, i.e., gas thermal gradients, respectively. $\boldsymbol{F}_{other}$ are other forces, such as the gravity force or the force on a charged particle in a electric field, that are not included in CADS, but may be added later. When there is only one component in the gas, $\boldsymbol{F}_{diffusiophoresis}$ reduces to $-f(\boldsymbol{v}_{part} - \boldsymbol{v})$, where $\boldsymbol{v}$ is the velocity of the gas, and f is the drag coefficient of the particle, previously introduced in Chapter 4, Eqn. (4.4).

The non-dimensional time constant for satisfaction of the right equality in Eqn. (8.6) can be derived from a nondimensionalization of the particle transport equation, Eqn. (8.6). The dimensionless time constant, τ, Eqn. (8.7), which is derived assuming the Stokes drag flow expression, $f = 3\pi\mu d_p$.

$$\tau = \frac{\rho_{part} d_p^2}{18\mu} \tag{8.7}$$

The Stokes number, which is equal to the dimensionless ratio of the stopping distance to a charac-

teristic length scale, L, of the problem may then be defined as

$$\mathrm{Stk} = \frac{\rho_{part} d_p^2 u_o}{18 \mu L} \tag{8.8}$$

where u_o is a characteristic velocity of the gas.

In CADS, the assumption is made that Stk is small, and that therefore inertial effects don't have to be tracked. This turns out to be valid for applications where CADS has been applied to. For example, typical numbers for opposing flow flames are $u_o = 100$ cm s^{-1}, and $L = 1$ mm. For air, it turns out that $\mathrm{Stk} = 1$ requires particles of ~25 μm, which are larger than normal soot particles. Support for inertial effects may be added later to CADS; as there are important applications that may only be approached via the inclusion of these effects [12].

In order to calculate the terminal velocity, the forces on a particle are summed up and the terminal velocity is determined from the condition that the resulting sum of the forces must equal zero, Eqn. (8.6). The terminal velocity of the particle, $\boldsymbol{v}^t_{\mathrm{part}}(\mu)$, will be a function of the particle total monomer unit size. In general, $\boldsymbol{v}^t_{\mathrm{part}}(\mu)$ is closely related to the mass-averaged velocity of the gas, $\mathbf{v}$, and may be expressed in terms of $\mathbf{v}$ and a particle-size dependent perturbation, $\Delta\boldsymbol{v}^t_{\mathrm{part}}(\mu)$, from $\mathbf{v}$.

$$\boldsymbol{v}^t_{\mathrm{part}}(\mu) = \mathbf{v} + \Delta\boldsymbol{v}^t_{\mathrm{part}}(\mu) \tag{8.9}$$

8.3 Diffusiophoresis

Let's for the moment just consider the force on a particle due to molecular diffusion, i.e., the diffusiophoretic force. The expression for diffusiophoresis comes from Waldmann [17] as reported in Waldmann and Schmitt [46]. The force exerted on a small particle, defined as one which is small enough such that the gas flow is not perturbed, by the gas is given by the expression in Eqn. 8.10 [p. 140, ref 46].

$$\boldsymbol{F}_{diff}(\mathrm{Kn} \to \infty) = \left(-\frac{32}{3} r_{part}^{\,2}\right) \sum_k \left(\left(1 + \frac{\pi}{8}\alpha_k\right) p_k \frac{(\boldsymbol{v}_{part} - \boldsymbol{v}_k)}{c_k}\right) \tag{8.10}$$

In the above expression, $\boldsymbol{v}_{part}$ is the velocity of the particle. α_k is the accommodation coefficient for collisions of species k with the particle, α_k, the accommodation coefficient, is the fraction of species i which reflects thermally with a Maxwellian distribution from the particle as contrasted with specular (i.e., elastic) collisions. p_k is the partial pressure of species k, and r_{part} is the radius of the particle. c_k is the average thermal velocity of species k,

$$c_k = \sqrt{\frac{8kT}{\pi m_k}}. \tag{8.11}$$

$\boldsymbol{v}_k$ is average velocity of species *k*, which includes the diffusive flux for species *k*. For example, if $\boldsymbol{v}$ is the mass averaged velocity of the gas, and $\boldsymbol{j}_k$ is the diffusive mass flux of species *k*, then $\boldsymbol{v}_k$ could be calculated from the following expression:

$$\rho_k \boldsymbol{v}_k = \rho_k \boldsymbol{v} + \boldsymbol{j}_k, \tag{8.12}$$

where ρ_k is the density of species *k*. If there is no diffusion, then all species move with the mass average velocity, and a particle also moving with the mass averaged velocity subject only to $\boldsymbol{F}_{diff}$ force will experience zero force if it too moves with the mass averaged velocity. If there is species diffusion, then in general the value of $\boldsymbol{v}_{part}$ which creates a net zero force on the particle will not be equal to the mass average velocity of the gas. It will depend in a complicated way on the species gradients. The general principle is that aerosol particles move in the direction of the diffusion flux of the heavier gas components.

Eqn. (8.10) may be put in terms of the frictional drag coefficient, *f*, used in Chapter 4. *f*, in the limit of infinite Kn, see Eqn. (4.8), may be rearranged to the following value

$$f(\mathrm{Kn} \to \infty) = f_\infty = \frac{32}{3} r_{part}{}^2 \left(1 + \frac{\pi}{8}\alpha\right) \frac{p}{\bar{c}}, \tag{8.13}$$

where $\bar{c}$ is the average velocity of the gas, based on the average molecular weight, and α is the average accommodation coefficient. With this result in hand, it seems prudent and necessary (after introducing thermophoresis force) to extend the diffusiophoresis formula from infinite Knudsen numbers to all Knudsen numbers via the following formulation:

$$\boldsymbol{F}_{diff}(\mathrm{Kn}) = \boldsymbol{F}_{diff}(\mathrm{Kn} \to \infty)\left(\frac{f}{f(\mathrm{Kn} \to \infty)}\right) \tag{8.14}$$

Here, f is given by the Knudsen-Weber interpolating formula, Eqn. (4.4), for the friction of a particle of radius *r*. In Chapter 4, it was found that for a particular value of α, 0.91, Eqn. (4.4) may yield the same value f_∞ in the $\mathrm{Kn} \to \infty$ limit.

Note, if the diffusiophoresis force is the only force acting on the particles, then the terminal velocity of particles will not vary due to the correction in Eqn. (8.14). And, therefore, the predicted diffusiophoresis terminal velocity will be independent of the particle radius. While there is experimental data in binary gases [46] which demonstrates that the diffusiophoresis terminal ve-

locity does depend on Kn, the existing theories on large-particle diffusiophoresis employ empirically derived ratios of accommodation coefficients [46] to calculate an empirical “diffusion slip factor”, and also have been applied only to binary gases. Therefore, the decision was made not to incorporate them within CADS.

8.4 Thermophoresis

For small particles the force on a spherical particle in which there is a temperature gradient is given by Eqn. (8.15) [46].

$$\boldsymbol{F}_{thermophoresis} = -\frac{32}{3} r_{part}^{2} \frac{1}{5} \lambda_{tran} \frac{\nabla T}{\bar{c}} \tag{8.15}$$

λ_{tran} is the translational heat conductivity (or monatomic heat conductivity) expression

$$\lambda_{tran} = \frac{15R\mu}{4\overline{M}}, \tag{8.16}$$

with viscosity μ, and molecular weight $\overline{M}$. For a single gas, the terminal velocity in the small particle limit may be calculated from Eqn. (8.10) and (8.15) to yield Eqn. (8.17) [2].

$$\boldsymbol{v}_{part} = \frac{-3\mu\nabla T}{4\rho\left(1 + \frac{\pi\alpha}{8}\right)T} \tag{8.17}$$

The force on larger particles, those near the continuum limit has been a subject of much conjecture. CADS has employed the formulation due to Brock [48], derived from using Maxwell’s classical boundary conditions involving thermal and frictional slip, Eqn. (8.18).

$$\boldsymbol{F}^{brock}_{thermophoresis} = -6\pi\mu R\left(\frac{2C_s(\lambda + C_t(\mathrm{Kn})\lambda_{part}))}{(1 + 3C_m(\mathrm{Kn}))(2\lambda + \lambda_{part} + 2C_t(\mathrm{Kn})\lambda_{part}))}\right)\frac{1}{5}\frac{\lambda}{p}\nabla T \tag{8.18}$$

In the equation, λ is now the full thermal conductivity of the gas. λ_{part} is the thermal conductivity of the particle phase. The ratio λ_{part}/λ is an important parameter for the large particle ratio limit.T The numerical constants C_t and C_m are given by

$$C_t = \frac{15}{8}\left(\frac{2-\alpha_t}{\alpha_t}\right) \text{ and } C_m = \frac{2-\alpha_m}{\alpha_m} \tag{8.19}$$

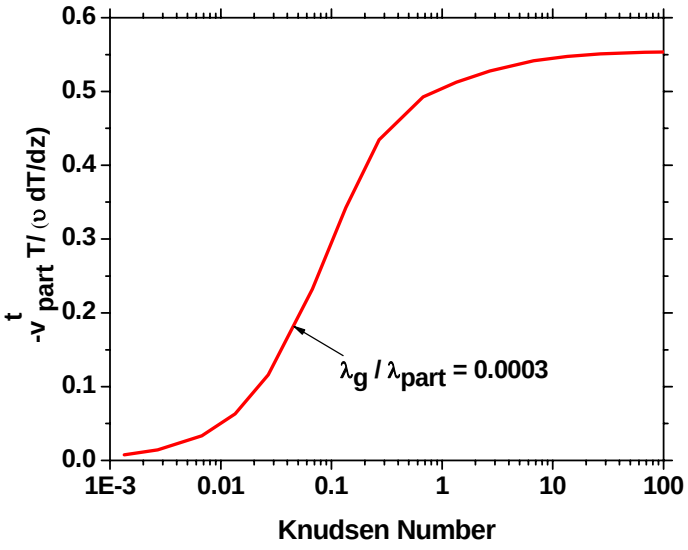

Figure 8.2 Dimensionless terminal velocity from the thermophoretic force as a function of the particle Knudsen number. Ratio of the gas thermal conductivity to the particle thermal conductivity is 0.0003. Calculation is for air at 300 K and 1 atm.

where α_t and α_m are the thermal and momentum accommodation coefficients, and within CADS have been assigned values of $C_t = 2.20$ and $C_m = 1.14$. Also, $C_s = 1.17$.

The high Knudsen number limit may be taken for Eqn. (8.18) to see how closely it agrees with the free-molecular limiting expression Eqn. (8.5). The result,

$$\boldsymbol{F}_{thermophoresis} = -\frac{8C_s r_{part}^2}{C_m}\frac{1}{\bar{c}}\frac{1}{5}\lambda\nabla T \tag{8.20}$$

is actually very close to Eqn. (8.5). The difference is a factor of $(3\lambda C_s)/(4C_m\lambda_{tran})$.

