To the Graduate Council:
I am submitting herewith a dissertation written by Elisha Don Feger entitled "Evaluating Explicit Methods for Solving Astrophysical Nuclear Reaction Networks." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Physics.

Michael Guidry, Major Professor

We have read this dissertation
and recommend its acceptance:

Vasilios Alexiades

Christian Cardall

W. Raphael Hix

Kate Jones

Accepted for the Council:

Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

Evaluating Explicit Methods for Solving Astrophysical Nuclear Reaction Networks

A Dissertation

Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Elisha Don Feger

August 2011

This dissertation is dedicated to my parents.

Acknowledgements

I want to thank my parents for always being there when I needed them, whether it was for advice or to get a ride to some place I needed to be. They have always believed in me and have been supportive of my goals.

I want to thank my major professor Dr. Michael Guidry for always being available when I had a question or concern. This work would not have been possible without his continued tutelage and guidance. Dr. Raph Hix also played a major role in helping to guide my research and work, and answering my questions no matter how rudimentary they might have been. Discussions with both of them have always been insightful and fruitful.

Dr. Bronson Messer's advice was invaluable for understanding FLASH and for understanding the general process of running large parallel codes on the supercomputers out at Oak Ridge National Lab. I consider him to be an honorary member of my doctoral committee. Questing for the Grail continues.

Speaking of FLASH, my good friend Dr. Viktor Chupryna and I worked hand-in-hand to port the asymptotic approximation solver that myself, Jay Billings, and Reuben Budiardja have developed over the years into a module for FLASH. I'd like to thank Chris Smith and Dr. Suzanne Parete-Koon for being "FLASH buddies" and sharing in the pain and suffering inflicted by that code; that's the sort of thing only a friend would willingly do. Jay, while he was still here at the University of Tennessee-Knoxville, and I had many good discussions, not always on topic but always interesting nonetheless.

I'd like to thank my remaining committee members for volunteering their time and interest by being on it. Thank you, Dr. Vasilios Alexiades, Dr. Christian Cardall, and Dr. Kate Jones.

Finally, I'd like to acknowledge the contribution El Guapo has made to my dissertation, because everyone has an El Guapo, and mine just happens to be the real El Guapo.

The popular stereotype of the researcher is that of a skeptic and a pessimist. Nothing could be further from the truth! Scientists must be optimists at heart, in order to block out the incessant chorus of those who say "It cannot be done."

-*Prokhor Zakharov,* Alpha Centauri

Abstract

Many systems of physical interest are difficult to manage computationally because of the intrinsic nature of the equations that govern them. Many of these systems of equations are stiff, meaning that the standard approach to solving them is with implicit methods, because explicit methods either are unstable or require timesteps too small to be computationally efficient. Presented here is a study of explicit methods that decouple stability from accuracy under certain conditions, allowing for larger timesteps to be taken.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Large systems of differential equations are frequently encountered in physical problems of interest, ranging from populations of isotopes in astrophysical simulations to geochemical models. Treating such large systems of equations with Runge-Kutta, or other explicit forward difference algorithms, is a standard and simple way to begin to attack them. Unfortunately many physical systems are stiff; that is to say, the systems have some components that evolve much more rapidly than others.

For stiff systems, numerical stability typically requires that the maximum timestep be limited by the quickly-evolving components. This wastes processor cycles as the program loops over the network and performs calculations on the slowly evolving equations as often as it does on the rapidly evolving equations. This processing overhead makes such a method of solution undesirable, and thus it is standard to use implicit techniques, which allow for larger timesteps and are more stable, to solve such systems. Even taking a moderate timestep in a stiff system with an explicit method allows small errors to creep into the network and accumulate over time, ruining accuracy.

Implicit techniques bring their own challenges even as they solve some of the problems associated with using explicit methods on stiff systems. Implicit techniques are by their nature typically stable, but they are numerically intensive, requiring

many more operations for each equation that is solved in every timestep for large networks. Sometimes semi-implicit methods are used to solve stiff systems; these attempt to solve the implicit method's equations by linearizing them, and provide a compromise between the simplicity of explicit methods and the stability of implicit methods [1].

Stiff systems of equations have been encountered, both in nature and in textbook examples, for many years [2]. Solutions to them have been studied extensively, and many methods for treating them have been devised [3][4][5][6]. This study investigates the feasibility of a new composite explicit technique to solve stiff systems, incorporating multiple ideas including a flux-limiting forward differencing (FLFD) approach, an asymptotic approximation, and a partial-equilibrium treatment.

This study is organized as follows. Chapter 1 presents a general introduction to the contents of this dissertation. Chapter 2 discusses stiffness and why it makes numerical simulations more difficult. Chapter 3 develops and presents new explicit methods. Chapter 4 discusses the test problems the FLFD combined with asymptotic method has been applied to and gives comparisons against implicit and semi-implicit techniques. The final chapter summarizes the results from this study, discusses the use of these new explicit methods in other calculations, lists known remaining issues and gives a brief discussion of possible future work and opportunities for continuing success.

Chapter 2

Stiff Systems of Equations and Instability

Boyce and DiPrima give a simple and intuitive definition for stiffness [7]: "Problems for which a much smaller step size is needed for stability than for accuracy are called stiff." Press, et. al., say that "Stiffness occurs in a problem where there are two or more very different scales of the independent variable on which the dependent variables are changing [1]." Nagle, Saff, and Snider give the following statement about stiff equations: "Differential equations that involve one or more fast decaying processes in time, with time constants that are short compared to the time span of interest, are called stiff problems [8]."

Figure 2.1 shows the direction field for a typical stiff equation, the stiffness of the problem manifests itself visually in the rapid change from shallow slopes to extremely steep slopes along the y-axis for any given x [3]. The slopes near the curve G, which is the solution to be calculated in this case, are relatively small, however values of y even moderately larger or smaller than the value at G for a given x have radically larger slopes. One can say that the slopes vary quickly near the curve G so that a small error in calculation can result in a value of y which has a very different slope than the curve G.

This mental picture is important because as calculation proceeds, small errors, whether from truncation or from the approximation scheme being used, will creep into the calculation. This shifts the solution curve being followed away from the desired curve [9]. If the slope at the point where the solution has been shifted is too steep, the calculation will typically destabilize and run off to infinity. This behavior can be seen in Figure 2.1 from the way the derivatives dy/dx quickly become almost vertical. This property of a system being extremely sensitive to small errors is an inherent side-effect of the stiffness.

Taking large steps in the independent variable (typically steps in time, hence the term timesteps) is desirable for computational efficiency, but for the reasons outlined above doing so is impossible in stiff sets of equations, at least with explicit methods. To illustrate this point, two stiff systems are examined below.

2.1 Example: Stiff Equation

In their book, Boyce and DiPrima give an example problem of a single equation that is stiff [7], which we reproduce here. Consider the initial value problem

$$y(t)' = -100y + 100t + 1 \qquad y(0) = 1. \tag{2.1}$$

This is a relatively simple linear differential equation and the solution to the equation with the initial value given is

$$y = e^{-100t} + t. \tag{2.2}$$

This solution is plotted in Figure 2.2. At $t = 0$ it is clear that the exponential term dominates the solution, but even a small distance in the positive direction away from 0 the exponential term drops off quickly and the linear term dominates the solution. Figure 2.3 displays numerical solutions to Eqn. (2.1).

For a timestep h of 0.0166 the solution starts out wildly oscillatory, but is stable in the long term, approaching the analytic answer fairly well by $t = 0.2$ and virtually

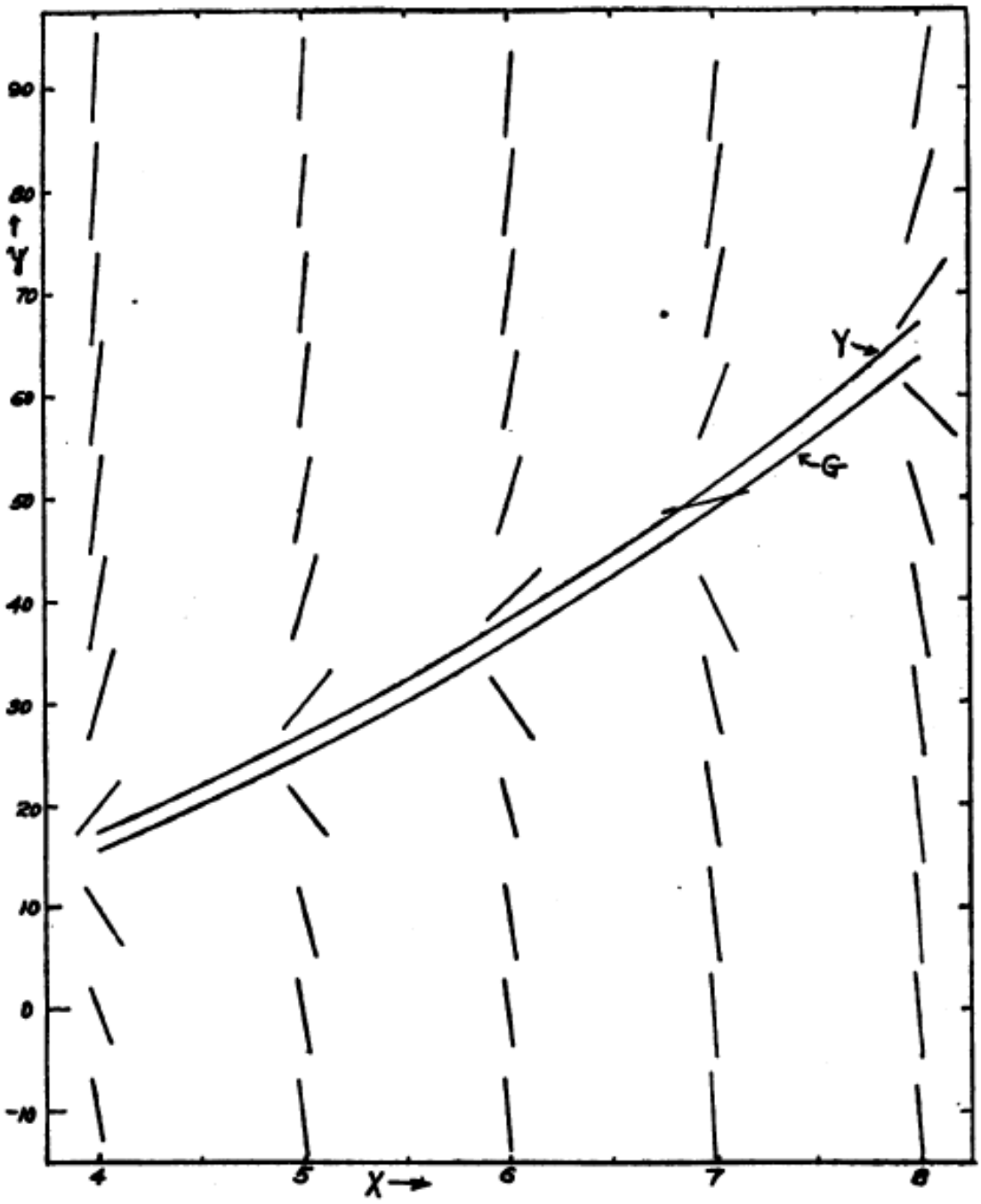


Figure 2.1: Plot showing the slopes dy/dx (the direction field) for a typical stiff equation, $dy/dx = 5(y - x^2)$. Plot taken from [3].