And, therefore, Eqn. (8.18) may be used as the basis of an interpolation formula over the entire range of Knudsen numbers. However, following the example of Talbot et al. [47] who have utilized this and compared the resulting expression to experimental data, Eqn. (8.18) is adjusted so that it agrees with Eqn. (8.5) in the molecular limit, at least up to a factor C_s/C_m. The final equation, which is used within CADS, is Eqn. (8.21).

$$\boldsymbol{F}^{Talbot}_{thermophoresis} = -6\pi\mu R\left(\frac{2C_s(\lambda + C_t(\mathrm{Kn})\lambda_{part}))}{(1 + 3C_m(\mathrm{Kn}))(2\lambda + \lambda_{part} + 2C_t(\mathrm{Kn})\lambda_{part}))}\right)\frac{\mu}{\rho T}\nabla T \tag{8.21}$$

Talbot's agreement with the data was generally adequate, yielding agreement of the thermophoresis velocity calculated using the Millikan drag formula for the friction factor at the 20% level.

Using Eqn. (8.14), Eqn. (8.21), and assuming a zero mass-averaged velocity, the dimensionless terminal velocity,

$$\frac{-v_{part}^{t}\rho T}{\mu(dT/dz)},$$

where v_{part}^{t} is the terminal velocity, in air at 300 K is calculated as a function of the particle Knudsen number in Fig. (8.2). The dimensionless terminal velocity is in good agreement with a similar figure from Friedlander [p. 51, ref. 2], for low ratios of the gas to particle thermal conductivities.

8.5 Sectional averages

In previous sections, the value of the brownian diffusion coefficient D^b has been shown to be a strong (and decreasing function) of the particle size. In order to employ the particle diffusion coefficient in particle transport equations, D^b, and the other transport coefficients, must be averaged over each section as well taking into account of the current monomer unit distribution in the section, $q_t(\mu)$. Eqn. (8.22) is the sectional average for section *i*.

$$D_i^b = \frac{\int_{\mu_i}^{\mu_{i+1}} q_t(\mu) D^b(\mu) d\mu}{\int_{\mu_i}^{\mu_{i+1}} q_t(\mu) d\mu} \tag{8.22}$$

$v_{part,\,i}^{t}$, the terminal velocity due to external forces on particles of size μ within section *i*, is also sectionally averaged. The gravity force and the thermophoretic force are examples of external forces.

$$v_{part,\,i}^{t} = \frac{\int_{\mu_i}^{\mu_{i+1}} q_t(\mu)\boldsymbol{c}(\mu)d\mu}{\int_{\mu_i}^{\mu_{i+1}} q_t(\mu)d\mu} \tag{8.23}$$

9. TDcads: Time Dependent CADS

9.1 Introduction

TDcads is a zero-dimensional application that solves the time-dependent conservation equations for species, particle unknowns, and temperature, in an enclosed volume or in a CSTR.

The primary purpose of TDcads is to test the CADS package in a 0D setting thoroughly before advancing to a one-dimensional (1D) setting. It employs a simple first order predictor-corrector backwards Euler algorithm for time, even though, in some cases an explicit time stepping scheme would be warranted. The reason for the use of the implicitly backwards-Euler scheme is that the 1D applications that CADS are targeted for must also use an implicit time-stepping solution scheme. TDcads has been designed with an implicit scheme so that it may be used as a test platform. In that capacity, it has proven to be extremely valuable. It has shown the necessity of using a semi-analytical jacobian due to both accuracy and speed concerns. Additionally, TDcads has been used to set up several validation examples to demonstrate that the CADS package can solve the aerosol particle dynamics equations robustly, accurately, and verifiable.

The unknowns and equation set are described below.

9.2 Equations

A list of the independent unknowns in the problem is $\Xi[b_i^L, b_i^H, Z_{i,\,k\neq 0}]$, presented in Eqn. (2.15) and Eqn. (2.16). Eqn. (9.1) is the conservation equation for particles in section *i*. S_i^{part} is the source term for particles in section *i;* it has units of kmol particles m^{-3} s^{-1}. N_i is the concentration of particles in section *i* and has units of (kmol particles) m^{-3}. N_i and Q_i are linear combinations of b_i^L and b_i^H The "m^{-3}" in the units expression denotes per cubic meter of gas.

$$\frac{d(N_i V)}{dt} = S_i^{part} V \tag{9.1}$$

$$\frac{d(Q_i V)}{dt} = S_i^{TMU} V \tag{9.2}$$

$$\frac{d(Z_{i,\,k} V)}{dt} = S_i^{\mathrm{MU\_k}} V \tag{9.3}$$

Eqn. (9.1), the particle conservation equation, is only used in the DOF 2 simulations. In the DOF 1 simulations, it is ignored. However, all particle source term kernels accurately account for particle number changes in the DOF 1 case at the expense of increased numerical diffusion. Eqn. (9.2) is the expression for the total monomer unit balance in section *i*. Q_i is the total concentration of total monomer units (TMU) in section *i*; it has units of (kmol TMU) m^{-3}. S_i^{TMU} is the source term for total monomer units for section *i*; it has units of (kmol monomer units) m^{-3} s^{-1}. Eqn. (9.3) is the conservation of monomer units of type $k > 0$ in section *i*. $Z_{i,k}$ is the concentration of monomer units of type *k* in section *i*; it has units of (kmol monomer units of type *k*) m^{-3}. $S_i^{\mathrm{MU\_k}}$ is the source term for monomer units of type $k > 0$ in section *i*; it has units of (kmol monomer units k) m^{-3} s^{-1}. The balance on monomer units of type $k = 0$ is not used, as it would lead to a singular equation system.

When the gas is included, the conservation equation for gas species *k* is added to the equation set, Eqn. (9.4). C_j, the independent unknown, is the molar concentration of gas species *j*.

$$\frac{d(C_j V)}{dt} = (S_j^{gas} + W_j^{gas})V \tag{9.4}$$

W_j^{gas} is the source term (kmol m^{-3} s^{-1}) due to gas phase reactions. S_j^{gas} is the source term (kmol m^{-3} s^{-1}) for gas phase species from interactions with the particle phase.

Derivatives are taken in a “space-time manner”, and a predictor-corrector backwards Euler method is employed. Therefore, on going from step $n-1$ to *n*, the following equations are used:

$$\frac{C_j^n V^n - C_j^{n-1} V^{n-1}}{\Delta t^n} = (S_j^{gas} + W_j^{gas})V \tag{9.5}$$

$$\frac{N_i^n V^n - N_i^{n-1} V^{n-1}}{\Delta t^n} = S_i^{part,\,n} V^n . \tag{9.6}$$

$$\frac{Q_i^n V^n - Q_i^{n-1} V^{n-1}}{\Delta t^n} = S_i^{TMU} V^n \tag{9.7}$$

$$\frac{Z_{i,k}^n V^n - Z_{i,k}^{n-1} V^{n-1}}{\Delta t^n} = S_i^{\mathrm{MU\_k}} V^n \tag{9.8}$$

An extra equation for V is used when the gas concentrations are followed. Under constant pressure conditions, the extra equation tracks the volume:

$$\frac{dV}{dt} = (\sum S_j^g + W_j^{gas})V\frac{RT}{P} \tag{9.9}$$

It's discrete form is:

$$\frac{V^n - V^{n-1}}{\Delta t^n} = (\sum S_j^{g,n} + W_j^{gas,n})V^n\frac{RT}{P}. \tag{9.10}$$

Under constant volume conditions, there is no extra equation. The volume is assumed to be a constant $V = 1\ \mathrm{m}^3$. Internally, whenever Cantera is updated, the ideal gas equation is used to determine the current pressure, given the temperature and the complete set of gas-phase species concentrations. Cantera can accurately track the pressure under constant-volume conditions.

An additional equation must be used to close out the formulation of the gas-phase species. There are several different formulations for this equation.

In formulation #1, used for constant pressure cases, one of the gas phase species conservation equations, corresponding to the species with the highest concentration, is replaced by the expression for the "time dependent" ideal gas equation of state in terms of the independent variables, C_j and T, Eqn. (9.11).

$$\sum_{j=1}^{N} \frac{dC_j}{dt} = -\frac{P}{RT^2}\frac{dT}{dt} \tag{9.11}$$

In formulation #2, no special equation is employed. Instead it's recognized that if you sum up Eqn. (9.4) over all j and add in Eqn. (9.9), Eqn. (9.11) results. Formulation #2 is the default for the constant pressure case, and has been shown to lead to fairly good element conservation properties despite the fact that the volume is being solved for.

Formulation #3 uses the constant volume case. Eqn. (9.11) is replaced by the trivial case of Eqn. (9.12).

$$\frac{V^n - V^{n-1}}{\Delta t^n} = 0 \tag{9.12}$$

All N species equations are kept. Then, the pressure is solved for implicitly via the ideal gas equation of state. The constant volume case conserves elements up to round-off error.

9.3 Specification of the Equation Unknowns

Unknowns that describe the particle distribution are the coefficients, b_i^L and b_i^H, for the sectional basis functions in the interpolation of the total monomer unit concentration within a section, $q_i(\mu)$:

$$q_i(\mu) = b_i^L \phi_i^L(\mu) + b_i^H \phi_i^H(\mu) \tag{9.13}$$

where

$$\phi_i^L(\mu) = \frac{\mu_{i+1} - \mu}{\mu_{i+1} - \mu_i} \text{ and } \phi_i^H(\mu) = \frac{\mu - \mu_i}{\mu_{i+1} - \mu_i} \tag{9.14}$$

Note,

$$Q_i = \int_{\mu_i}^{\mu_{i+1}} q_i(\mu) d\mu \tag{9.15}$$

Q_i is the total monomer unit concentration in section i. The total concentration of monomer units of type $k > 0$ in section i, $Z_{i,k}$, is also an independent unknown.

For the gas, the concentrations of species j from 1 to the number of species, C_j, are the unknowns. The temperature is the unknown in the energy equation, which is expressed in total enthalpy form. The pressure is not an independent variable in the formulation. For constant pressure boundary conditions, it is assumed. For constant volume boundary conditions, the pressure is calculated from the N species concentrations, the temperature, and the ideal gas law.

When the energy equation is added to the equation system, the temperature of the reactor becomes an unknown. If the energy equation is not used, the temperature is assumed to be a given constant.

9.4 Adding the Energy Conservation Equation

Within `TDcads`, it is possible to specify either isothermal flow or include the evolution of the energy equation. The derivation of the energy equation is not as straightforward as it may seem,

due to particles and molecular species being treated differently in terms of the “flow work” that they perform. First, the derivation of total internal energy in the particle phase is carried out. The internal energy of the particle phase is used instead of the enthalpy, because particles don’t participate in any volume expansion work done by the gas, using the current level of approximation where particle concentrations don’t participate in the overall gaseous equation of state. However, differences between enthalpies and internal energies for solids, which are roughly equal to pV_{solid}, are relatively small, because the solid molar volumes are very relatively small compared to their gas counterparts.

The total internal energy of the particle phase, QU^{part}, an extensive thermodynamic quantity, is evaluated by summing up over individual particles sections:

$$Q_T u^{part} = \sum_i \int_{\mu_i}^{\mu_{i+1}} n(\mu) u_i^{\text{part}}(\mu) d\mu = \sum_i Q_i u_i^{part}. \tag{9.16}$$

$n(\mu)$ is the concentration of particles having monomer unit size μ. $u_i^{part}(T, P, X_{i,k})$ is the intrinsic (i.e., on a per kmol MU basis) internal energy of particles in the i^{th} section bin. It is constant within the bin, since $X_{i,k}$, the mole fraction of monomer units within the bin, is assumed constant within the bin, and can be taken out of the integral, leaving Q_i, the total concentration of monomer units as a straight multiplier. No surface energy contributions to the total particle energy are used here; if there were, the evaluation of Eqn. (9.16) would follow along similar lines, but with additional surface tension terms that would have to be integrated over the distribution within a section.