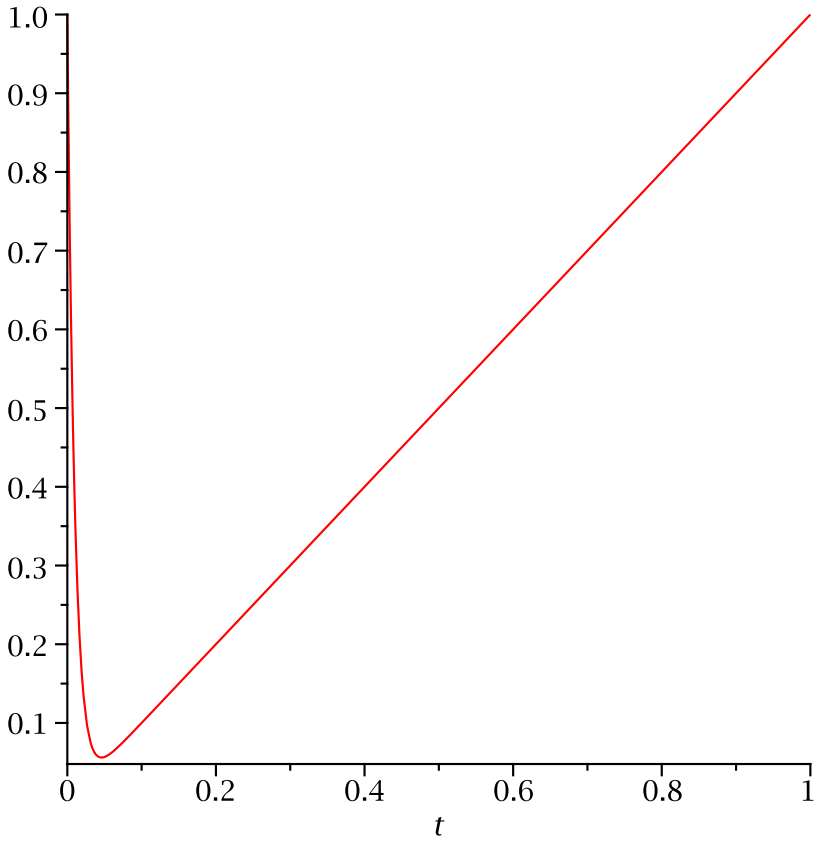


Figure 2.2: A plot of Eqn. (2.2). Note how the solution is dominated initially by the exponential decay, which quickly gives way to the linear term.

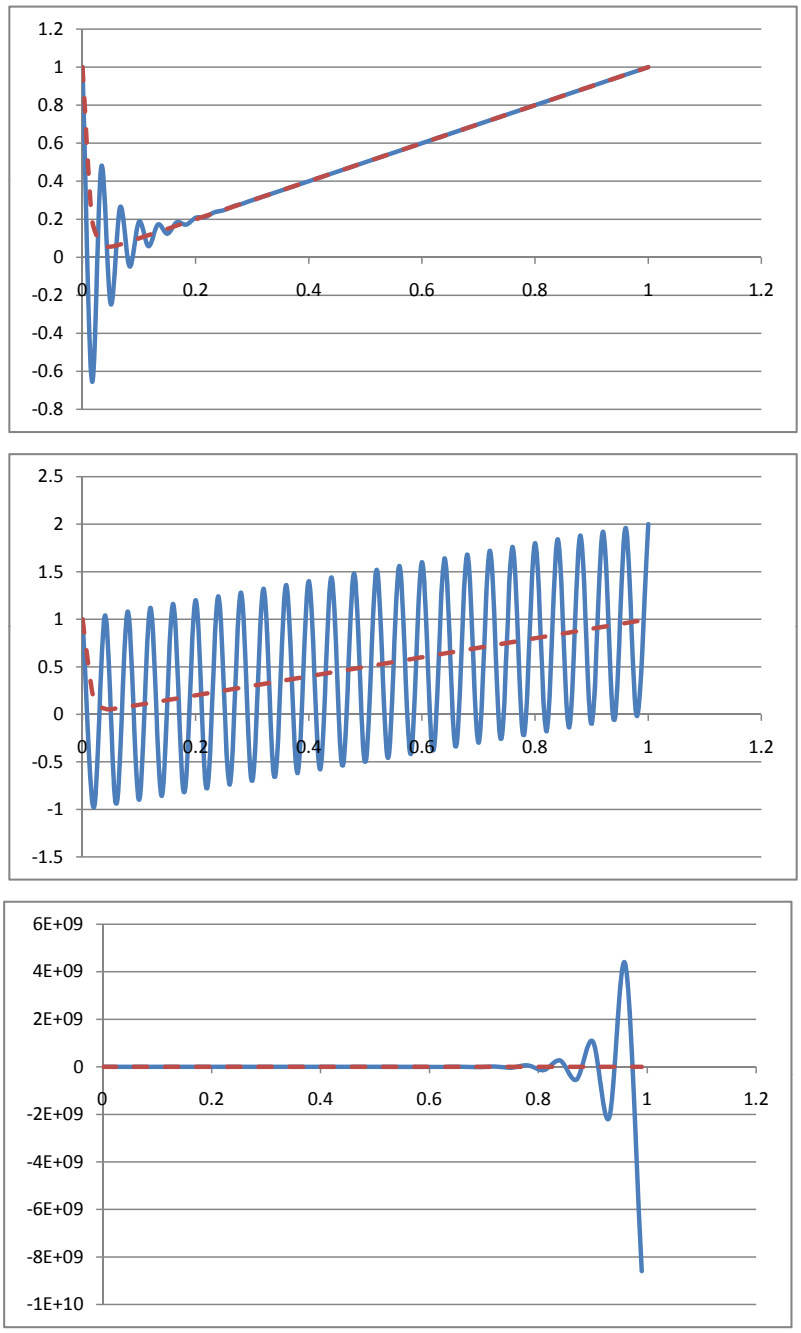


Figure 2.3: Numerical solutions to Eqn. (2.1) computed using Euler's method for three different timesteps h, from top to bottom: $h = 0.0166$, $h = 0.02$, and $h = 0.03$. The numerical solutions are the solid curves and the analytic solutions are the dashed curves.

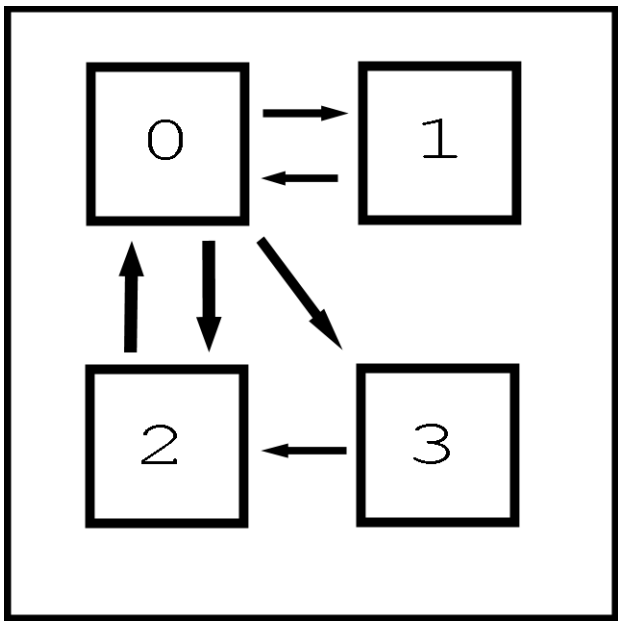


Figure 2.4: A simple 4-box model. Connections between the boxes that allow for flux between them are marked with arrows.

matching it in all the points after that time. When h is 0.02 the numerical solution tends to oscillate around the true solution - for $x = 0.98$ y_{analytic} is 0.98, y_{numeric} is -0.02, and for $x = 1.0$ $y_{\text{analytic}} = 1.0$ we see that y_{numeric} is 2.0. An h of 0.03 is highly unstable and quickly runs away to infinity. These behaviors emerge because for a very small region in t the rate of change in the system is extremely rapid, while outside of that region the rate of change can be well approximated with much larger timesteps. To maintain stability requires very small timesteps, while the behavior of the analytic solution suggests that to be accurate only a much looser and longer timestep is needed. Thus we say that this equation is stiff.

2.2 Example: 4-Box Network

The preceding example in Section 2.1 is of a single stiff equation. In order to illustrate the sorts of instabilities that arise in more complicated systems, Figure 2.4 is a simplistic model of four population boxes which has a number of connections among

Table 2.1: Rate constants for the flux calculations in the 4-box problem

Constant	Value
k_{01}	0.0020
k_{02}	0.0025
k_{03}	0.0250
k_{10}	0.5000
k_{20}	0.0500
k_{32}	0.0050

each other. This network as illustrated in Figure 2.4, is described by a coupled set of ordinary differential equations (ODEs)

$$\frac{dY_i}{dt} = \sum_j F_{ij} \tag{2.3}$$

where Y_i are the independent variables, t is the independent variable, and F_{ij} are the fluxes from box i to box j. In this 4-box example (with the boxes labeled as in Figure 2.4) our set of coupled ODEs becomes

$$\frac{dY_0}{dt} = F_{20} + F_{10} - F_{01} - F_{02} - F_{03} \tag{2.4}$$

$$\frac{dY_1}{dt} = F_{01} - F_{10} \tag{2.5}$$

$$\frac{dY_2}{dt} = F_{02} + F_{32} - F_{20} \tag{2.6}$$

$$\frac{dY_3}{dt} = F_{03} - F_{32} \tag{2.7}$$

where the fluxes F_{ij} are assumed for simplicity to have the form $F_{ij} = k_{ij}Y_i$, and k_{ij} is a constant rate for the transition from box i to j. The rate constants k_{ij} can be found in Table 2.1.

We will treat the problem with Euler's method, so the populations in the boxes N_i are given by $N_i + (F_{\text{incoming}} - F_{\text{outgoing}})\Delta t$ where F_{incoming} are the fluxes which go from other boxes into box i, F_{outgoing} are the fluxes which go from box i to other boxes, and Δt is the timestep (referred to as h previously).

Figure 2.5 shows numerical calculations for this 4-box network, using Euler's method and three different timestep sizes. The figure is a log-log plot of the time and the abundances of each population (making 0 on the y-axis equal to the total population of the system). One observes that at a timestep of 1 the calculation is stable. The calculation oscillates wildly initially for a larger timestep ($\Delta t = 3.9$) but eventually settles into essentially the same values as the smaller timestep calculation. Thus we lose early accuracy, but the system ultimately is stable over long timescales. For a timestep of 4 the system becomes completely unstable, oscillates, and rapidly runs off to infinity with the population of Box 1 growing the most quickly.

Euler's method can treat the problem for small enough step sizes with accuracy and stability, but because the problem is inherently stiff the method loses its viability rather quickly for larger stepsizes. First accuracy degrades, then stability soon follows. A technique that keeps the simplicity and low computational cost of Euler's method while also retaining accuracy and stability with larger timesteps would be ideal to solve this example problem. The asymptotic approximation and partial equilibrium method developed in the next chapter hold promise to be such a technique.