An expression for the total mixture enthalpy of the gas and particle mixed-phase system is sought. The best way to understand the derivation is to imagine initially that the particles are inert within a reacting gas medium. Their only influence on the energy equation would be from their effect on the combined mixture heat capacity of the multiple phase system. The governing equation for a constant pressure closed system would then be Eqn. (9.17):

$$\frac{d(Ch_{gas}V)}{dt} + \sum_i Q_i V \frac{d(u_i^{part})}{dt} = \dot{Q}_{surr} \tag{9.17}$$

Now, if we allow for creation of particles, then the energy that went into that creation must be subtracted from the enthalpy balance equation in Eqn. (9.17).

$$\frac{d(Ch_{gas}V)}{dt} + \sum_i Q_i^{\mathrm{TMU}} V \frac{d(u_i^{part})}{dt} = \dot{Q}_{surr} - \left(\sum_i u_i^{part} S_i^{\mathrm{TMU}} \right) V \tag{9.18}$$

This extra term on the right of Eqn. (9.18) is associated with the creation of particle mass inside the domain, and may be derived from a total enthalpy balance on surface growth terms, and has been documented before in codes such as `SPIN` which calculates an energy balance on a surface due to bulk film growth [50]. Adding Eqn. (9.2) multiplied by u_i^{part} summed over all bins, *i*,

$$\sum_i u_i^{part} \frac{d(Q_i^{\mathrm{TMU}} V)}{dt} = \left(\sum_i u_i^{part} S_i^{\mathrm{TMU}} \right) V, \tag{9.19}$$

where we have allowed for differences in the molar internal energies with respect to particle bins, *i*, into Eqn. (9.17), yields the energy balance for the entire system, Eqn. (9.20).

$$\frac{d(Ch_{gas}V)}{dt} + \sum_i \frac{d(Q_i^{\mathrm{TMU}} u_i^{part} V)}{dt} = \dot{\Sigma}_{surr} \tag{9.20}$$

C is the concentration of the gas. h_{gas} is the molar enthalpy of the gas mixture. $\dot{\Sigma}_{surr}$ is the heat transfer to the gas-particle mixture from the surroundings [49]. This includes radiative transfer. Qu^{part} is the total internal energy in the particle phase.

Eqn. (9.20) may be derived alternatively by constructing a rigorous energy balance over the gas phase including energy transport and enthalpy transport from the gas phase to the particle phase due to particle growth:

(9.21)

$$\frac{d(Ch_{gas}V)}{dt} = \sum_j S_j^{gas} \tilde{h}_k V - \sum_i (Q_i^{\mathrm{TMU}} V) \frac{d(u_i^{part})}{dt} + \dot{\Sigma}_{\mathrm{surr}}$$

The first term represents the loss of enthalpy from the gas phase due to gas molecules leaving the gas phase. In Eqn. (9.21), the second term on the rhs is the conductive heat transfer to the particle

phase, which can further be deduced to be equal to the sensible heat increase in the particle phase (disregarding radiation effects again).

Then, a rigorous energy balance may be conducted on the particle phase. The total change in energy in the particle phase is due to the energy exchanged due to the surface growth reaction, S_i^{TMU}, and the energy due to conduction to the particle phase:

$$\sum_i \frac{d(Q_i^{\mathrm{TMU}} u_i^{part} V)}{dt} = \left(\sum_i S_i^{\mathrm{TMU}} u_i^{part} \right) V - Q_{\mathrm{gas\_to\_part}} . \tag{9.22}$$

Addition of Eqn. (9.21) and Eqn. (9.22), with the added simplification that the total enthalpy for the surface reaction process is conserved, i.e.,

$$\sum_i S_i^{\mathrm{TMU}} u_i^{part} = -\sum_j S_j^{gas} \tilde{h}_j V, \tag{9.23}$$

again yields the combined total mixture enthalpy formulation, Eqn. (9.20).

For the constant volume case, Eqn. (9.24), where the gas enthalpy term has been replaced by a gas-phase internal energy term, holds. There is no pV expansion for the constant volume case.

$$\frac{d(Cu_{gas} V)}{dt} + \frac{d(Qu^{part} V)}{dt} = \dot{\Sigma}_{surr} \tag{9.24}$$

$\dot{\Sigma}_{surr}$ is the rate of heat exchange with the surroundings, and is equal to zero if the adiabatic approximation holds.

9.5 Adding in a CSTR Approximation

The starting point are the equations given in Coltrin and Kee [49], extended for the presence of a particle phase. Constant volume and constant pressure are assumed in the CSTR approximation. The expression for the conservation of gas phase species *j* is:

$$\frac{d(C_j V)}{dt} + \dot{F}^{\mathrm{out}} X_j - \dot{F}^{\mathrm{in}} X_j^{\mathrm{in}} = (S_j^{gas} + W_j^{gas}) V \tag{9.25}$$

$\dot{F}^{\text{in}}$ is the molar flow rate into the reactor, and $\dot{F}^{\text{out}}$ is the molar flow rate out of the reactor. Normally, Eqn. (9.25) is expressed on a mass basis, because under steady state conditions the mass flow rates into the PSR reactor is equal to the mass flow rate out of the reactor. This is not the case for molar flow rates. When particles are included, the mass flow rates incoming and outgoing are equal at steady state if the sum is carried out over both gas and particle phases.

One can add up the individual ways that volume is created and lost within the reactor, and use the constant pressure assumption to derive an explicit expression for how the molar flux out, $\dot{F}^{\text{out}}$, varies with the molar flux in, $\dot{F}^{\text{in}}$. This maybe done by summing up Eqn. (9.25) over all species *j* and then by using the ideal gas law.

$$\begin{aligned} \dot{F}^{\text{out}} &= \dot{F}^{\text{in}} + \sum_j (S_j^{gas} + W_j^{gas})V - \frac{d(CV)}{dt} \\ &= \dot{F}^{\text{in}} + \left(\sum_j (S_j^{gas} + W_j^{gas})V \right) + \frac{CV}{T}\frac{d(T)}{dt} \end{aligned} \tag{9.26}$$

This gives an explicit equation for $\dot{F}^{\text{out}}$ that may be plugged back into Eqn. (9.25). One of the species conservation equations must be replaced by the sum of the mole fractions equals one condition in order to avoid a singular equation system.

The equation for the particle species may be derived fairly easily once $\dot{F}^{\text{out}}$ is determined:

$$\frac{d(N_i V)}{dt} = S_i^{part} V + \dot{F}^{\text{in}}_{part,\,i} - \frac{\dot{F}^{\text{out}}}{C} N_i \tag{9.27}$$

$$\frac{d(Q_i V)}{dt} = S_i^{TMU} V + \dot{F}^{\text{in}}_{\text{TMU},\,i} - \frac{\dot{F}^{\text{out}}}{C} Q_i \tag{9.28}$$

$$\frac{d(Z_{i,\,k} V)}{dt} = S_i^{\text{MU\_k}} V + \dot{F}^{\text{in}}_{\text{mu\_k},\,i} - \frac{\dot{F}^{\text{out}}}{C} N_i \tag{9.29}$$

$\dot{F}^{\text{in}}_{part,\,i}$ is the molar flux of particles in section *i* entering the reactor; it has units of (kmol particles) s^{-1}. The other quantities are defined analogously.

The energy equation, including the particle phase contributions, is given in Eqn. (9.30).

$$\frac{d(Cu^{gas}V)}{dt} + \frac{d(Qu^{part}V)}{dt} = \dot{\Sigma}_{surr}$$
$$-\frac{\dot{F}^{\text{out}}}{C}\sum_i Q_i u_i^{part} - \dot{F}^{\text{out}} h_{gas} + \sum_i (\dot{F}_{TMU}^{\text{in},\,i})(u_i^{part,\,in}) + \dot{F}^{\text{in}} h_{gas}^{\text{in}} \tag{9.30}$$

The particle terms in Eqn. (9.30) can be derived in two ways. The first way would be to start with Eqn. (9.24), and then add in the enthalpy flux terms for the gas and the internal energy flux terms for the particle phase, representing the inflow of energy in the inlet and the outflow of energy via the outlet flow.

The second method would be to start with a consistent energy equation for the CSTR approximation for just the gas phase only, with additional terms representing the sensible heat of the particle particle phase. Then, the addition of Eqn. (9.28) multiplied by the internal energy of the i^{th} section yields Eqn. (9.30), after summation over all particle bins.

9.6 Brief Description of the Code Layout

The main routine where the residuals are calculated is the member function `evalResidNJ()` of the C++ object `resid_adsm`, located in the file `tdcads.cpp`. The object `resid_adsm`, is derived from the class `ResidJacEval`, which is the C++ object that is need by the backwards Euler C++ object, `BEulerInt`.

The `resid_adsm` object also has a member function, `evalJacobian()`, which is responsible for calculating the jacobian of the residual in `evalResidNJ()`. It does this by calling functions which return jacobians of the individual source terms that contribute to Eqns. (9.1) to (9.8).

To add another equation or modify the existing equations, the primary routine that needs to be altered are in the file `TDcads.cpp`. The primary C++ object that needs to be changed is the object `resid_adsm`. It contains, for example, the number of unknowns in the time dependent residual.

9.7 Input File Options

Specific input file options for `TDcads` are separated into 3 main blocks: `TDcads Model Definition`, `TDcads Time Step Parameters`, and `TDcads Initial Conditions`. These blocks are in addition to the `Section Model Definition` block and the `ADS Model Definition` block described in the previous chapters of this manual. The overall format of the input file, which is

```
START BLOCK TDCADS MODEL DEFINITION
   !Include Gas Species in Solution Vector = bool (required)
   !   Defaults = false -> implies constant gas composition
   Include Gas Species in Solution Vector = no
   ! ------------------------------------------------------------
   ! Include Temperature in Solution Vector = bool (required)
   !   Defaults = false -> implies isothermal conditions
   Include Temperature in Solution Vector = no
   ! ------------------------------------------------------------
   ! Boundary Conditions = [ constant pressure | constant volume | psr ] (required)
   !  Default = constant pressure
   Boundary Conditions = constant pressure
   ! ------------------------------------------------------------
   ! Volume of Reactor = [ double ] unit_of_length^3
   !      (optional) (default = 1.0E-6 m^3 = 1 cm^3
   Volume of Reactor = 1.0E-6 m^3
   ! ------------------------------------------------------------
   ! Special Gas Species - string (defalt = 0) (required)
   Special Gas Species = N2
! ------------------------------------------------------------
   START BLOCK TDCADS INFLOW CONDITIONS
       Inflow Temperature = 300.
       ! ------------------------------------------------------
       ! Inflow Pressure = [double] [Pressure_Units]
       !  (default = 1.0 atm) (optional)
       Inflow Pressure = 1.0 atm
       ! ------------------------------------------------------
       ! Inflow Molar Flow Rate = [double]
       !  (default = none) (required) (units = kmol s-1)
       Inflow Molar Flow Rate = 1.0
       !-------------------------------------------------------------
       START BLOCK INFLOW GAS MOLE FRACTIONS
           N2   = 0.90
           C2H2 = 0.10
       END BLOCK INFLOW GAS MOLE FRACTIONS
       !-------------------------------------------------------------
       ! Particle Initial Conditions BLOCK
       !    (optional)
       !    In this block we specify the initial conditions for
       !    the particle unknowns in an additive format.
       !    The additive format here indicates that we may
       !    have more than one distribution block and the
       !    compositions will add.
       START BLOCK PARTICLE INLET CONDITIONS
           START BLOCK PARTICLE MONOMER DISTRIBUTION
           END BLOCK PARTICLE MONOMER DISTRIBUTION
           START BLOCK Particle Gaussian Distribution
           END BLOCK Particle Gaussian Distribution
           START BLOCK Inflow Particle Raw Solution Vector
                Particle File = SolnFile.txt
           END BLOCK Inflow Particle Raw Solution Vector
       END BLOCK PARTICLE INLET CONDITIONS
   END BLOCK TDCADS INFLOW CONDITIONS
END BLOCK TDCADS MODEL DEFINITION
```

Figure 9.1 Input file for the TDcads Model Definition block

made up of collections of distinct blocks, is best presented in tutorial problems which is distributed with the code.