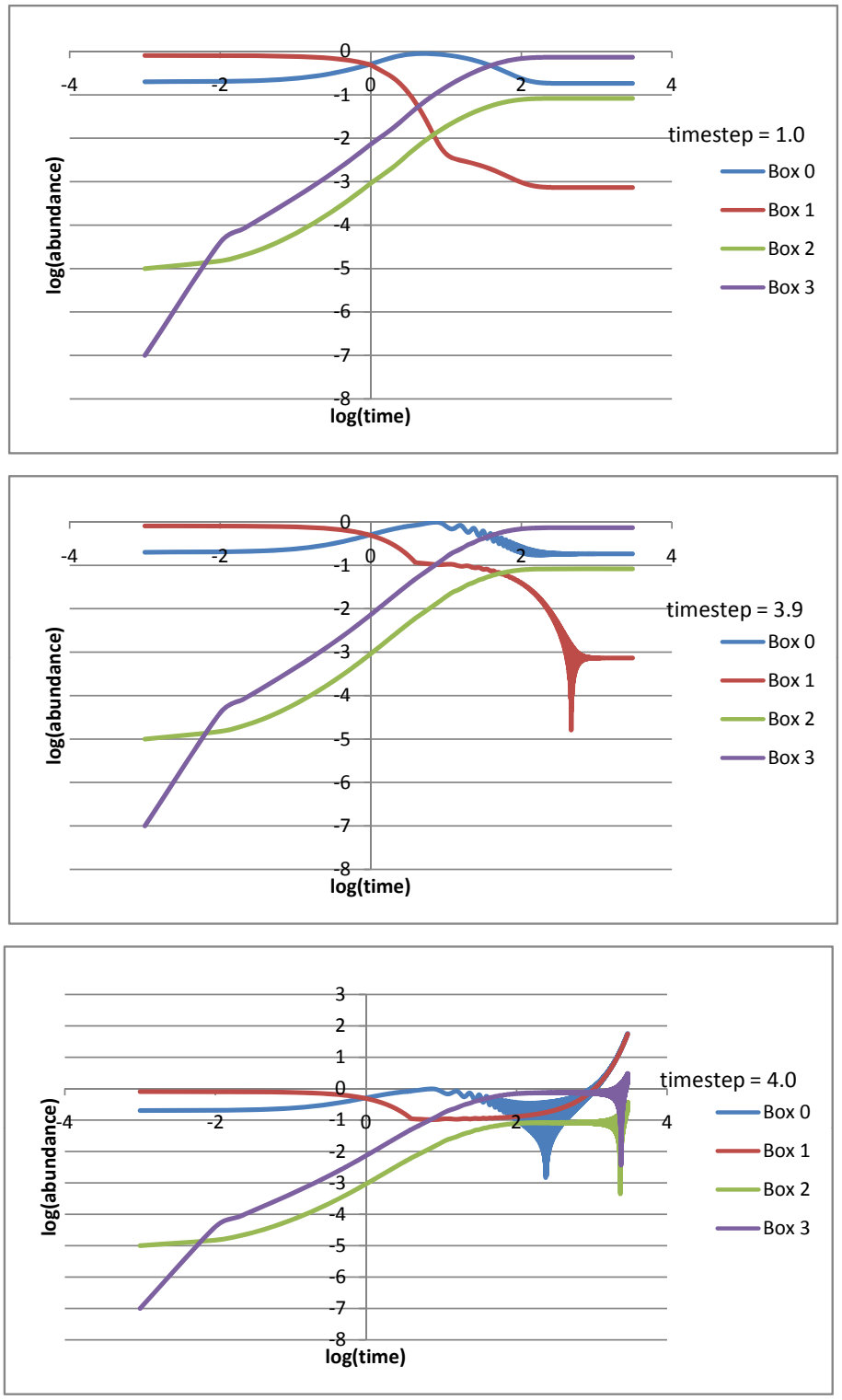


Figure 2.5: Plots of the 4-box network's populations made using Euler's method for varying timesteps, from top to bottom $h = 1.0$, $h = 3.9$, and $h = 4.0$.

2.3 Implicit and Semi-Implicit Methods for Treating Stiff Equations

Stiff equations are traditionally treated with implicit and semi-implicit techniques. Implicit techniques have the character of requiring that solving for the current timestep requires derivatives be evaluated at a future time. A semi-implicit technique is the result of solving an implicit method by linearization. Below are introduced the implicit or backward Euler method and a semi-implicit technique discovered by Bader and Deuflhard. These two techniques are the "gold standard" against which the explicit methods developed in Chapter 3 will be compared.

2.3.1 Implicit Euler

Backward Euler is an implicit differencing method and is stable under a wide variety of conditions [10]. A forward or explicit Euler timestep is of the form

$$y_{n+1} = y_n + h\frac{dy_n}{dt}, \tag{2.8}$$

where dy_n/dx is the derivative of the solution evaluated at the nth timestep, y_{n+1} is the solution evaluated at $(n+1)$th timestep, y_n is the solution evaluated at the nth timestep, and h is the timestep. Backward Euler is an implicit method, so finding out what the solution is at the next timestep depends on already knowing some information from that timestep. The Backward Euler timestep is of the form

$$y_{n+1} = y_n + h\frac{dy_{n+1}}{dt}. \tag{2.9}$$

In order to simplify the notation from here on, instead of using $\frac{dy}{dt}$ for the derivatives we will simply use y', and bold facing means that the symbol stands for a matrix.

For a linear system of ODEs with constant coefficients

$$\vec{y'} = -\mathbf{C} \cdot \vec{y}, \tag{2.10}$$

where $\vec{y'}$ is the vector containing the derivatives, $\vec{y}$ is the vector containing the dependent variables, and $\mathbf{C}$ is a positive definite matrix containing the rate coefficients. Treating such a system with backward Euler results in the following solution:

$$\vec{y}_{n+1} = (\mathbf{1} + \mathbf{C}h)^{-1}\vec{y}_n, \tag{2.11}$$

where $\vec{y}_{n+1}$ is the vector containing the solutions at the $(n+1)$th timestep, $\vec{y}_n$ is the vector containing the solutions at the nth timestep, and $\mathbf{1}$ is the unit matrix. Each timestep advanced this way requires the inversion of the matrix $\mathbf{1} + \mathbf{C}h$ which is a computational costly operation, however the method is very stable, even when treating stiff equations.

2.3.2 Bader-Deuflhard

The method named after Bader and Deuflhard is a semi-implicit scheme. Their original paper on the method is Ref. [11], and Timmes briefly introduces the method to the astrophysical audience in Ref. [12]. Bader-Deuflhard is derived by starting with the implicit form of the midpoint rule

$$\vec{y}_{n+1} - \vec{y}_{n-1} = 2h\vec{f}\left(\frac{\vec{y}_{n+1} + \vec{y}_{n-1}}{2}\right), \tag{2.12}$$

where

$$\vec{f}(\vec{y}_n) = \vec{y'}. \tag{2.13}$$

Linearize the implicit midpoint rule around $\vec{f}(\vec{y}_n)$ to get the semi-implicit midpoint rule, which is given by

$$(\mathbf{1} - h\mathbf{J}) \cdot \vec{y}_{n+1} = (\mathbf{1} - h\mathbf{J}) \cdot \vec{y}_{n-1} + 2h(\vec{f}(\vec{y}_n) - \mathbf{J} \cdot \vec{y}_n), \tag{2.14}$$

where $\mathbf{J} = \frac{\partial \vec{f}}{\partial \vec{y}}$ is called the Jacobian matrix. Use

$$\vec{y}_1 = \vec{y}_0 + h\left(\vec{f}(\vec{y}_0) + \mathbf{J}|_{\vec{y}_0} \cdot (\vec{y}_1 - \vec{y}_0)\right) \tag{2.15}$$

to evaluate the first timestep taken and

$$\vec{y}_n = \frac{1}{2}(\vec{y}_{n+1} + \vec{y}_{n-1}) \tag{2.16}$$

for the final step taken.

Practical implementations of the Bader-Deuflhard method operate by advancing through a large timestep H by breaking it into a number of smaller substeps

$$h = \frac{H}{m}, \tag{2.17}$$

where h is the length of the substep and m is the number of such substeps. For notational compactness as Eqns. (2.14) through (2.16) are rewritten below, let

$$\vec{\Delta}_k = \vec{y}_{k+1} - \vec{y}_k. \tag{2.18}$$

Then Bader-Deuflhard proceeds in three phases from the initial time of t_n with initial values $\vec{y}_n$ to the final time of t_{n+1} with final values of $\vec{y}_{n+1}$. The first phase is the initial substep using the following equations:

$$(\mathbf{1} - \mathbf{J}) \cdot \vec{\Delta}_0 = h\vec{y'}_n \tag{2.19}$$

$$\vec{y}_1 = \vec{y}_n + \vec{\Delta}_0. \tag{2.20}$$

The second phase advances the system up to substep $m-1$. For $k = 1, 2, ..., m-1$

$$(\mathbf{1} - \mathbf{J}) \cdot \vec{x} = h\vec{y'}_k - \vec{\Delta}_{k-1}, \tag{2.21}$$

$$\vec{\Delta}_k = \vec{\Delta}_{k-1} + 2\vec{x}, \tag{2.22}$$

$$\vec{y}_{k+1} = \vec{y}_k + \vec{\Delta}_k. \tag{2.23}$$

The final phase advances from $m-1$ to m with

$$(\mathbf{1} - \mathbf{J}) \cdot \vec{\Delta}_m = h\vec{y'}_m - \vec{\Delta}_{m-1}, \tag{2.24}$$

$$\vec{y}_{n+1} = \vec{y}_m + \vec{\Delta}_m. \tag{2.25}$$

To get estimates of error, for stepsize control, and to use this as a higher order method, apply Richardson extrapolation to the Eqns. (2.17) through (2.24) [11] [13] and execute them at least twice with two different choices of m, with $m = 2$ and $m = 6$ giving a 5th order method [12]. It can be executed several times, with up to $m = 50$ in the 7th iteration [1] giving a 15th order method. For every timestep $\vec{y}_n$ to $\vec{y}_{n+1}$ that is advanced, at least two matrix inversions are calculated (it can involve more depending on how many iterations of this method are taken), making this method potentially more costly per timestep than implicit Euler.

Chapter 3

New Explicit Methods

3.1 Flux-Limited Forward Differencing Method

Forward Euler's method is one of the simplest ways to treat differential equations. However as demonstrated with examples in Chapter 2, Euler's method is unsuitable for tackling stiff ODEs. A common occurrence when using Euler's method to treat a system of stiff equations, like those that arise in nuclear astrophysics problems, is for one of the isotopic populations to become negative. The potentially many reactions which couple the isotopic populations together then transmit the effect of these negative populations and corrupt the entire system. This effect is demonstrated in Figure 2.5.

The system of differential equations that arises in nuclear astrophysics takes the form of

$$\frac{dY_i}{dt} = \sum_j F_{ij}, \tag{3.1}$$

where F_{ij} are the fluxes for the reactions that connect the population of species i to the other species in the network. It can also be rewritten in the form of

$$\frac{dY_i}{dt} = F_i^+ - F_i^- \tag{3.2}$$

where F_i^+ is the flux going into species i increasing its population and F_i^- is the flux going out of species i which decreases its population. Some reactions add to the abundance of a species, and other reactions subtract from it. For the purposes here, each reaction has a direction associated with it. By that we mean that a reaction such as the triple-alpha reaction ($^4He + {}^4He + {}^4He \rightarrow {}^{12}C$) depletes three ^4He nuclei and creates one ^{12}C nucleus. The flux resulting from the inverse reaction ($^{12}C \rightarrow {}^4He + {}^4He + {}^4He$) which takes a ^{12}C nucleus and breaks it apart into three ^4He nuclei is treated as a separate thing, not as a "negative" flux. For the population of ^{12}C, the triple-alpha reaction contributes to the flux into the box $F^+_{^{12}C} = F_{3^4He \rightarrow {}^{12}C} +$ other terms and its inverse contributes to the flux out of the box $F^-_{^{12}C} = F_{^{12}C \rightarrow 3^4He} +$ other terms. The equation to solve for ^{12}C then can be written as

$$\frac{dY_{^{12}C}}{dt} = F_{3^4He \rightarrow {}^{12}C} - F_{^{12}C \rightarrow 3^4He} + \text{other terms.} \tag{3.3}$$

The emphasis here is that by assigning reactions this way, the presence of negative fluxes is naturally eliminated. The reactions in astrophysical nuclear networks generally have the form of

$$F_{ij} = k_{ij} Y_i, \tag{3.4}$$

where k_{ij} is some parameterized thermonuclear reaction rate which depends on the temperature and density. The parameterized reaction rates are a measure of the probability of a single nucleus transitioning from species i to species j. Probabilities cannot be negative, so if $k_{ij} \geq 0$, and the populations are also $Y_i \geq 0$, then the flux $F_{ij} \geq 0$. The only way to get a negative flux in this scheme is for either an error in the calculation of k_{ij}, which can occur if the parameterization is outside its range of validity, or if the populations Y_i become negative. Assuming that the k_{ij} are calculated correctly in every timestep, then only negative populations can cause the fluxes to go negative. This suggests that if we can somehow keep the fluxes from becoming negative, we can limit the effect that a population becoming negative will have on the rest of the system.