The TDcads model definition block sets up the problem type and prescribes what unknowns are in the solution vector. Figure 9.1 contains the options for this block. The first two keylines state whether gas phase concentrations and the temperature are included in the solution vector. If they

are not included, then their values are assumed to be constant. The `Boundary Condition` keyline specifies how the gas conditions change as a function of the reactions. A constant pressure condition means that the volume of gas tracked by the simulation changes as a function of time. The volume becomes part of the solution vector. If the simulation is constant volume, the pressure will change as a function of time. The pressure is determined by the ideal gas law at every time step. A PSR condition means that the reactor has an inflow and an outflow and is at constant pressure. The inflow conditions are specified by the `TDcads Inflow Conditions` block within the `TDcads Model Definition` block; the inflow block becomes a required block. The outflow conditions are solved for.

The `Volume of Reactor` keyline specifies the initial volume of the reactor. The default value is 1 cm^{-3}. The `Special Gas Species` keyline specifies the special gas phase species identity. In some formulations, this species has a different equation applied to it than the rest of the species in order to conserve total mass for example. Usually, the species with the greatest concentration is set as the special gas-phase species.

The `TDcads Inflow Conditions` is a required block when the PSR boundary condition is chosen, and it is an error to include this block when the PSR boundary condition is not chosen. In the block, the inlet gas mole fraction vector, the inlet particle distribution, the inlet pressure, and the inlet temperature are specified. An example of the block format for specification of the inlet mole fractions is given in Figure 9.1. The format must have a species on the left hand side of the equals sign and a double value on the right hand side. The values of all mole fractions must sum up to (or be very close to) one; they are normalized exactly to a sum of one within the code.

The `Particle Inlet Conditions` block specifies the particle content of the inlet stream. Options for a equi-sized distribution, a gaussian distribution, and a distribution specified by input from an ascii file are possible. All blocks within this section are additive in their effect. Figure 9.2 provides a detailed description of options in this block.

The `Particle Monomer Distribution` block will install a given concentration of particles all with the same size into the solution vector. Note, This may entail putting distributions into two adjacent bins to obtain the correct total number to size ratio overall. The monomer size is specified with the `Monomer Size` keyline, which accepts a double with an optional unit of length conversion string (the default is always to employ MKS units). The particle number concentration is specified with the `Monomer concentration` keyline, which accepts a double with an optional unit of concentration. The default units for concentration is kmol particles m^{-3}. The `MU mole fraction` block specifies the mole fraction of the particles in the inlet stream.

```
START BLOCK PARTICLE INLET CONDITIONS
   START BLOCK PARTICLE MONOMER DISTRIBUTION
        !  Monomer size = [dbl] [Length unit]
        !      (required)
        !      (Length unit defaults to MKS -> meters)
        !     This card specifies the diameter of the particle.
        Monomer size = 1 nm
        !-----------------------------------------------------
        !  Monomer Concentration = [dbl] [Concentration unit]
        !      (required) (concentration unit defaults to MKS -> kmol/m3)
        !     This card specifies the concentration of the particles.
        Monomer concentration = 1.0E16 molec/cm3
        !-----------------------------------------------------
        !  Block format for specifiation of the mole fractions of the particle phase
        START BLOCK MU Mole Fractions
             C2H2 = 1.0
        END BLOCK MU Mole Fractions
        !-----------------------------------------------------
   END BLOCK PARTICLE MONOMER DISTRIBUTION
   START BLOCK Particle Gaussian Distribution
        !  Monomer Unit Peak = [dbl]
        !      (required)
        ! This card specifies the number of monomer units at the peak of the distribution.
        ! This value is in # molecules.
        Monomer Unit Peak = 30
        !-----------------------------------------------------
        !  Monomer Unit Spread = [dbl]
        !      (required)
        !     This card specifies the gaussian spread of the distribution. This value is in # molecules.
        Monomer Unit Spread = 30
        !-----------------------------------------------------
        ! Monomer Concentration = [dbl] [Concentration unit]
        !      (required)
        !     This card specifies the net concentration of the particles.
        Monomer concentration = 1.0E16 molec/cm3
        !-----------------------------------------------------
        !  Block format for specifiation of the mole fractions of the particle phase
        START BLOCK MU Mole Fractions
             C2H2 = 1.0
        END BLOCK MU Mole Fractions
   END BLOCK Particle Gaussian Distribution
   !-------------------------------------------------------
   ! Direct specification of the particle solution vector via an ascii file
   START BLOCK Inflow Particle Raw Solution Vector
        Particle File = SolnFile.txt
   END BLOCK Inflow Particle Raw Solution Vector
   !-------------------------------------------------------
   ! Default Small Positive Section Value = bool
   !         (optional)
   !   This optional boolean will set a small positive value for the quantity of particles in each
   !   of the sectional bins. Currently, it is set to 1.0E-200. This value sometimes enhances the
   !   startup of the simulation.
   ! Default Small Positive Section Value = true
   END BLOCK PARTICLE INITIAL CONDITIONS
```

Figure 9.2 Particle Initial Conditions block

The `Particle Gaussian Distribution` block installs a given concentration of particles with a gaussian distribution of sizes, Eqn. (9.31).

$$N(\mu) = \frac{N_o}{\sqrt{\pi}(\Delta\mu_s)} \exp\left[-\frac{(\mu-\mu_o)^2}{\Delta\mu_s{}^2}\right] \tag{9.31}$$

```
START BLOCK TDCADS TIME STEP PARAMETERS
   ! Initial Time Step - double (required)
   Initial Time Step = 1.0E-11
   !-------------------------------------------------------------
   ! Final Time - double(required)
   Final Time = 0.01
   ! -------------------------------------------------------------
   !  Analytic or Numerical Jacobian = [ analytic, numerical ]
   !   (optional) Default = numerical
   Analytic or Numerical Jacobian = analytic
       ! -------------------------------------------------------------
   ! Maximum Number of Time Steps - Int (default large) (optional)
   !  Maximum Number of Time Steps = 10000
   ! -------------------------------------------------------------
   ! Relative Time Step Error Tolerance = dbl (optional)
   !  Set the relative time step error tolerances
   !       Default = 1.0E-3 Limits = 0.5 > rtol > 1.0E-9
   !  Relative Time Step Error Tolerance = 1.0E-3
   ! -------------------------------------------------------------
   ! Absolute Time Step Error Tolerance = dbl (optional)
   !  Set the absolute time step error tolerances
   !  Default = 1.0E-6 (Limits = 1.0E-4 > atol > 1.0E-60)
   !  Absolute Time Step Error Tolerance = 1.0E-6
   ! -------------------------------------------------------------
   ! Number of Constant Delta T Time Steps = int (optional)
   !  Number of Constant Delta T Time Steps = 0
   ! -------------------------------------------------------------
   ! PrintFlag - Int (default 1) (optional)
   !  Set the amount of printing from each time step 1 to 3
   PrintFlag = 1
   ! -------------------------------------------------------------
   !  Print Solution Every N Steps  - int (default 1) (optional)
   !   Set the time step interval at which the solution will will be printed. This defaults
   !   to printing at every time step
   ! Print Solution Every N Steps = 1
   ! -------------------------------------------------------------
   !  Print Solution at N Regular Intervals - int ( default 0) (optional)
   !  Prints the solution at n regular intervals wrt to the final time, TOUT.
   ! Print Solution at N Regular Intervals = 10
   ! -------------------------------------------------------------
   !  Print Solution for first N Time Steps - int (default 0) (optional)
   ! Print Solution for first n Time Steps = 10
   ! -------------------------------------------------------------
   !  Minimum number of newton iterations = int (default 0) (optional)
   !  Minimum number of newton iterations = 0
   ! -------------------------------------------------------------
   !  Matrix Conditioning = boolean ( default = false) (optional)
   !  Matrix Conditioning = false
   ! -------------------------------------------------------------
   !  Matrix Row Scaling = boolean ( default = true) (optional)
   !  Matrix Row Scaling = true
   ! -------------------------------------------------------------
   !  Matrix Column Scaling = boolean ( default = false) (optional)
   !  Matrix Column Scaling = false
   !-------------------------------------------------------------
   !  Dump Jacobians to Disk = boolean ( default = false) (optional)
   !  Dump Jacobians to Disk = false
   !-------------------------------------------------------------
END BLOCK TDcads TIME STEP PARAMETERS
```

Figure 9.3 `TDcads` Time Step Parameters block

μ_o is the value given in the `Monomer unit peak` keyline. $\Delta\mu_s$ is the standard deviation of the distribution and is given in the `Monomer unit spread` keyline. N_o is the value given in the `Mono-`

mer concentration keyline. There is also a MU Mole Fraction block to specify the monomer unit mole fraction in the gaussian distribution.

The Inflow Particle Raw Solution Vector block installs a user specified distribution of particles. These are read in from a file specified within the block. The format of this file is a straight ascii input of the particule state vector, with one double value per line.

Figure 9.3 contains options for the implicit time stepping algorithm within TDcads. The Initial Time Step keyline sets the initial time step within TDcads. The Final Time keyline sets the final time of the simulation. The Analytical or Numerical Jacobian keyline sets the type of jacobian approximation to be used. The Relative Time Step Error Tolerance and Absolute Time Step Error Tolerance keylines sets the overall magnitudes of the relative and absolute error tolerances in both the nonlinear convergence criteria and the time-step truncation error tolerance criteria.

The Number of Constant Delta T Time Steps keyline sets the number of initial time steps that will be taken where the time step error control will not be applied. The default is zero.

The Print Solution Every N Steps keyline sets the time step interval at which the solution will be printed. This defaults to printing at every time step. Setting this to zero will turn off this trigger for printing. There are two other triggers for printing of the solution. The Print Solution at N Regular Intervals option prints out the solution at equispaced time intervals. The Print Solution for first n Time Steps option prints out the solution for the specified number of initial time steps.

The Minimum number of newton iterations keyline turns on the capability for ensuring that a minimum number of newton iterations are carried out at each time step. This option is useful for checking the quality of jacobian approximations.

The Matrix Conditioning keyline turns on the capability for matrix preconditioning. For DOF 1 methods, this option is ignored. For DOF 2 section models, a small block diagonal matrix, the mass matrix, is used as a preconditioner before the large matrix is inverted. This is an experimental option.