Therefore, to prevent the system from being destabilized by the presence of a negative population via negative fluxes appearing and communicating the computational error to the rest of the system, we impose a constraint on the outward fluxes so that they never become negative. This flux constraint has the form

$$F_{ij} \geq 0. \tag{3.5}$$

Any flux that doesn't meet the condition of Eqn. (3.5) is set to 0, that is $F_{ij} < 0 \rightarrow F_{ij} = 0$. Combining this constraint on the fluxes with Euler's method is called the flux-limited forward differencing (FLFD) method.

3.1.1 4-Box Example Revisited

As a simple demonstration of the utility of the FLFD method, we return to the 4-box example worked out previously. Recall that it is a set of four coupled ODEs with some level of stiffness that causes the problem to become highly unstable and inaccurate for moderate timesteps when treated with a standard forward Euler method. In fact, the system is completely destabilized for a timestep h of 4.

Suppose, instead of using forward Euler to integrate the network, we use forward Euler modified with a flux constraint of the form of Eqn. (3.5). The results are shown in Figure 3.1.

For the smallest timestep, $h = 1.0$, the FLFD method produces results identical to those of forward Euler. For the larger timestep of $h = 3.9$, the network spends a small amount of time with the smallest population (Box 1 near and immediately after log(time) = 2) unstable, but the network quickly recovers and maintains stable, though the accuracy of results for the population of Box 1 is poor - this is still an improvement over the inaccuracies and behavior of forward Euler for this size of timestep. While forward Euler would go completely unstable with a timestep $h = 4.0$ and run off to infinity, for the FLFD results in Figure 3.1 the three largest populations converge to stability values near those of the accurate $h = 1.0$ calculation. The

smallest population oscillates between a small positive value (the numbers around -3 on the y-axis) and a small negative value (the numbers around -5 on the y-axis). This instability in the smallest population is prevented from propagating and affecting the populations in the other three boxes substantially by the flux constraint; additionally conservation of the total population is preserved. Stability has again been achieved for the populations of most consequence (the largest ones, i.e. Box 0, Box 2, and Box 3), while accuracy has been sacrificed for the population of Box 1. As the FLFD method assumes that the small populations are not important, a loss of accuracy in tracking Box 1 is acceptable and expected, while maintaining stability is a great improvement over the result obtained with forward Euler.

3.1.2 Nova Burning

Figure 3.2 shows an example calculation using the FLFD method for a thermonuclear network as might be used to simulate a classical nova outburst with fixed temperature of $T = 2.5 \times 10^8$ K and density of $\rho = 500$ g cm^{-3}. The calculation was made using the REACLIB library [14] for its reaction rate parameterization. For the arbitrarily selected populations illustrated in Figure 3.2(a) there is very good agreement between FLFD and an implicit calculation over the entire range. The FLFD implementation here used an adaptive timestep to limit the maximum population transferred in a timestep, and as shown in Figure 3.2(c) this timestep (represented by the curve labeled dt) is of order 10^{-2} seconds while the maximum stable explicit step is down around 10^{-7} seconds. An implicit calculation under these conditions using backward Euler takes timesteps which are initially comparable to the FLFD timestep, but after a time $t = 10$ seconds the implicit timestep becomes a factor of 10 larger than the FLFD timestep and beyond a time of about $t = 100$ seconds the implicit timestep becomes $dt_{\text{implicit}} > 1$ second up to a maximum of $dt_{\text{implicit}} = 15$ seconds. FLFD is able to keep the accuracy of the implicit method and take much larger timesteps than are allowed

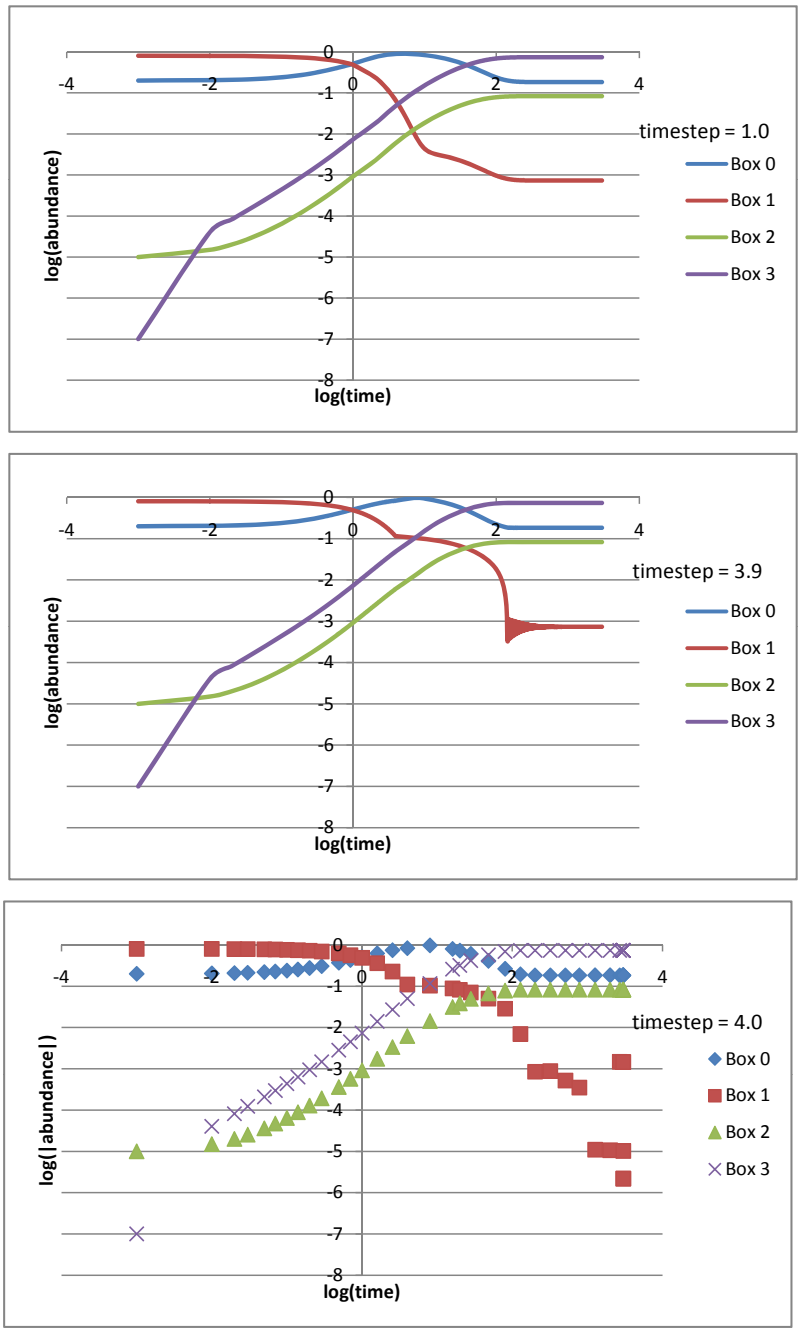


Figure 3.1: 4-box problem calculated using forward Euler with a Flux Constraint of the form $F_{ij} \geq 0$. The bottom graph differs from the two above it, plotting a selection of data points instead of smooth curves for purposes of making clear the behavior of the population of Box 1, this figure is also plotted with the absolute values of the abundances as the population of Box 1 becomes negative.

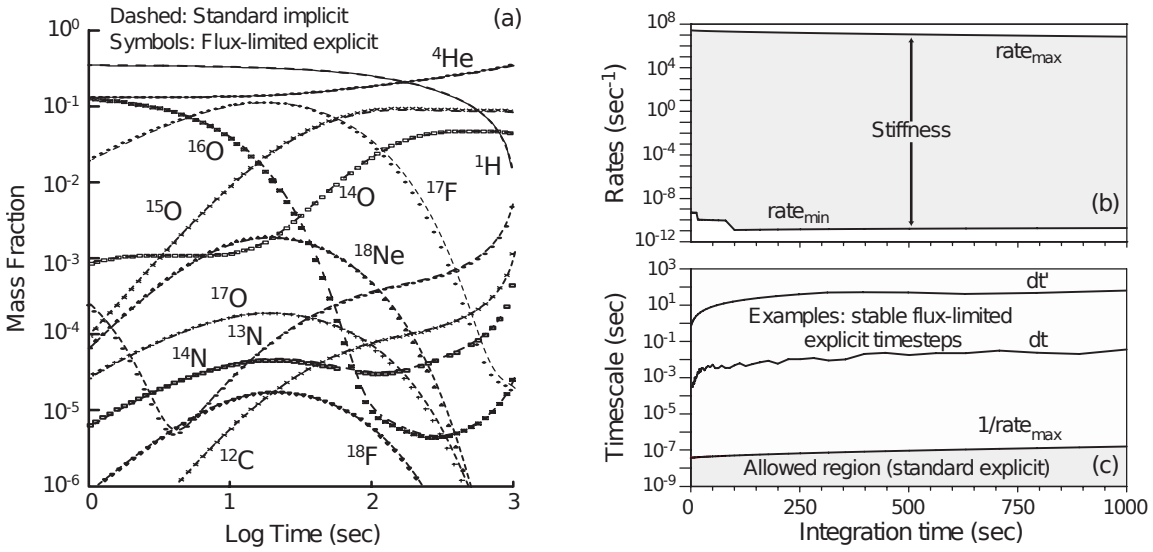


Figure 3.2: (a) Representative mass fractions for Nova burning. The dashed lines are from an implicit calculation and the symbol lines were calculated using FLFD. Calculation made with 145 isotopes and 924 non-zero reaction coupling the isotopic populations. (b) Difference between the fastest and slowest rates in the network for the same calculation as in (a) but with a larger network of 896 isotopes and 8260 non-zero reactions. (c) Comparison of the maximum stable timesteps for two representative integrations using FLFD against the maximum stable explicit timestep ($1/\text{rate}_{\text{max}}$) from (b). Reproduced from Ref. [15].

by standard explicit techniques for a very stiff system with a difference between the "fastest" and "slowest" rates of 18 orders of magnitude.

3.2 Equilibrium Stiffness

The FLFD method above relaxes one kind of stiffness in the calculation. By allowing small negative populations to creep into the system, but not allowing those populations to propagate bad fluxes, larger step sizes are possible than with standard Euler's method. This isn't enough on its own to produce an explicit method that can compete directly with implicit methods on very stiff systems such as those encountered in astrophysics.

As a typical nuclear reaction network calculation moves forward in time, it will approach an equilibrium condition where the flux going into (F^+, the sum of all F_{ij}

that add to the population of the box) and the flux coming out of a box (F^-, the sum of all F_{ij} that remove population from the box) will trend to become close to each other such that the difference between them will become very small in comparison to either number (effectively $F = F^+ - F^- \approx 0$ very close to equilibrium). As a physical example, the main isotopes involved in the CNO-cycle (^{12}C, ^{13}C, ^{13}N, ^{14}N, ^{15}N, and ^{15}O) reach this kind of equilibrium condition as they catalyze hydrogen into helium [16]. When this situation occurs small numerical errors in the calculation of F^+ or F^- can produce large changes in the net flux F. This particular form of equilibrium for a single isotope is referred to as the isotope being asymptotic.

These changes, caused by otherwise tiny errors in the calculation, can quickly skew a nuclear reaction network calculation in a manner similar to negative populations and negative fluxes. Imagine a situation in which a single isotope is asymptotic. The population of that isotope then won't change over time for as long as it remains asymptotic. If you then take a too large of a time step, the calculation of the directional components, F^+ and F^-, may have a slight inaccuracy in it. The total flux F (which is typically a small difference of very large numbers under these conditions) will then have a slight inaccuracy in its value. If this single isotopic population is truly asymptotic, its population shouldn't change, but due to numerical errors F may not be 0. This error will propagate and accumulate, and because of the interlinked nature of nuclear reaction networks it will feed back and can cause severe accuracy issues. The FLFD method can solve this by making the timesteps smaller and smaller to preserve accuracy.

This is a sort of stiffness instability, not due to populations becoming negative or the propagation of negative fluxes, but arising solely because of numerical errors magnified by asymptotic conditions. A population which doesn't change over time introduces a timescale approaching infinity to the system. So while the network's largest stable timestep is determined by the fastest reactions taking place within the network, stiffness has been greatly increased by the introduction of very long timescales upon which little to no change occurs. In general, the FLFD method

cannot on its own resolve stiffness arising from equilibria. The development of the asymptotic approximation, which helps alleviate stiffness arising from these asymptotic conditions, follows in the next section.

3.2.1 Asymptotic Approximation

The differential equations we solved in Section 3.1 had the form of

$$dY_i = \sum_j F_{ij} dt \tag{3.6}$$

which can be rewritten using the F^+ and F^- notation introduced in Section 3.2 into

$$\frac{dY_i}{dt} = F_i = F_i^+ - F_i^-, \tag{3.7}$$

where Y_i is the population of isotope i, F_i is the total or net flux of population i, F_i^+ is the flux into isotope i increasing its population, and F_i^- is the flux out of isotope i decreasing the population. For each isotope in the network, there will be one such equation like this, with the flux terms coupled to and dependent on the populations Y_j of other isotopes in the network, meaning that Eqn. (3.7) is often non-linear, with terms such as $Y_i Y_j$ or Y_i^3 appearing.

For astrophysical networks the flux out of the box for species i is proportional to the size of the population Y_i,

$$F_i^- = (k_1^i + k_2^i + ... + k_m^i) Y_i \equiv k^i Y_i, \tag{3.8}$$

where the k_n^i are rate parameters (in units of inverse time) for each of the m reactions that can remove population from Y_i, and k^i is an effective rate parameter summing all of the other rate parameters. The rate parameters may depend on various quantities

such as other populations Y_j, temperature, density, or even Y_i in the case of three-body reactions such as triple- α, which turns three helium nuclei into one carbon nucleus - ^{4}He(2^4He,γ)^{12}C.

The rate parameters naturally lend themselves to thinking about characteristic timescales, where the timescale τ is the inverse of the rate parameter k. Timescales for each individual reaction rate in the network can be defined such that $\tau_n^i = 1/k_n^i$. It is perhaps more useful to think of each isotope i as having its own timescale for the population to become depleted, and the effective rate parameter introduced on the right side of Eqn. (3.8) allows us to do that

$$k^i \equiv \frac{F_i^-}{Y_i}. \tag{3.9}$$

Eqn. (3.9) allows us to write Eqn. (3.7) as

$$Y_i = \frac{1}{k^i}(F_i^+ - \frac{dY_i}{dt}). \tag{3.10}$$

If we take a finite-difference approximation at timestep t_n we get

$$Y_i(t_n) = \frac{F^+(t_n)}{k^i(t_n)} - \frac{1}{k^i(t_n)}\frac{dY_i}{dt}\bigg|_{t=t_n} \tag{3.11}$$

So far this has only changed the form of the equations, the basic content is still the same. Recall that we want to reduce stiffness as we approach equilibrium in the system. Thus we can define an asymptotic limit for species i such that $F_i^+ \simeq F_i^-$, which as mentioned in Section 3.2 and implied by Eqn. (3.7) means that $dY_i/dt \simeq 0$. Therefore we can rewrite Eqn. (3.11) as

$$Y_i(t_n) = \frac{F_i^+(t_n)}{k^i(t_n)}, \tag{3.12}$$

which is a first approximation to the asymptotic abundance Y_i at time $t(n)$. We can improve the approximation by adding a correction term to Eqn. (3.12). To find

this correction term we use Eqn. (3.12) to approximate the derivative term in Eqn. (3.11) for the regime where $dY_i/dt \simeq 0$ is true. The derivative term at time t_n is approximate as its average between times t_n and t_{n-1} such that

$$\left.\frac{dY_i}{dt}\right|_{t=t_n} \simeq \frac{1}{\Delta t}(Y_i(n) - Y_i(n-1)), \tag{3.13}$$

where n is the index for the current timestep (at time $t(n)$), Δt is the timestep size, $Y_i(n)$ is the approximated Y from Eqn. (3.12) evaluated at timestep n, and $Y_i(n-1)$ is the same evaluated at timestep $n-1$. Substituting Eqn. (3.11) into Eqn. (3.13) gives

$$\left.\frac{dY_i}{dt}\right|_{t=t_n} = \frac{1}{\Delta t}\left(\frac{F_i^+(t_n)}{k^i(t_n)} - \frac{F_i^+(t_{n-1})}{k^i(t_{n-1})}\right). \tag{3.14}$$

If we substitute this back into Eqn. (3.11) we get an improved approximation,

$$Y_i(t_n) \simeq \frac{F_i^+(t_n)}{k^i(t_n)} - \frac{1}{k^i(t_n)\Delta t}\left(\frac{F_i^+(t_n)}{k^i(t_n)} - \frac{F_i^+(t_{n-1})}{k^i(t_{n-1})}\right). \tag{3.15}$$

This approximation is most valid when the right-hand term, which came from the approximated derivative, is small. This can be accomplished if the fluxes into population i are small, but is more likely to occur because the product $k\Delta t$ is large. If we normalize our populations such that $Y_i^{max} = 1$, then the fluxes F^- and the rate constant k^i will have the same units. Just as a total population transfer in one timestep, $\Delta Y_i = F^-\Delta t$, which is greater than Y_i will over-deplete a population and reduce it to a negative number, then if the product of the rate constant and the timestep becomes greater than 1 the population Y_i will be over-depleted (at least in the absence of any F^+ repopulating the box). However, with Eqn. (3.15) a larger $k^i\Delta t$ product means advancing more strongly into the regime of Eqn. (3.12) which cannot give a negative population (unless somehow a negative flux has already entered the network from somewhere).

3.2.2 Asymptotic Flux-Limiting Algorithm

The FLFD method can be combined with the asymptotic approximation into a new algorithm which attacks instability problems in two different ways. The basic outline of this algorithm is very similar to the FLFD method. First, in every timestep compute the rates, then from that calculate $k^i\Delta t$ for each isotope i using an initial trial timestep Δt. As discussed at the end of the previous section (3.2.1), this product $k^i\Delta t$ is important for determining whether the system will be stable under a standard explicit algorithm. If $k^i\Delta t < 1$ then the system will be stable with FLFD. If $k^i\Delta t \geq 1$, then the asymptotic algorithm will be necessary to compute the next timestep. Therefore for $k^i\Delta t < 1$, update the populations using the FLFD algorithm. If $k^i\Delta t \geq 1$ then update the populations using the asymptotic approximation as given by Eqn (3.15). This is similar to what standard asymptotic approaches, such as that employed by CHEMEQ [17] where the non-asymptotic equations are treated by a modified forward Euler predictor-corrector scheme and the asymptotic equations are treated with their own set of equations in a predictor-corrector scheme.

For any one timestep, some isotopic populations may be updated either by the FLFD method or by the asymptotic approximation. This lets the more powerful asymptotic techniques only be used when they are needed, and the less costly FLFD be used for the remaining populations. To optimize the algorithm further, it can be coupled to an adaptive timestepping algorithm, which should have a parameter to ensure the conservation of particle number (or mass) in the system as the asymptotic approximation does not conserve particle number (unlike FLFD); generally making timesteps smaller will help improve conservation of particle number, as it will lower $k^i\Delta t$ and reduce the number of isotopes being treated with the asymptotic approximation. The simple timestepping algorithm used in the below examples is as follows: if the sum of the fractional masses of all the populations in the system deviates from 1 by more than some tolerance, decrease the timestep by a factor of

10; if the sum of all the fractional masses is within 25% of the tolerance increase the timestep size by a factor of 2, otherwise continue using the same timestep.

3.2.3 Asymptotic Examples

An alpha network consists of a small set of isotopes and the reactions connecting the isotopes. Alpha networks are commonly used in astrophysics for many problems which couple nuclear reaction networks to larger simulations where computational cost is a major concern. An alpha network contains the following isotopes: ^{4}He, ^{12}C, ^{16}O, ^{20}Ne, ^{24}Mg, ^{28}Si, ^{32}S, ^{36}Ar, ^{40}Ca, ^{44}Ti, ^{48}Cr, ^{52}Fe, ^{56}Ni, and sometimes heavier isotopes that can be made from combining helium nuclei. The name alpha network comes from ^{4}He nuclei also being called alpha particles, and each nucleus in an alpha network can be created solely from the fusion of ^{4}He. Figure 3.3 graphically illustrates which isotopes are included in the network.

The asymptotic approximation requires that the total flux F be small in comparison to either F^{+} or F^{-}, or stated another way that $F^{+} \sim F^{-}$. Figure 3.4 shows the total flux F, the flux into the population F^{+}, the flux out of the population F^{-}, as well as the product kdt which we use in the asymptotic flux-limiting algorithm to determine if a population should be treated asymptotically or not. In the ^{4}He and ^{20}Ne graphs the dashed portion of the kdt curve is the section of the calculation performed using the non-asymptotic flux-limited forward differencing technique. After a time of about 1×10^{-5} seconds, ^{4}He becomes asymptotic. ^{20}Ne is treated asymptotically after about 1×10^{-1} seconds. For both of those isotopes, $F^{+} \sim F^{-}$ and the total flux F is satisfied before $kdt > 1$ occurs. Likewise that is true for ^{28}Si, however it differs in that while the condition $F^{+} \sim F^{-}$ is met after times of about 1 second, kdt is never greater than 0, thus ^{28}Si is never treated asymptotically and the FLFD method handles the computation.