The Matrix Row Scaling keyline turns on the capability for matrix row scaling. Before a nonlinear iteration, the matrix and the right hand side is row sum scaled before being given to the linear solver. This has the effect of reducing the condition number of the linear system, and therefore its use increases the accuracy of the linear solver step. By default, matrix scaling is turned on.

The `matrix column scaling` keyline turns on the ability to carry out matrix column scaling during the linear solve step. For example, when both row and column scaling are turned on, a Newton iterative step, which could previously be described by

$$\boldsymbol{J}\big|_{\boldsymbol{x}_n}(\Delta \boldsymbol{x}) = -\boldsymbol{R}(\boldsymbol{x}_n),$$

where $\Delta \boldsymbol{x} = \boldsymbol{x}_{n+1} - \boldsymbol{x}_n$ is the solution update at step *n*, $\boldsymbol{J}\big|_{\boldsymbol{x}_n}$ is the jacobian at step *n*, and $\boldsymbol{R}(\boldsymbol{x}_n)$ is the residual at step *n*, is altered to the following formulation:

$$[\boldsymbol{RS}^T(\boldsymbol{J}\big|_{\boldsymbol{x}_n})\boldsymbol{C}]\boldsymbol{C}^T(\Delta \boldsymbol{x}) = -(\boldsymbol{RS}^T)(\boldsymbol{R}(\boldsymbol{x}_n)) \tag{9.32}$$

$\boldsymbol{C}$ is the column scaled matrix obtained by taking the weighting vector described in Section 2.12 and multiplying by a factor of 10^4. $\boldsymbol{RS}$ is the row sum scaled vector, obtained by row sum scaling the matrix $(\boldsymbol{J}\big|_{\boldsymbol{x}_n})\boldsymbol{C}$. The transformed matrix in brackets, $[\boldsymbol{RS}^T(\boldsymbol{J}\big|_{\boldsymbol{x}_n})\boldsymbol{C}]$, has a much lower condition number as determined by lapack routines than the original matrix. The transformed update vector, $\boldsymbol{y} = \boldsymbol{C}^T(\Delta \boldsymbol{x})$, is scaled appropriately so that each component of $\boldsymbol{y}$ has the same order or magnitude in relation to how well that component satisfies its individual convergence requirement.

The `Dump Jacobians to Disk` keyline turns on the capability for dumping jacobians to disk. This dump is in a comma separated format that is easily readable by a spreadsheet program.

Figure 9.4 contains the options for specifying the initial conditions of a `TDcads` simulation. Gas temperature, gas pressure and gas composition must be specified. As previously discussed around Figure 9.2, initial particle profiles are derived either from a mono-dispersed distribution or from a gaussian profile. The `Read Solution Vector` block may be used to completely specify the initial condition. The file `SolnFile.txt` is written out at every time step from `TDcads`. The `Read Solution Vector` option may be used as a restart capability.

9.8 Conservation Properties of the Space-Time Derivative

A question has come up about the use of the space-time derivative in Eqns. (9.1) to (9.30) instead of a different time discretization. I will show space-time derivative conserves elements during a time step for any amount of time-step truncation error, while the normal approximation only conserves elements in the limit of zero time-step truncation error.

```
START BLOCK TDcads INITIAL CONDITIONS
   ! Gas Temperature - double (no default) (required)
   !    Specify the gas temperature (Kelvin)
   Gas Temperature = 1800.
   !-------------------------------------------------------------
   ! Gas Pressure - double [Pressure_unit] (no default) (required)
   !       Specify the pressure - pressure units default to MKS unless specified.
   Gas Pressure = 1.0 atm
   !-------------------------------------------------------------
   !  Initial Gas Mole Fractions ( required )
   !        Block format for specifiation of the mole fractions
   !        of the gas phase.
   START BLOCK INITIAL GAS MOLE FRACTIONS
       N2   = 0.90
       C2H2 = 0.10
   END BLOCK INITIAL GAS MOLE FRACTIONS
   !-------------------------------------------------------------
   ! Particle Initial Conditions BLOCK (optional)
   !    In this block we specify the initial conditions for
   !    the particle unknowns in an additive format.
   START BLOCK PARTICLE INITIAL CONDITIONS
   !-----------------------------------------------------------
       START BLOCK PARTICLE MONOMER DISTRIBUTION
            Monomer size = 1 nm
            Monomer concentration = 1.0E16 molec/cm3
            START BLOCK MU Mole Fractions
                 C2H2 = 1.0
            END BLOCK MU Mole Fractions
       END BLOCK PARTICLE MONOMER DISTRIBUTION
       START BLOCK Particle Gaussian Distribution
            Monomer Unit Peak = 30
            Monomer Unit Spread = 30
            Monomer concentration = 1.0E16 molec/cm3
            START BLOCK MU Mole Fractions
                 C2H2 = 1.0
            END BLOCK MU Mole Fractions
       END BLOCK Particle Gaussian Distribution
       Default Small Positive Section Value = true
   END BLOCK PARTICLE INITIAL CONDITIONS
   ! Read Solution Vector Block
   !       Optional Block to override much of what was just
   !       input above, and supply the initial conditions
   !       for the solution vector
   !         (optional)
   !-------------------------------------------------------------
   ! START BLOCK Read Solution Vector
   !   Solution Vector File = {string} (required)
   !   Solution Vector File = SolnFile.txt
   ! END BLOCK Read Solution Vector
END BLOCK TDCADS INITIAL CONDITIONS
```

Figure 9.4 `TDcads` initial conditions block.

The space-time discretization of the volume-concentration product is done by treating the time dimension like a physical coordinate, constructing an integral around the domain, and then formulating a conservation property by accounting for the fluxes out of the domain. It's use has been attributed to Tayfoon Tezduyar and Tom Hughes [52], who have used the space-time derivative discretization in 2D and 3D finite element fluid mechanics codes. Using the integral approach described above, the discretized time derivative becomes:

$$\frac{dC_jV}{dt} = \frac{C_j^nV^n - C_j^{n-1}V^{n-1}}{\Delta t^n}. \tag{9.33}$$

n is the time step number. and Δt^n is the time step increment from time $n-1$ to n. The normal discretization of the volume concentration product is

$$\frac{dC_jV}{dt} = C_j\frac{dV}{dt} + V\frac{dC_k}{dt} = C_k^n\frac{V^n - V^{n-1}}{\Delta t^n} + V^n\frac{C_k^n - C_k^{n-1}}{\Delta t^n}. \tag{9.34}$$

The difference between these two expressions is

$$\zeta = (C_k^n - C_k^{n-1})\frac{(V^n - V^{n-1})}{\Delta t^n}, \tag{9.35}$$

which is the same- or higher-order term as the backwards Euler discretization error.

An example problem with two species may be used to test the elemental conservation property. Assume that there are two species, *A* and *B*, comprising a constant *T* and *P* ideal gas, with one reaction where $A \rightarrow 2B$ using a constant reaction rate constant of k. At each time step the conservation of elements property depends on the following equality holding:

$$(2C_A^n + C_B^n)V^n = (2C_A^{n-1} + C_B^{n-1})V^{n-1} \tag{9.36}$$

Is Eqn. (9.36) satisfied for each time derivative discretization method, using Eqn. (9.33) or Eqn. (9.34) to formulate the time derivative? The discretized form of the sample problem for the space-time derivative is:

$$\frac{C_A^nV^n - C_A^{n-1}V^{n-1}}{\Delta t^n} = -kC_A^nV^n \tag{9.37}$$

$$C_A^n + C_B^n = \frac{P}{RT} \tag{9.38}$$

$$\frac{V^n - V^{n-1}}{\Delta t^n} = (kC_A^n)V^n\frac{RT}{P} \tag{9.39}$$

Then, combining Eqn. (9.38) and Eqn. (9.39):

$$(C_A^n + C_B^n)\left(\frac{V^n - V^{n-1}}{\Delta t^n}\right) = (kC_A^n)V^n \tag{9.40}$$

Adding Eqn. (9.40) to Eqn. (9.37):

$$\frac{C_A^n V^n - C_A^{n-1}V^{n-1}}{\Delta t^n} + (C_A^n + C_B^n)\left(\frac{V^n - V^{n-1}}{\Delta t^n}\right) = 0 \tag{9.41}$$

Collecting V^n terms on one side of the equation and V^{n-1} term on the other side, Eqn. (9.42) is obtained.

$$\begin{aligned} 2C_A^n V^n + C_B^n V^n &= C_A^{n-1}V^{n-1} + (C_A^n + C_B^n)V^{n-1} \\ &= 2C_A^{n-1}V^{n-1} + (C_B^{n-1})V^{n-1} \end{aligned} \tag{9.42}$$

The relation, $C_A^n + C_B^n = C_A^{n-1} + C_B^{n-1}$, has been used to complete the proof. Thus, the space-time derivative does satisfy conservation of elements irrespective of the time-step truncation error.

The discretized form of the sample problem for the normal derivative formulation is:

$$C_A^n \frac{V^n - V^{n-1}}{\Delta t^n} + V^n \frac{C_A^n - C_A^{n-1}}{\Delta t^n} = -kC_A^n V^n \tag{9.43}$$

$$C_A^n + C_B^n = \frac{P}{RT} \tag{9.44}$$

$$\frac{V^n - V^{n-1}}{\Delta t^n} = (kC_A^n)V^n \frac{RT}{P} \tag{9.45}$$

Adding Eqn. (9.40) to Eqn. (9.43), the following is obtained:

$$C_A^n \frac{V^n - V^{n-1}}{\Delta t^n} + V^n \frac{C_A^n - C_A^{n-1}}{\Delta t^n} + (C_A^n + C_B^n)\left(\frac{V^n - V^{n-1}}{\Delta t^n}\right) = 0 \tag{9.46}$$

Collecting terms, the following is obtained:

$$
\begin{aligned}
2C_A^n V^n + C_B^n V^n &= 2C_A^n V^{n-1} + C_B^n V^{n-1} + C_A^{n-1} V^n - C_A^n V^n \\
&= C_A^{n-1} V^{n-1} + C_B^{n-1} V^{n-1} + C_A^n V^{n-1} + C_A^{n-1} V^n - C_A^n V^n \\
&= 2C_A^{n-1} V^{n-1} + C_B^{n-1} V^{n-1} \\
&\quad + (C_A^n V^{n-1} + C_A^{n-1} V^n - C_A^{n-1} V^{n-1} - C_A^n V^n) \\
&= 2C_A^{n-1} V^{n-1} + C_B^{n-1} V^{n-1} \\
&\quad + ((C_A^n - C_A^{n-1})(V^{n-1} - V^n))
\end{aligned}
\tag{9.47}
$$

Note, the statement of the element balance is off by the difference in the formulations of the time derivative. Therefore, the normal time derivative doesn't conserve elements when there is time-step truncation error, and thus is inferior to the space-time derivative formulation in a very fundamental way.

10. Example Calculations

10.1 Growth of a MonoDispersed Aerosol in a Bath of Constant Growth Medium

Consider an initial distribution of dilute particles in a medium. Let the medium initially consist of two species. The first species is inert, while the second species may undergo a simple growth-addition reaction with an aerosol phase. Then, an analytical solution to the rate of growth of the aerosol can be derived under several limiting assumptions. First let's assume that the initial particles have a monodispersed size distribution.

Let C_2 be the concentration of species 2 in the gas phase. Let N_T be the total particle density in the gas phase, expressed in kmol particles m^{-3}. Let Q_T be the total concentration of particles in the gas phase, expressed in kmol MU m^{-3}. Then, the average number of monomer units per particle is initially equal to

$$\mathrm{MU}_{init} = \frac{Q_T}{N_T}. \tag{10.1}$$

Let V_o be the molar volume of monomer units in the particle phase, expressed in m^3 $(\text{kmol particles})^{-1}$. Then, the initial total volume of particles in the gas phase, expressed as the volume fraction - m^3 particle m^{-3} gas, is equal to

$$V = V_o Q. \tag{10.2}$$

Each particle is assumed to be spherical. The initial volume of each particle is equal to $\mathrm{MU}_{init} V_o A_v^{-1}$. The radius of each particle is

$$r = \left(\frac{3\mathrm{MU}_{init} V_o A_v^{-1}}{4\pi} \right)^{1/3}, \tag{10.3}$$

and the surface area of each particle is

$$a = 4\pi \left(\frac{3\mathrm{MU} V_o A_v^{-1}}{4\pi} \right)^{2/3}. \tag{10.4}$$

Assume that the growth reaction is proportional to the surface area of the particles and the con-

centration of species 2. The rate of progress, ROP, of the reaction per particle surface area may be expressed as

$$ROP = kC_2\left[\frac{\text{kmol}}{\text{m}^2 s}\right], \tag{10.5}$$

where the rate constant, Eqn. (10.6), may be expressed in terms of the sticking coefficient, γ, and average gas-species velocity, $\bar{v}_2$.

$$k = \frac{1}{4}\bar{v}_2\gamma \tag{10.6}$$

The total rate of production of monomer units, MU's, is given by Eqn. (10.7).

$$\begin{aligned} \frac{dQ_T}{dt} &= (ROP)N_T a = kC_2 N_T a \\ \frac{dQ_T}{dt} &= kC_2 N_T 4\pi\left(\frac{3V_o A_v^{-1}}{4\pi}\right)^{2/3} Q_T{}^{2/3} = \kappa\left(\frac{Q_T}{N_T}\right)^{2/3} \end{aligned} \tag{10.7}$$

κ is introduced to adsorb all of the constants. In Eqn. (10.7), C_2 has been assumed to be a constant. Integrating Eqn. (10.7) yields Eqn. (10.8).

$$Q_T = \left(\frac{\kappa t}{3N_T^{2/3}} + Q_T^{o1/3}\right)^3 \tag{10.8}$$

Q_T^o is the initial total monomer units.

10.1.1 `TDcads` Solution

Let's fill in some of the constants to be used in the `TDcads` solution. `TDcads` was run in a constant pressure mode. The temperature and pressure of the simulation was set to 1000 K and 1 atm. The mole fraction of the condensable species was set to 0.9. Therefore,

$$C_2 = 1.097 \times 10^{-5}\ \frac{\text{gmol}}{\text{cm}^3} = 1.097 \times 10^{-2}\ \frac{\text{kmol}}{\text{m}^3}$$

Setting $M_2 = 39.95$, the mean thermal velocity of the condensing species is equal to

$$\bar{v}_2 = \left(\frac{8RT}{\pi M_2}\right)^{\frac{1}{2}} = 728.0\frac{\text{m}}{\text{s}}$$

Therefore, with $\gamma = 0.001$, the reaction rate is equal to

$$k = \frac{1}{4}\bar{v}_2\gamma = 0.182\frac{\text{m}}{\text{s}}.$$

The initial number density of particles has been set to

$$N_T^o = 1.0 \times 10^{-10}\ \frac{\text{gmol}}{\text{cm}^3} = 1.0 \times 10^{-7}\ \frac{\text{kmol}}{\text{m}^3} = 6.022 \times 10^{19}\ \frac{1}{\text{m}^3}\ .$$

Note, this isn't a constant in the `TDcads` simulation, because the constant pressure assumption requires the reactor volume to shrink. However, with the volume set to 1 cm^3, the number of particles in reactor, 1.0×10^{-13}, is a constant. The initial monomer size is set to 2 nm. The molar volume of the condensed phase, V_o, is 0.0255 kmol m^{-3}. Therefore, the initial number of monomers per particle is

$$\text{MU}_{init} = \frac{A_v}{V_o}\left(\frac{4\pi}{3}(1.0 \times 10^{-9}\ \text{m})^3\right) = 98.88\,.$$

Therefore, the initial number density of monomer units in the particle phase is

$$Q_T^o = 9.89 \times 10^{-6}\ \frac{\text{kmol}}{\text{m}^3}$$

Putting all the numbers together, a value of κ can be calculated

$$\begin{aligned}\kappa &= kC_2N_T4\pi\left(\frac{3V_oA_v^{-1}}{4\pi}\right)^{2/3}\\ &= (0.182)(1.097 \times 10^{-2})(6.022 \times 10^{19})4\pi\left(\frac{3(0.0255)}{4\pi(6.022 \times 10^{26})}\right)^{\frac{2}{3}}\\ &= 0.0663\ \frac{\text{kmol}}{\text{s m}^3}\end{aligned}$$

In Figure 10.1, the cube root of Eqn. (10.8) is plotted against the cube root of the Q_T prediction.

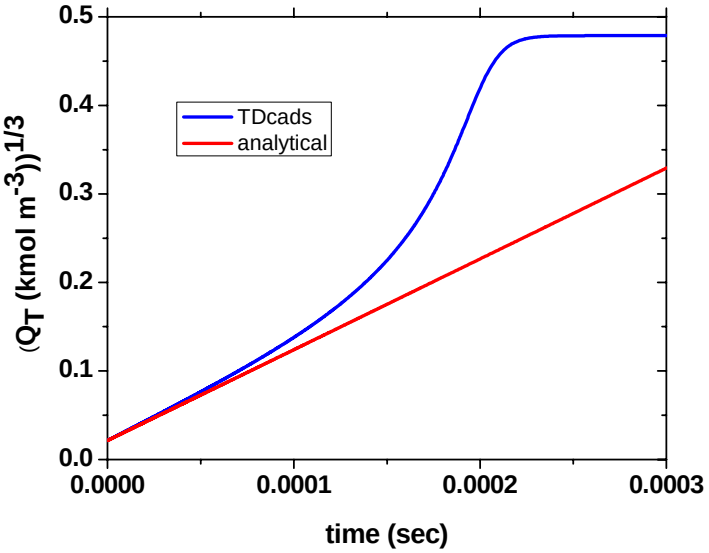

Figure 10.1 `TDcads` results vs. small-time Analytical solution which assumes no gas contraction.

from `TDcads`. A straight line is to be expected at small time. Good agreement is obtained at small times for the absolute value of $Q_T^{1/3}$ and its slope with respect to time. At later times the analyti-

```
START BLOCK TDCADS MODEL DEFINITION

   Include Gas Species in Solution Vector = yes
   Include Temperature in Solution Vector = no
   Boundary Conditions = psr
   Volume of Reactor = 10.0E-6 m^3
   Special Gas Species = N2
   START BLOCK TDCADS INFLOW CONDITIONS
       Inflow Temperature = 1000.
       Inflow Pressure = 1.0 atm
       Inflow Molar Flow Rate = 1.0
       START BLOCK INFLOW GAS MOLE FRACTIONS
            N2 = 0.10
            SPEC2 = 0.90
       END BLOCK INFLOW GAS MOLE FRACTIONS
       START BLOCK PARTICLE INLET CONDITIONS
            START BLOCK PARTICLE MONOMER DISTRIBUTION
                 Monomer Size = 2 nm
                 Monomer Concentration = 1.0E-10 gmol/cm3
                 Start Block MU Mole Fraction
                      condensesd-new = 1.0
                 End Block MU Mole Fraction
            END BLOCK PARTICLE MONOMER DISTRIBUTION
       END BLOCK PARTICLE INLET CONDITIONS
   END BLOCK TDCADS INFLOW CONDITIONS
END BLOCK TDCADS MODEL DEFINITION
```

Figure 10.2 `TDcads` Model definition block for the PSR Sample Problem

cal model fails to account for the change in volume of the reactor and, therefore, the increase in particle density. `TDcads` eventually reaches a steady state where all of the condensible material has migrated to the particle phase. At that point, since the original mole fraction of condensible gas was 0.9, the volume of the reactor is 1/10 of its original value.

10.2 PSR Implementation of the Growth of a Monodispersed Aerosol

Taking the first problem as a starting case, Let's add an inflow, an outflow, and a reactor volume to the problem to get a PSR. The `TDcads` Model Definition Block is replicated from the input file in Figure 10.2. The volume is set to 10 cm^3, the pressure is set to 1 atm, while the inlet flow rate is set to 1 $m^3\ s^{-1}$, i.e., 10^6 cm^3 /sec. Therefore, the residence time is 1.0×10^{-5} sec.

Figure 10.3 contains the resulting size distributions that flow out of the reactor. q is the concentration distribution of mononer units as a function of the number of monomer units in the particle, and has units of (kmol MU) m^{-3} MU^{-1}. In agreement with expectation, q decreases in a geometric progression as the number of monomer units increases. The vertical drops in the distribution are due to the intersectional boundaries and the discontinuous Galerkin formulation. The slight change of slope at very low particle concentrations at the highest particles numbers is due to a change in the algorithm that kicks in for low particle numbers, which was previously discussed in

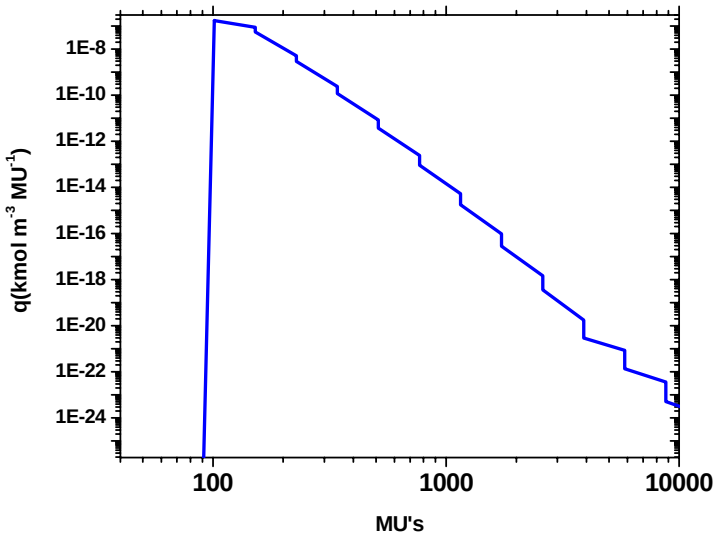

Figure 10.3 TMU concentration distribution function exiting the PSR

Section 2.13. The change in algorithm helps to preserve the nonnegativity of the sectional basis function coefficients.