Figure 3.5 shows the timesteps taken during the calculation. The network spends some time early increasing its timestep exponentially with an area of oscillation and

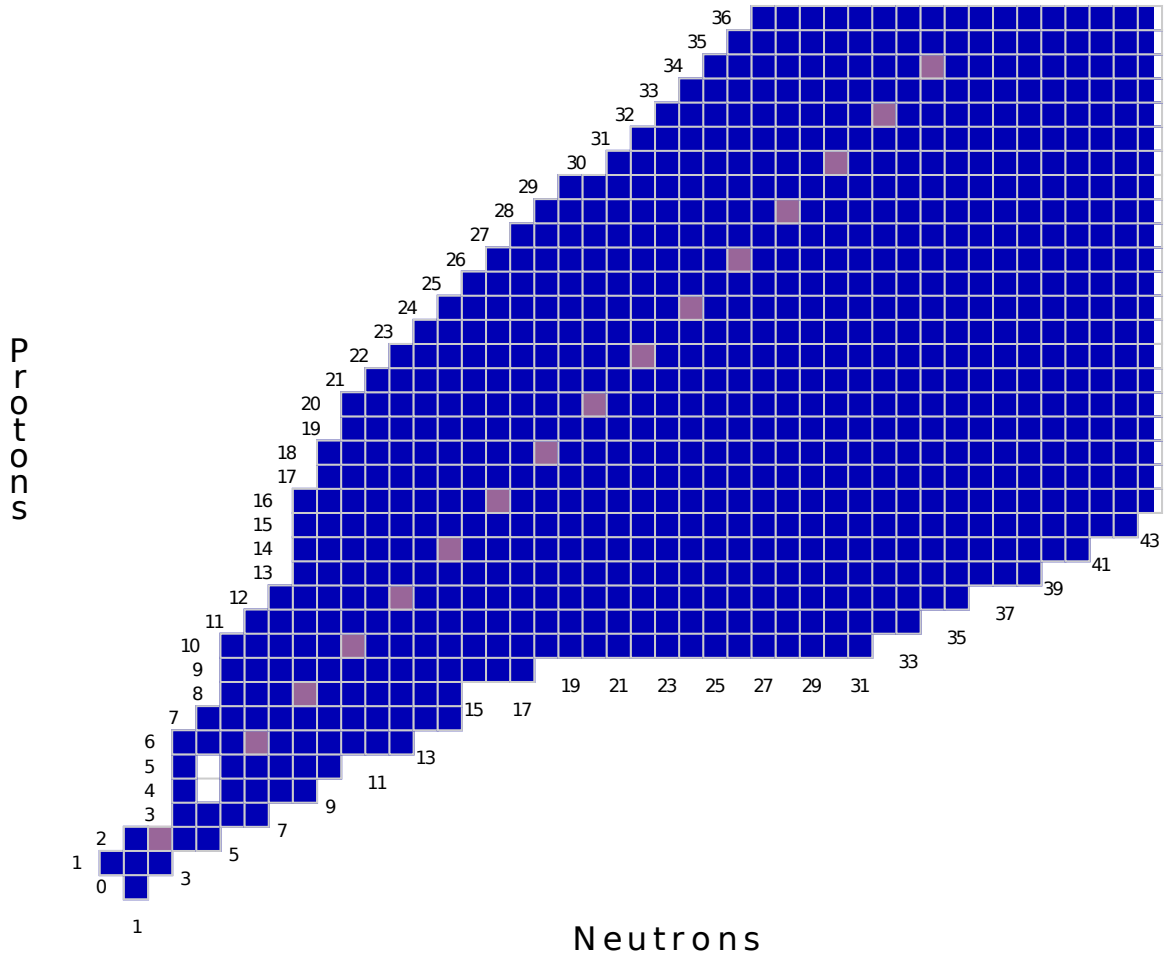


Figure 3.3: Schematic view of an alpha network. The purple colored boxes represent the isotopes included in the network.

no growth (around $t = 10^{-6}$ seconds). At about 10^{-2} seconds the network timestep flattens again, with the asymptotic approximation taking timessteps comparable to the maximum stable explicit timestep. After about 1 second however, the asymptotic approximation relaxes enough stiffness that the timesteps again increase and begin taking timesteps two or three orders of magnitude larger than the maximum allowed explicit timestep.

As reaction rates in astrophysical systems have non-linear and often exponential dependence on temperature, increasing the temperature in the system can have a dramatic effect upon it, speeding up reactions and potentially increasing the stiffness

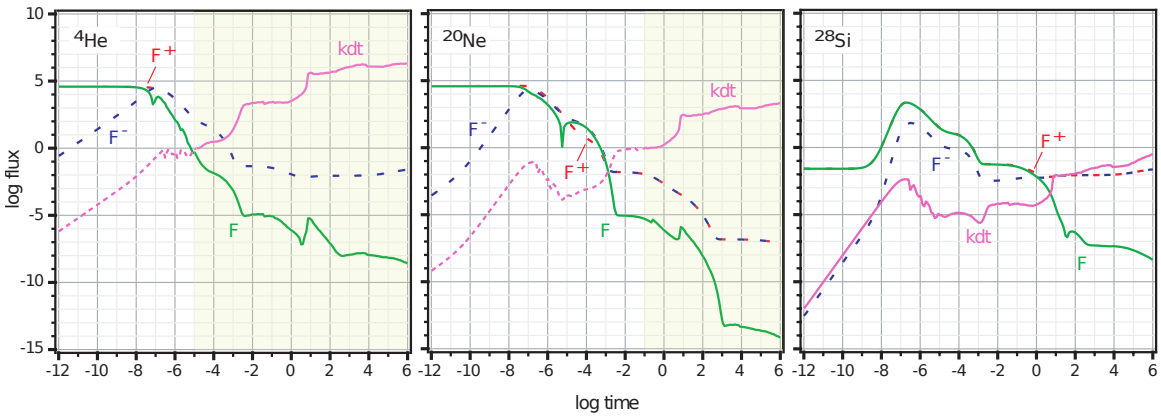


Figure 3.4: Fluxes and *kdt* for selected isotopes in an alpha network run out to 10^6 seconds with a constant temperature of 3×10^9 K and density $\rho = 1 \times 10^7$ g cm $^{-1}$. Figure reproduced from Ref. [18].

inherent in the nuclear reaction network. Therefore we present an example of the asymptotic approximation used to compute a second alpha network, this time with a higher fixed temperature of $T = 7 \times 10^9$ K and constant density of $\rho = 1 \times 10^8$ g cm^{-3} to present a more quantitative analysis of the asymptotic approximation than the above. It should be noted here that the alpha network used in this calculation has been extended to include heavier nuclei, up to ^{68}Se.

One of the requirements on the calculation is that in every timestep nucleon number is conserved to within a tolerance specified at run time. The conservation of nucleon number in this way is referred to as conservation of mass, though it is true that the mass of the nuclei in the system is not conserved as they undergo nuclear reactions. The tolerance parameter for this conservation is called the "mass tolerance" and is part of the timestep control algorithm. By making it a smaller number, the timesteps taken will generally become smaller and the system will better maintain accuracy and stability. As examples, a mass parameter of 10^{-6} requires the system to conserve mass to one part in a million between any two timesteps; a mass parameter of 10^{-9} requires the system to conserve mass to one part in a billion between any two timesteps.

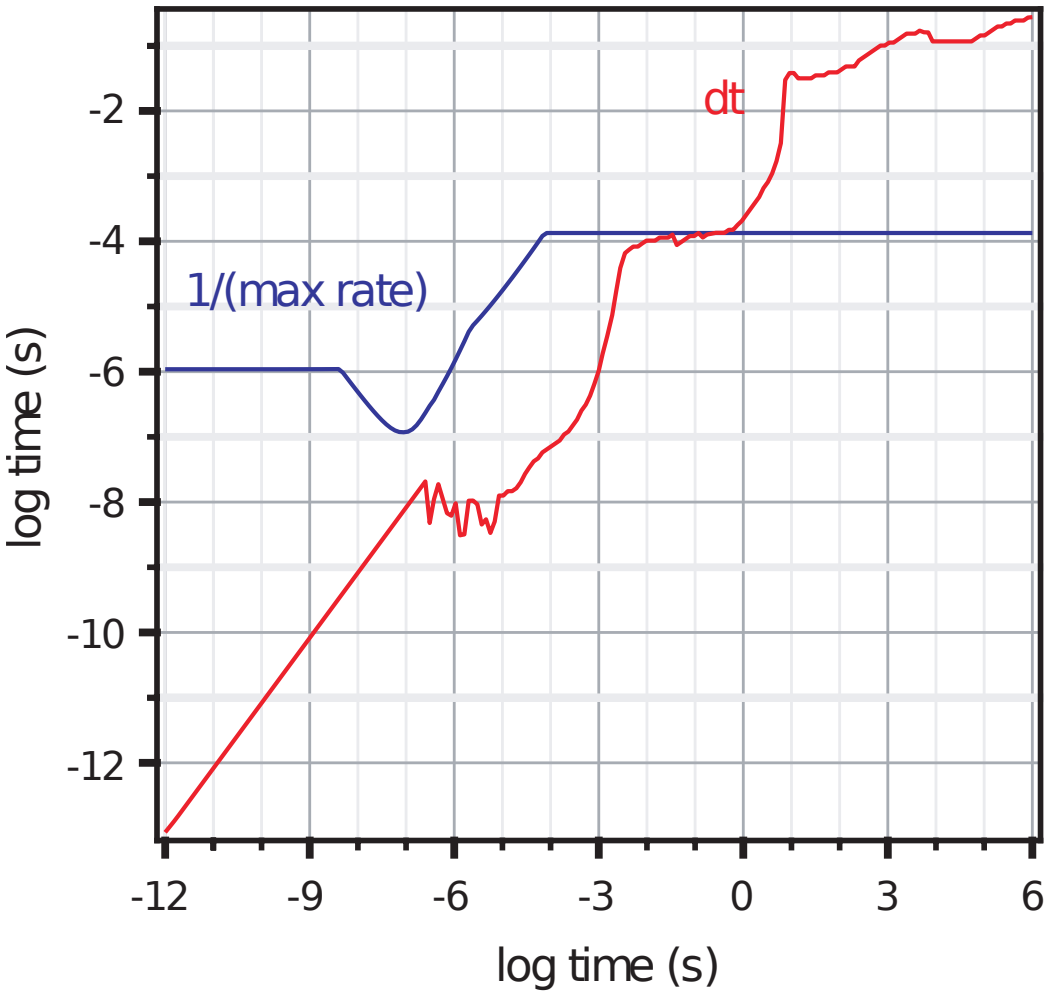


Figure 3.5: Timesteps taken by the asymptotic solver (dt) compared to the maximum stable explicit timestep 1/(max rate) for the calculation in Figure 3.4. Figure reproduced from Ref. [18].

Figure 3.6 shows the energy released by the nuclear reactions in the network as a function of time for both an exact numerical explicit integration with timesteps small enough to remain stable and several asymptotic approximation calculations made with varying mass parameters. Over most of the integration range, all the curves essentially lie on top of each other. It's only after a time of about 10^{-5} seconds that there is any deviation. The largest deviation occurs for the curve with a mass parameter of 10^{-6}, which deviates enough to cause the total integrated energy release to dip down an order of magnitude. The 10^{-7} curve has a much smaller deviation; and the 10^{-8} curve is essentially the same as the explicit calculation curve labeled "Exact".

Figures 3.7 and 3.8 show the evolution of the mass fractions of the populations for the case where the mass parameter is 10^{-6} and 10^{-8} respectively. A mass fraction X is defined as a ratio between the amount of mass contained in a single isotope's population and the total mass of the system $X_i = M_i/M_{total}$. At end of the calculation the populations are in nuclear statistical equilibrium.

For the more relaxed mass parameter of 10^{-6} the calculation is essentially correct until late in the calculation; it begins to diverge noticeably after a time of 10^{-5} seconds, which is also where strong deviation began on the dE/dt plot in Figure 3.6. A mass parameter of 10^{-8} provides essentially an exact match to the "Exact" numerical calculation.