11.Index of Symbols

i (second is j)	Designation for the index of the section number. These usually appear as the first subscript.
j	Reaction index
k (second is l)	Designation for the index of the monomer unit type. The second index to be used is the letter, l. These usually appear as the second subscript, when there is also a subscript for the section number.
L	Index of the target section.
n	Time step number. Appears as the last superscript.
N^{sect}	Number of sections in the current discretization of the μ coordinate.
$N^{\text{MU\_types}}$	Number of monomer unit types

Particle Concentrations and Related Quantities

b_i^L	Coefficient for the lower basis function in section i. This is one of the independent variables in the particle solution vector. (note, $q_i(\mu, t) = \Theta_i^L(\mu) b_i^L(t) + \Theta_i^H(\mu) b_i^H(t)$)
b_i^H	Coefficient for the higher basis function in section i. This is one of the independent variables in the particle solution vector. (note, $q_i(\mu, t) = \Theta_i^L(\mu) b_i^L(t) + \Theta_i^H(\mu) b_i^H(t)$)
$n(\mu, t) = \frac{q(\mu)}{\mu}$	The concentration of particles (kmol particles m^{-3}) having monomer unit size μ at time, t
N_T	The total concentration of particles in the particle phase. This has units of (kmol particles) m^{-3}.
N_i	The total concentration of particles in section i. This has units of (kmol particles) m^{-3}.
$q(\mu, t)$	Distribution function for the concentration of monomer units in particles with monomer size μ at time t. The units are (kmol m^{-3}) MU^{-1}.
Q_T	The total concentration of monomer units in the particle phase. This has units of (kmol MU) m^{-3}.

Q_i	The total concentration of monomer units in section i in the particle phase. This has units of (kmol MU) m^{-3}
Q_i^L , Q_i^H	The total concentration of monomer units in section i attributable to the lower and higher degree of freedom for section *i*, respectively. These quantities have units of (kmol MU) m^{-3}
$X_{i,k} = X_{i,k}^{\text{part}}$	The mole fraction of monomer unit type *k* in the particles of section *i*. *k* is assumed to be evenly distributed within the section *i* (i.e., $X_{i,k} = Z_{i,k} / Q_i$).
$z_k(\mu, t)$	Distribution function for the concentration of monomer units of type *k* in particles with monomer size μ. The units are (kmol MUK m^{-3}) MU^{-1}. Note this is related to $q(\mu, t)$ by the multiplicative constant, $X_{i,k}$.
$Z_{T,k}$	The total concentration of monomer units of monomer unit type k in the particle phase. This has units of (kmol MU) m^{-3}.
$Z_{i,k}$	The total concentration of monomer units of monomer unit type k in section i in the particle phase. This has units of (kmol MU) m^{-3}. This is one of the independent variables in the particle solution vector, Ξ.
μ	Monomer unit number. This is the number of monomer units in a particle. Units are the number of monomer units.
$\Theta_i^L(\mu)$	Basis function for the lower mode of a 2 DOF description of the total monomer unit distribution within a section.
$\Theta_i^H(\mu)$	Basis function for the upper mode of a 2 DOF description of the total monomer unit distribution within a section.

Other Variables

$a(\mu, \boldsymbol{X}^{\text{part}})$	Surface area of a single particle, with total monomer units, μ, and particle monomer unit mole fractions, $\boldsymbol{X}^{\text{part}}$. Since this quantity refers to a single particle, the units are m^2.
$A_i(\boldsymbol{X}^{\text{part}})$	Total surface area of all particles in section *i*. This quantity is dependent upon the number of particles in the section and on the composition of the particle phase, $\boldsymbol{X}^{\text{part}}$. The units for this is m^2 of surface area per m^3 of gas phase medium.

A_v — Avogadro's number

c_i^L, c_i^H — Coefficient for the evaluation of N_i from the basis function coefficients, b_i^L and b_i^H, which are the independent variables in the problem.

$$(N_i = c_i^L b_i^L(t) + c_i^H b_i^H(t))$$

C_k^g — Gas phase concentration of species *k*. Units are kmol m^{-3}.

d_i^L, d_i^H — Coefficient for the evaluation of Q_i from the basis function coefficients, b_i^L and b_i^H, which are the independent variables in the problem.

$$(Q_i = Q_i^L + Q_i^H = d_i^L b_i^L(t) + d_i^H b_i^H(t)) \text{ or } d_i^L = \int_{\mu_i}^{\mu_{i+1}} \Theta^L(\mu) d\mu$$

D_m — The Brownian diffusivity of particle *m*. Units are $m^2\ s^{-1}$.

$\boldsymbol{F}$ — Vector force (diffusiophoresis, or net) on a particle.

$g(\mu, t)$ — The net growth rate of particles of monomer unit size μ at time *t*. Since this quantity is on a per particle basis, the units are number of monomer units per time.

$g_k(\mu)$ — The growth rate of monomer unit *k* on particles of monomer unit size μ at time *t*. Since this quantity is on a per particle basis, the units are number of monomer units of type *k* per time.

$G_i(t)$ — The net growth rate of all particles in section i. Since this quantity is on a concentration basis having been integrated over the particle distribution function, the units are (kmol MU) m^{-3}.

h_i^L, h_i^H — Coefficient for the evaluation of Q_i from the basis function coefficients, b_i^L and b_i^H, which are the independent variables in the problem.

$k_j^c(\mu)$ — The collisional rate constant for reaction j between a gas phase species and a particle of size μ. Has units of m^3 (kmol MU)$^{-1}$ s^{-1}.

M_α — Integral moment α of the particle distribution function, $n(\mu, t)$.

MU_i^L	Lower monomer unit per particle bound in section *i*. If all of the particles in section *i* were in the lower mode of the distribution function, then the monomer unit per particle value would be equal to this amount.
MU_i^H	Higher monomer unit per particle bound in section *i*. If all of the particles in section *i* were in the higher mode of the distribution function, then the monomer unit per particle value would be equal to this amount.
P^i	The average number of particles per monomer unit in section *i*.
$P_r(K)$	Set of piecewise polynomials of degree r defined on the specific finite element domain *K*.
$\boldsymbol{S_i}$	Source term vector for the particles unknowns for section *i*. Units are (kmol) m^{-3} s^{-1}.
S_i^{part}	Source term for particles in section *i*. Units are (kmol particles) m^{-3} s^{-1}.
S_i^{TMU}	Source term for the total monomer units for section *i*. Units are (kmol MU) m^{-3} s^{-1}.
$S_{i,k}$	Source term for the total monomer units of monomer unit *k* for section *i*. Units are (kmol MU) m^{-3} s^{-1}.
$S_{i,k}^{bulk}$	Source term for the total monomer units of monomer unit type *k* for section *i*, due to the solid phase particle kinetics operator only. Units are (kmol MU) m^{-3} s^{-1}.
$\boldsymbol{S}_i^{bulk}$	Source term for the particle conservation equations in section *i* due to the bulk phase particle kinetics operator. This is a vector, with each term having units of kmol m^{-3} s^{-1}. For each section for a DOF 2 implementation, with N^{MU_types} different monomer unit types, there are $N^{MU\_types} + 1$ conservation equations. The first equation is the particle conservation equation. The second equation is the total monomer unit conservation equation, and the last $N^{MU\_types} - 1$ equations are the individual monomer unit conservation equations excluding the zeroth monomer unit.
$\boldsymbol{S}_i^{coag}$	Source term vector for the particle conservation equations in section *i* due to the coagulation operator. Each term has units of kmol m^{-3} s^{-1}. For each section for a DOF 2 implementation, with N^{MU_types} different monomer unit types, there are $N^{MU\_types} + 1$ conservation equations. The first equation is the particle conservation equation. The second

	equation is the total monomer unit conservation equation, and the last $N^{\text{MU\_types}} - 1$ equations are the individual monomer unit conservation equations excluding the zeroth monomer unit.
$\boldsymbol{S}^{\text{coag}}_{B+A\rightarrow L}$	Each $\boldsymbol{S}^{\text{coag}}_{B+A\rightarrow L}$ term represents the source term due to one interaction, where a particle in section *B* collides and adds to a particle in section *A* creating one new particle mostly in section *L*. It may for stability and/or conservation purposes also add particles to an adjacent section to *L*, section *Lalt*.
$S^{\text{coag, part}}_{B+A\rightarrow L}$	This is the particle source term component of the expression, $\boldsymbol{S}^{\text{coag}}_{B+A\rightarrow L}$. There are entries in section B, A, and L.
$S^{\text{coag, TMU}}_{B+A\rightarrow L}$	This is the total monomer unit source term component of the expression, $\boldsymbol{S}^{\text{coag}}_{B+A\rightarrow L}$
SA_LB_L	Lowest MU number of the higher section in a pair which can collide and produce a particle in section *L*.
SA_UB_L	Highest MU number of the higher section in a pair which can collide and produce a particle in section *L*.
$SB\_LB_{L,\,A}$	Lowest MU number of the lowest section in a pair which can collide and produce a particle in section *L*, when the higher pair is in section *A*.
$SB\_UB_{L,\,A}$	Highest MU number of the lowest section in a pair which can collide and produce a particle in section *L*, when the higher pair is in section *A*.
$\boldsymbol{S}^{\text{nucl}}(b_i^L, b_i^H, Z_{i,\,m}, C_k^g)$	Complete source term for the nucleation operator.
$\boldsymbol{S}_i^{\text{nucl\_part}}$	Source term for the particle unknowns due to the nucleation operator.
$\boldsymbol{S}_i^{\text{nucl\_gas}}$	Source term for the gas phase concentrations due to the nucleation operator.
$T_h(\mu)$	Set of finite elements defined on the monomer unit coordinate, μ, whose size is bounded in all dimensions by the size, *h*.
v_k	Velocity of gas phase species *k* (m s^{-1}).
V or $v(\mu)$	Volume or volume of a particle with μ monomer units in it. Units are in m^3. Small caps are used for "per particle" quantities.

$\bar{V}_k$	Partial molar volume of species *k* or monomer unit type *k* in the particle phase. Units are in m^3 $kmol^{-1}$.
$\bar{V}(\mu)$	Molar volume of a particle with monomer size μ. Units are in m^3 $kmol^{-1}$.
W_h	Finite element space with characteristic sized elements *h*.
$\beta(\mu_i, \mu_j)$	Collision frequency due to Brownian motion between particles with monomer unit numbers μ_i and μ_j, in units of m^3/kmol/sec.
$S_{\mathrm{A,\,B}}^{L,\,p\_\mathrm{loss}}$	Collisional particle loss rate from section *A* and *B*, each, for the coagulation of a particle from the lower section *B* with the higher section *A*, which produces a new particle in section *L*.
$S_{\mathrm{A,\,B}}^{L,\,A\_\mathrm{loss}}$	Collisional section *A* monomer loss rate from the coagulation of a particle from the lower section *B* with the higher section *A*, which produces a new particle in section *L*.
$S_{\mathrm{A,\,B}}^{L,\,B\_\mathrm{loss}}$	Collisional section *B* monomer loss rate from the coagulation of a particle from the lower section *B* with the higher section *A*, which produces a new particle in section *L*.
$S_{\mathrm{A,\,B}}^{L,\,Lgain}$	Collisional section *L* monomer creation rate from the coagulation of a particle from the lower section *B* with the higher section *A*, which produces a new particle in section *L*.
ω_m	Particle mean thermal velocity for particle $m = \left(\frac{8kT}{\pi M_m}\right)^{\frac{1}{2}}$
$\Xi[b_i^L, b_i^H, Z_{i,\,k\neq 0}]$	Particle solution vector. This is a list of the independent unknowns in the particle package that completely determines the state of the particle distribution, when combined with the independent unknowns in the surrounding, medium.

12.References

[1] D. G. Goodwin, “An Open-Source, Extensible Software Suite for CVD Process Simulation,” Chemical Vapor Deposition XVI and EUROCVD 14, ECS Proceedings Volume 2003-08, M. Allendorf, F. Maury, and F. Teyssandier, editors, The Electrochemical Society, pp. 155-162 (2003); http://www.cantera.org; http://sourceforge.net/projects/cantera

[2] S. K. Friedlander, *Smoke, Dust, and Haze*, Oxford University Press, N. Y. 2000, Chap. 10.

[3] F. Gelbard, J. H. Seinfeld, “The general dynamic equation for aerosols. Theory and application to aerosol formation and growth,” *J. Coll. Int. Sci.*, **63**, 472-479 (1978).

[4] F. Gelbard, J. H. Seinfeld, “Simulation of Multicomponent Aerosol Dynamics,” *J. Coll. Int. Sci.,* **78**, 485-501 (1980).

[5] F. Gelbard, *MAEROS User Manual*, Sandia Report, SAND80-0822, Sandia National Laboratories, Albuquerque, NM (1980).

[6] C. J. Pope, J. B. Howard, Simultaneous particle and molecule modeling (SPAMM): An approach for combining sectional aerosol equations and elementary gas-phase reactions,” *Aerosol Sci. and Tech.*, **27**, 73-94 (1997).

[7] M. Jacobson, “Development and Application of a new air pollution modeling system. Part II: Aerosol module structure and design,” *Atmos. Environ.*, **31A,** 131-144 (1997).

[8] S. Kumar, D. Ramkrishna, “On The Solution of Population Balance Equations By Discretization - II A Moving Pivot Technique,” *Chem. Eng. Sci.*, **51**, 1333-1342 (1996).

[9] J. J. Wu, R. C. Flagan, “Discrete-Sectional Solution to the Aerosol Dynamics Equation, *J. Coll. Int. Sci*, **123**, 339-352 (1988).

[10] A. Violi, A. D’Anna, A. D’Alessio, “Modeling of particulate formation in combustion and pyrolysis,” *Chemical Eng. Science*, **54,** 3422-3442 (1999).

[11] S. E. Stein, A. Fahr, “High-Temperature Stabilities of Hydrocarbons,” *J. Phys. Chem*., **89**, 3714 - 3725 (1985).

[12] D. J. Rader, A. S. Geller, *Particle Transport in Parallel-Plate Reactors*, Sandia Report, SAND99-1539, Sandia National Laboratories, Albuquerque, NM (1999).

[13] M. Frenklach, S. J. Harris, “Aerosol Dynamics Modeling Using the Method of Moments,” *J. Colloid and Interface Science*, **118**, 252-261 (1987).

[14] J. D. Landgrebe, S. E. Pratsinis, “A Discrete-Sectional Model for Particulate Production by Gas-Phase Chemical Reaction and Aerosol Coagulation in the Free-Molecular Regime,” J. of Colloid and Interface Science, 139, 63 - 85 (1990).

[15] S. Kumar, D. Ramkrishna, “On the solution of population balance equations by discretization - III. Nucleation, growth and aggregation of particles, *Chemical Engineering Science,* **52,** 4659 - 4679 (1997).

[16] H. K. Moffat, “Obtaining an Accurate Jacobian within CADS”, Personal Notes, Sandia National Laboratories, Albuquerque, NM, 2004.

[17] L. Waldmann, *Z. Naturf.*, **14a**, 589 (1959).

[18] L. R. Petzold, “A Description of DASSL, A differential/algebraic system solver,” SAND82-8637, Sandia Laboratories, Livermore, CA (1992).

[19] P. N. Brown, G. D. Byrne, A. C. Hindmarsh, “VODE: A Variable-coefficient ODE Solver,” *SIAM J. Sci. Stat. Comput.*, **10**, 1038-1051 (1989).

[20] H. K. Moffat, “Using Cubic Ramps with CADS: Design choices and convergence Problems,” Sandia Memo, Sandia National Laboratories, Albuquerque, NM. March 22, 2005.

[21] P. Lasaint, P. A. Raviart, “On a finite element method for solving the neutron transport equation,” in C. de Boor, (ed.), *Mathematical Aspects of Finite Elements in Partial Differential Equations*, Academic Press, N. Y. 1974, pp. 89 - 123.

[22] K. S. Bey, A. Patra, J. T. Oden, “hp-Version Discontinuous Galerkin Methods for Hyperbolic Conservation Laws: A Parallel Adaptive Strategy”, *Int. J. Num. Methods in Eng.*, **38,** 3889 - 3908 (1995).

[23] S,-Y. Lin, Y.-S. Chin, “Discontinuous Galerkin Finite Element Method for Euler and Navier-Stokes Equations,” *AIAA J.*, **31**, 2016 - 2026 (1993).

[24] D. Polluter, A. Zaki, A. Fortin, “Adaptive Remeshing for Hyperbolic Transport Problems,” *Comp. Fluid Dyn.*, **3**, 79 - 99 (1994).

[25] S. Kumar, D. Ramkrishna, “On the Solution of Population Balance Equations by Discretization - I. A fixed Pivot Technique,” *Chemical Engineering Science,* **51,** 1311-1332 (1996).

[26] “Case Studies for Particle Modeling aimed at Verification,” Personal Communication, H. K. Moffat, J. E. Brockmann (2003).

[27] F. Gelbard, “Modeling Multicomponent Aerosol Particle Growth by Vapor Condensation,” *Aerosol Sci. and Tech.*, **12**, 399-412 (1990).

[28] M. M. R. Williams, *J. Colloid Interface Sci.*, **93**, 252-263 (1983).

[29] S. Vemury and S. E. Pratsinis, “Self Preserving Size Distributions of Agglomerates,” *J. Aerosol Sci.*, **26**, no. 2, 175-185 (1995) and “Erratum,” *J. Aerosol Sci.*, **26**, no 4, 701 (1995).

[30] R. A. Dobbins, R. A. Fletcher, H. C. Chang, "The Evolution of Soot Precursor Particles in a Diffusion Flame," *Combustion and Flame*, **115**, 258-298 (1998).

[31] N. A. Fuchs, *The Mechanics of Aerosols*, Pergamon, New York, 1964.

[32] M. M. R. Williams, S. K. Loyalka, *Aerosol Science Theory and Practice*, Pergamon Press, New York, 1991.

[33] *Numerical Recipes in C*, Eds. W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, Cambridge Univ. Press, Cambridge, UK (1992).

[34] W. R. Smith, R. W. Missen, *Chemical Reaction Equilibrium Analysis: Theory and Algorithms*, J. Wiley New York (1982).

[35] N. K. Wu, S. K. Friedlander, "Enhanced power law agglomerate growth in the free molecular regime," *J. Aerosol. Sci.,* **24**, 273-282 (1993).

[36] M. Frenklach, J. Wang, "Detailed Mechanism and Modeling of Soot Particle Formation," in *Soot Formation in Combustion: Mechanisms and Models*, p. 165 - 192, J. Bockhorn Ed. Springer Verlag, Heidelberg (1994).

[37] S. W. Benson, *Thermochemical Kinetics*, J. Wiley & Sons, New York, 1976.

[38] D. W. Goodwin, "Defining Phases and Interfaces, "Cantera's web site: (http://sourceforge.net/projects/cantera), Division of Engineering and Applied Science, California Institute of Technology, Aug. 2003.

[39] Salinger, A., K. Devine, et al. (1996). "MPSalsa: A Finite Element Computer Program for Reacting Flow Problems. Part 2 - User's Guide." SAND96-2331, Sandia National Labs, September 1996.

[40] Kelley, C. T. and D. E. Keyes (1998). "Convergence Analysis of Pseudo-Transient Continuation" *SIAM J. Numer. Anal.*, **35**(2): 508-523.

[41] M. Frenklach, H. Wang, "Detailed Modeling of Soot Particle Nucleation and Growth," in Twenty-third symposium (International) on Combustion, The Combustion Institute, Pittsburgh 1991, p. 1559.

[42] Harry Moffat, `cttables`, Utility program for `cantera`, that prints out gas phase species thermochemistry, transport properties, and reactions rates in table form, available on http://sourceforge.sandia.gov.

[43] H. K Moffat, "Implementation of a Pitzer Model for Strong Electrolyte Thermochemistry within Cantera," Sandia Memo, Sandia National Laboratories, Albuquerque, NM, June 12, 2006.

[44] M. E. Colket, R. J. Hall, "Success and Uncertainties in Modeling Soot Formation in Laminar, Premixed Flames," in *Soot Formation in Combustion*, Henning Bockhorn Ed.,

Springer-Verlag, New York, 1994.

[45] http://webbook.nist.gov/chemistry/

[46] L. Waldmann, K. H. Schmitt, “Thermophoresis and Diffusiophoresis of Aerosols,” in *Aerosol Science*, Ed. C. N. Davies, Academic Press, New York, 1966.

[47] L. Talbot, R. K. Cheng, R. W. Schefer, D. R. Willis, “Thermophoresis of particles in a heated boundary layer,” *J. Fluid Mech.*, **101**, 737-758 (1980).

[48] J. R. Brock, *J. Colloid Sci.*, **17**, 768 (1962).

[49] R. J. Kee, M. E. Coltrin, P. Glarborg, *Chemically Reacting Flow: Theory and Practice*, J. Wiley, New York, 2003.

[50] M. E. Coltrin, R. J. Kee, G. H. Evans, E. Meeks, F. M. Rupley, J. Grcar, *SPIN (Version 3.83): A Fortran program for modeling one-dimensional rotating-disk/stagnation-flow chemical vapor deposition reactors*, Sandia Report, SAND91-8003, Sandia National Laboratories, Livermore, CA. (1991)

[51] R. Bird, W. Stewart, E. Lightfoot, *Transport Phenomena*, 2nd Ed., J. Wiley & Sons, New York (2002).

[52] S. K. Aliabadi, T. E. Tezduyar, “Space-time finite element computation of compressible flows involving moving boundaries and interfaces”, *Computer Methods in Applied Mechanics and Engineering*, 107, 209-223 (1993).

Distribution

1	MS 0382	Domino, S. P., 1541
1	MS 0826	Rader, D., 1513
1	MS 0836	Brockmann, J. E., 1517
1	MS 0826	Geller, A. S., 1513
1	MS 0824	Prairie, M. R. 1510
1	MS 0836	Lash, J. S., 1514
10	MS 0836	Moffat, H. K., 1514
1	MS 1086	Coltrin, M. E.1126
1	MS 1086	Creighton, J. R., 1126
1	MS 1135	Hewson, J. C., 1532
1	MS 1135	Tieszen, S. R., 1532
1	MS 9051	Lignell, D., 8351
2	MS 9018	Central Technical Files, 8944
2	MS 0899	Technical Library, 4536

Prof. D. G. Goodwin
California Institute of Technology
Department of Engineering
1200 E California Blvd. 104-44
Pasadena, CA 91125

Prof. Phil Smith
3290 Merrill Engineering Building
50 South Central Campus Drive
Dept. of Chemical Engineering
University of Utah
Salt Lake City, Utah, 84112

Prof. Adel Sarofim
495 East 100 South
109 Kennecott Research Building
University of Utah
Salt Lake City, Utah 84112-1114

Zhiwhe Yang
3290 Merrill Engineering Building
50 South Central Campus Drive
Dept. of Chemical Engineering
University of Utah
Salt Lake City, Utah, 84112

Prof. J. Sutherland
50 South Central Campus Drive, MEB, Rm. 3290
Dept. of Chemical Engineering
University of Utah
Salt Lake City, Utah, 84112