Figures 3.9 and 3.10 provide a snapshot of the behavior of the same three isotopes looked at in Figure 3.4 under higher temperature and density conditions. It can be clearly seen how the mass parameter makes a difference in the nature of the calculation, with ^{28}Si becoming asymptotic much earlier in the mass parameter 1×10^{-6} case than it does in the 1×10^{-8} case. Figure 3.11 illustrates how a larger mass parameter allows for a larger timestep, with that case beginning to take timesteps larger than the explicit stable step much earlier, between 10^{-8} and 10^{-7} seconds, while the more conservative mass parameter doesn't exhibit that behavior until a time of

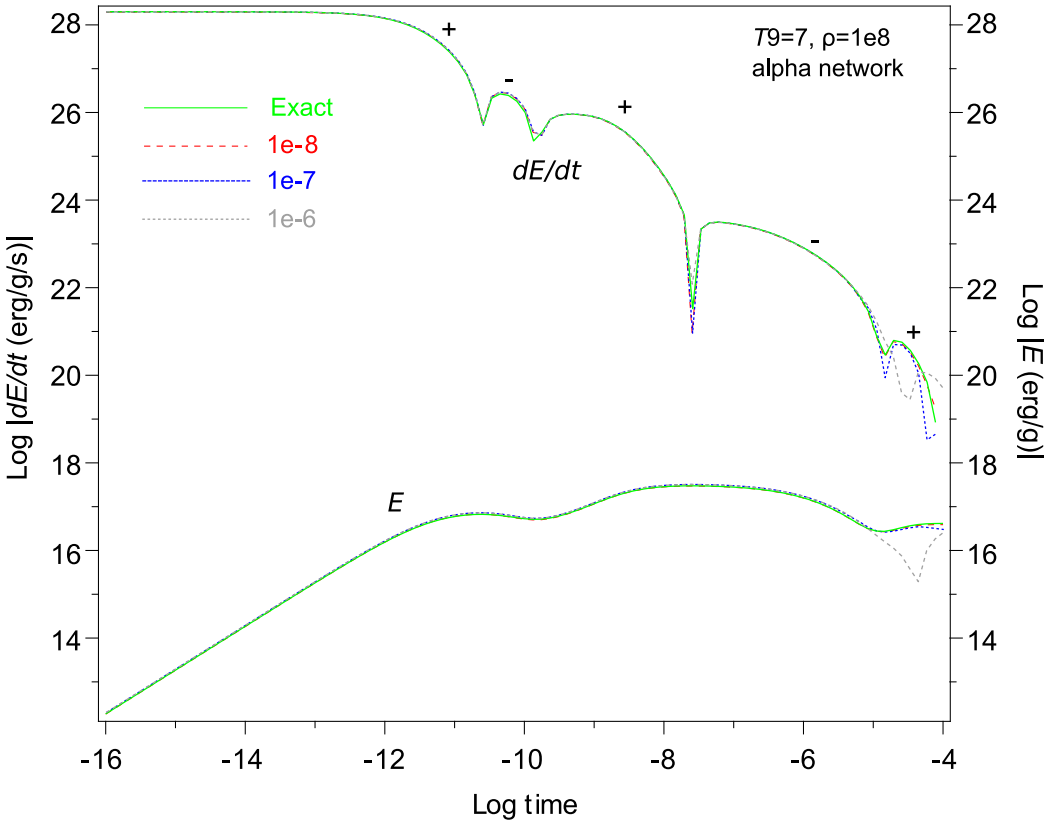


Figure 3.6: The energy production rate dE/dt and total energy production for an alpha network at a constant temperature of 7×10^9 K and density of 1×10^8 g cm^{-3}. The solid green line is an explicit calculation made with very small timesteps. The other curves are asymptotic approximation calculations made with varying mass tolerance parameters. From Ref. [18].

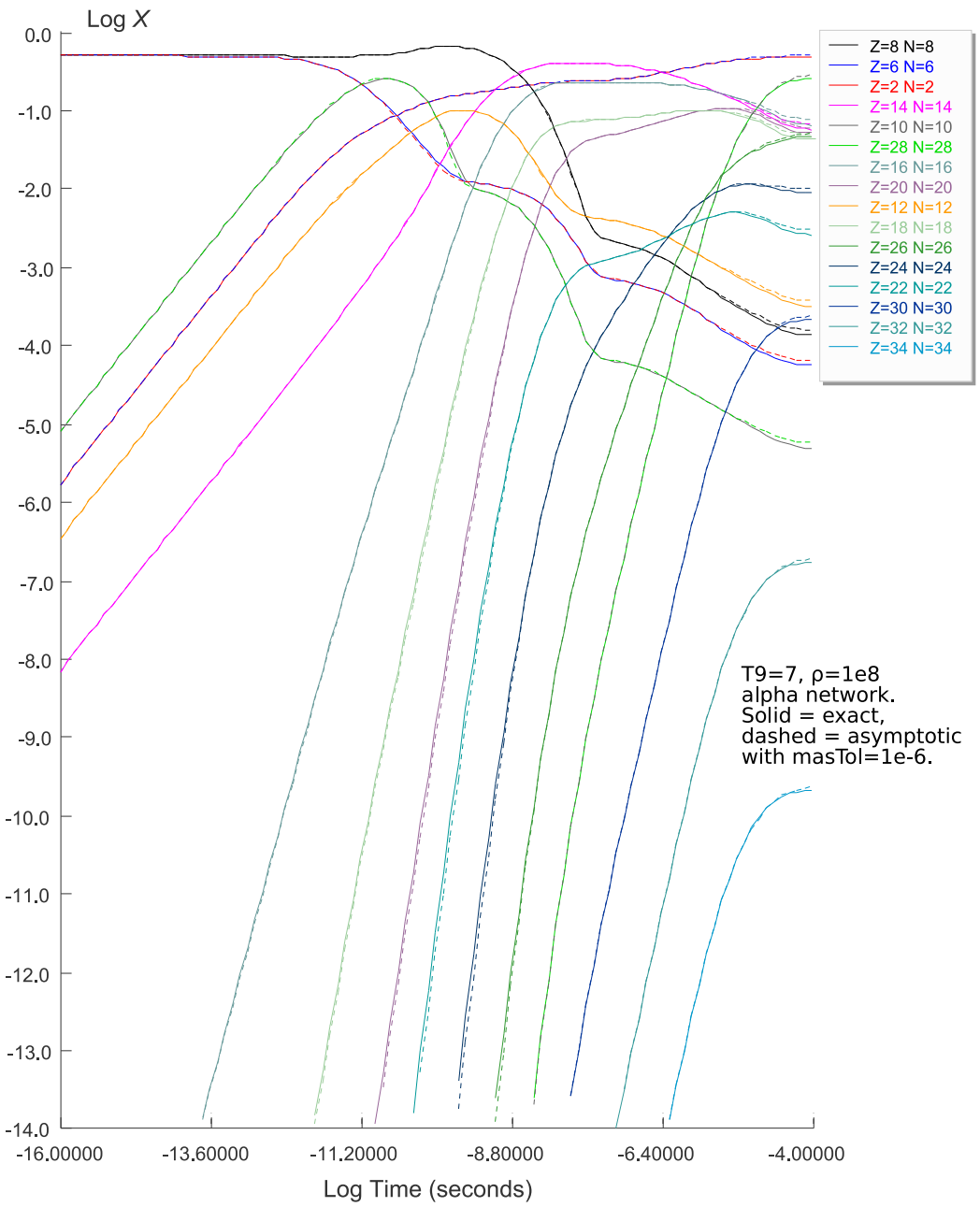


Figure 3.7: Mass fractions X for the populations in an alpha network. Solid lines are from an exact numerical integration and the dashed lines represent an asymptotic approximation with a mass parameter of 1×10^{-6}. From Ref. [18].

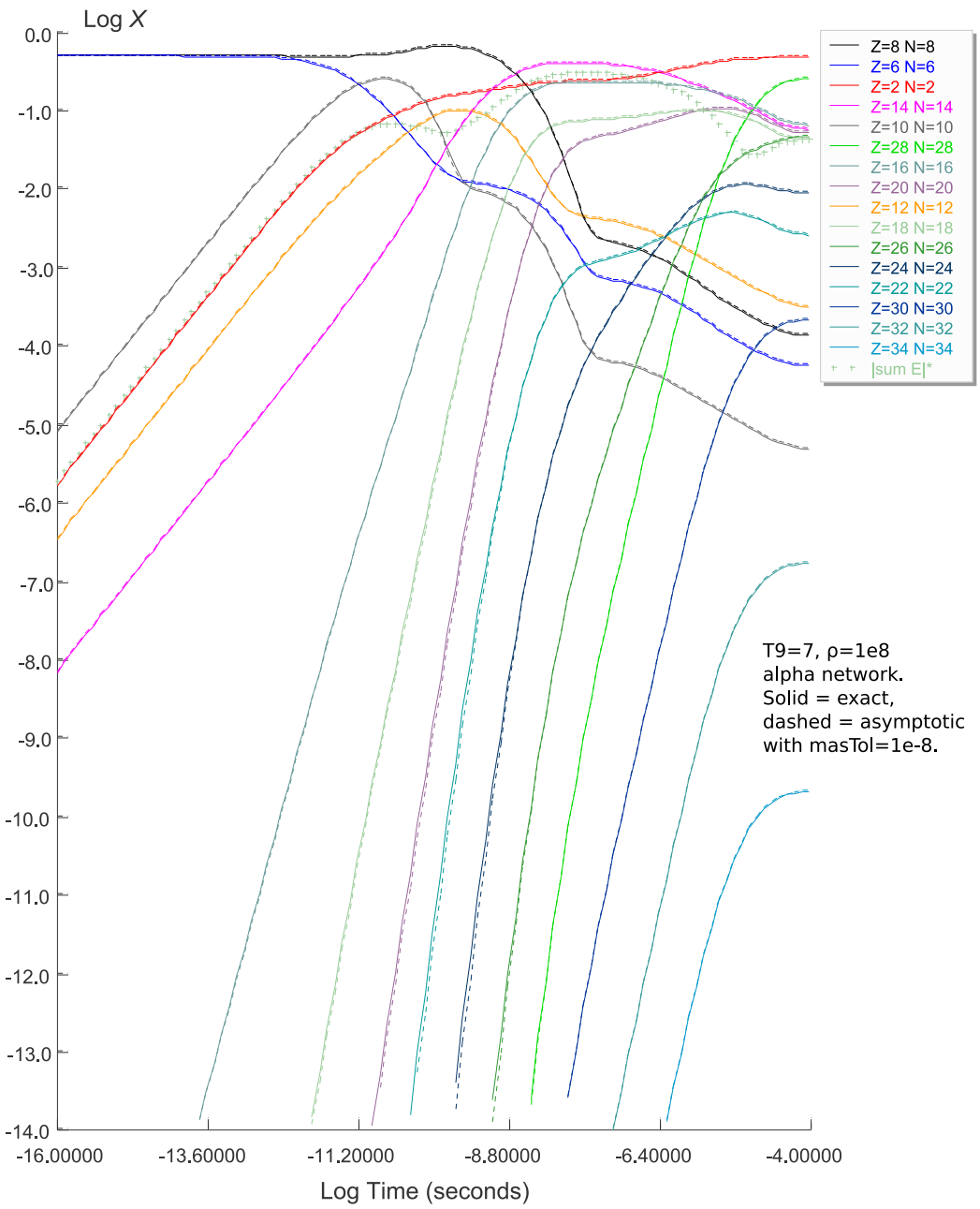


Figure 3.8: Mass fractions X for the populations in an alpha network. Solid lines are from an exact numerical integration and the dashed lines represent an asymptotic approximation with a mass parameter of 1×10^{-8}. From Ref. [18].

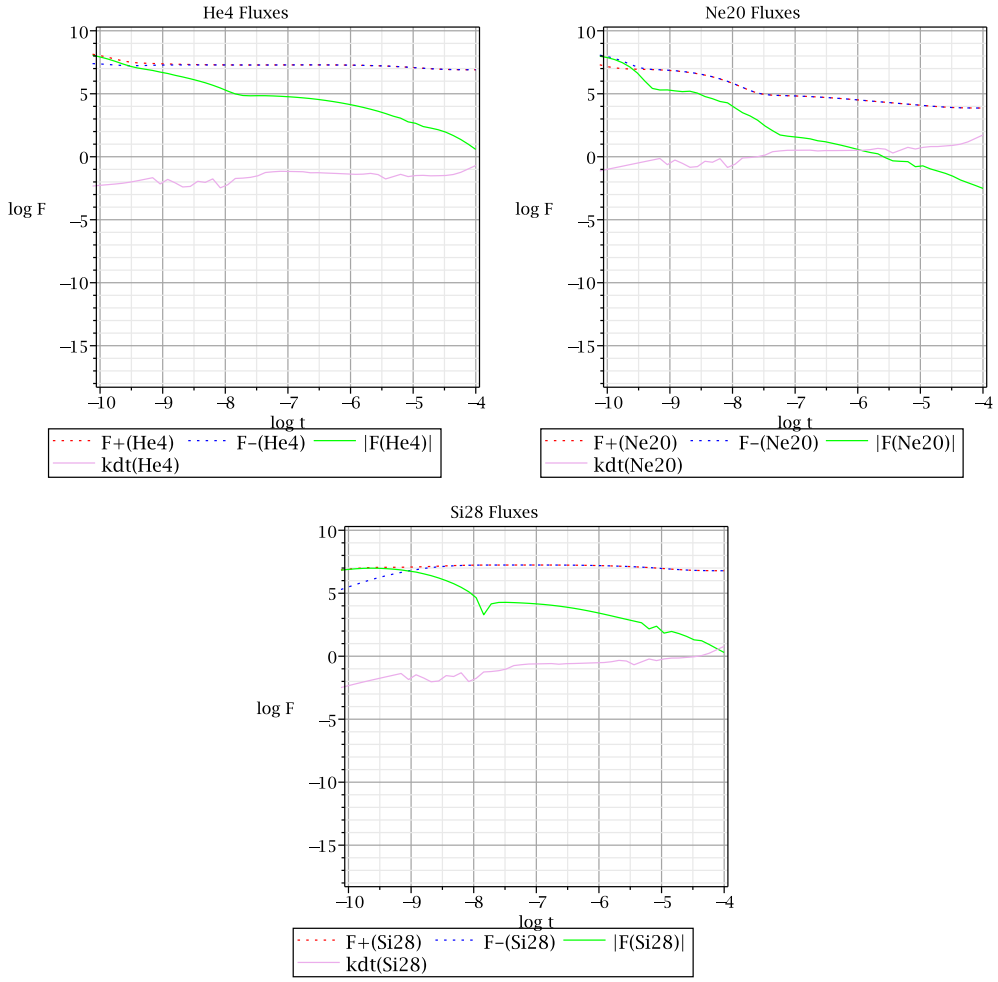


Figure 3.9: Fluxes and kdt for the calculation illustrated in Figure 3.7 with a mass parameter of 1×10^{-6}. By the end of the calculation both ^{20}Ne and ^{28}Si have become asymptotic.

10^{-6} seconds. Figure 3.12 shows the ratio of the dt to the maximum stable explicit step for the cases illustrated in Figures 3.7 and 3.8.

The reason for this increased timestep size is shown in Figure 3.13 where the larger mass parameter case is able to treat more isotopes with the asymptotic approximation at an earlier time. This relaxes the stiffness instability in the network by removing isotopes near equilibrium from the explicit calculation. The unfortunate side-effect of being too aggressive in our choice of timesteps (the 1×10^{-6} case took about 180,000 timesteps, the 1×10^{-8} case took about 475,000 timesteps), is the increasing loss

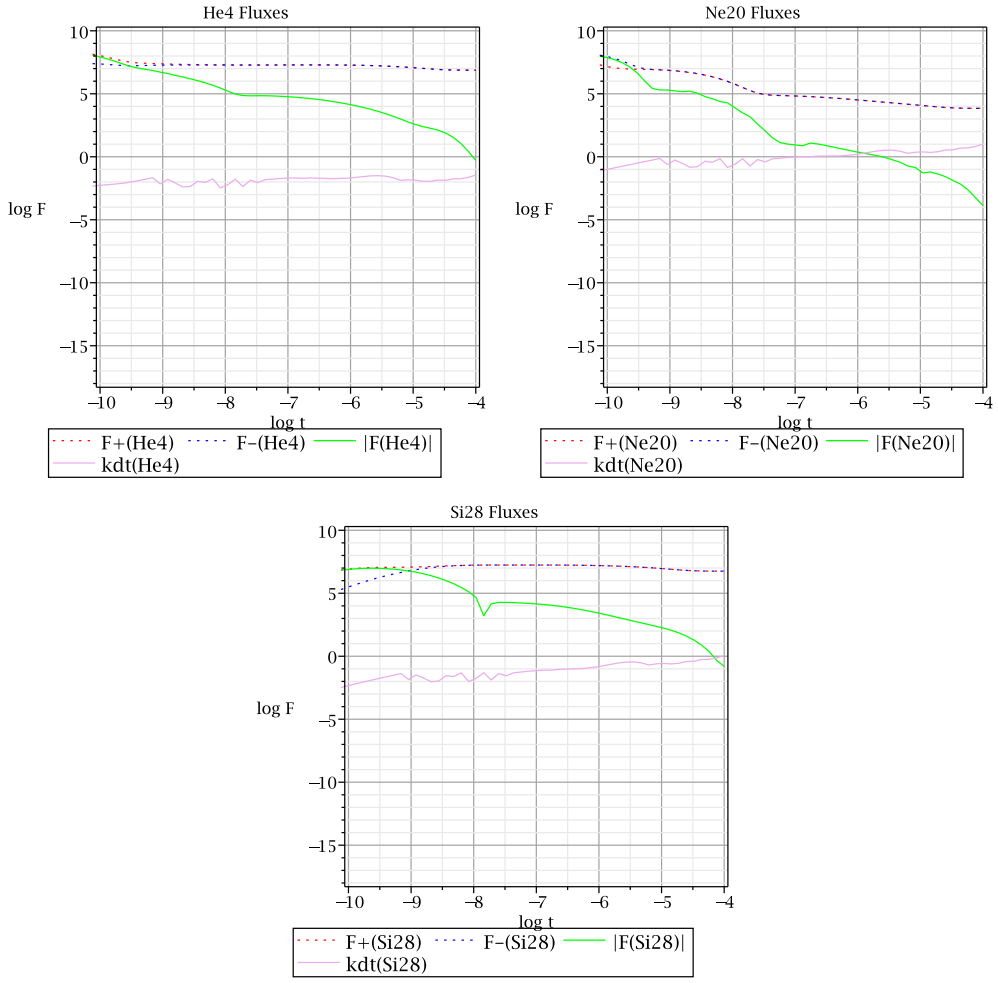


Figure 3.10: Fluxes and *kdt* for the calculation illustrated in Figure 3.8 with a mass parameter of 1×10^{-8}. By the end of the calculation of these three isotopes only ^{20}Ne has become asymptotic, though ^{28}Si is on the verge of becoming so.

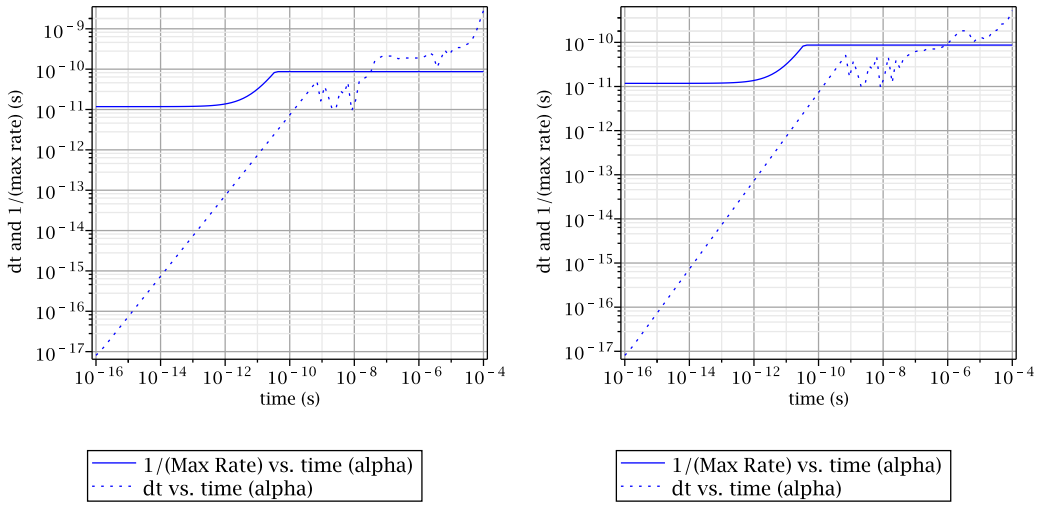


Figure 3.11: The network timesteps dt and maximum explicit timestep 1/(max rate) for the calculations depicted in Figures 3.7 (left) and 3.8 (right).

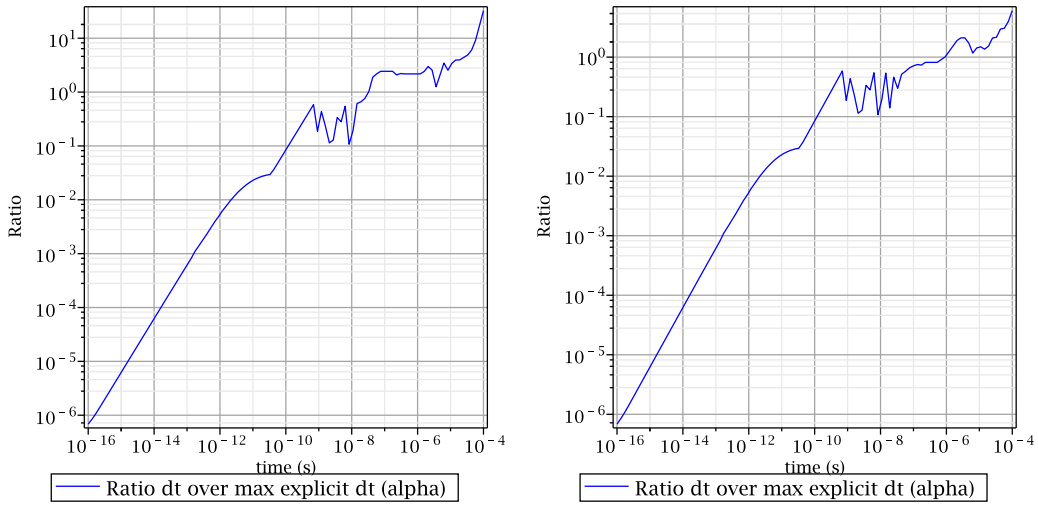


Figure 3.12: Ratio of the asymptotic timestep taken to the maximum stable explicit timestep for the calculations depicted in Figures 3.7 and 3.8. The left was calculated with a mass parameter of 1×10^{-6} and the right with a mass parameter of 1×10^{-8}.

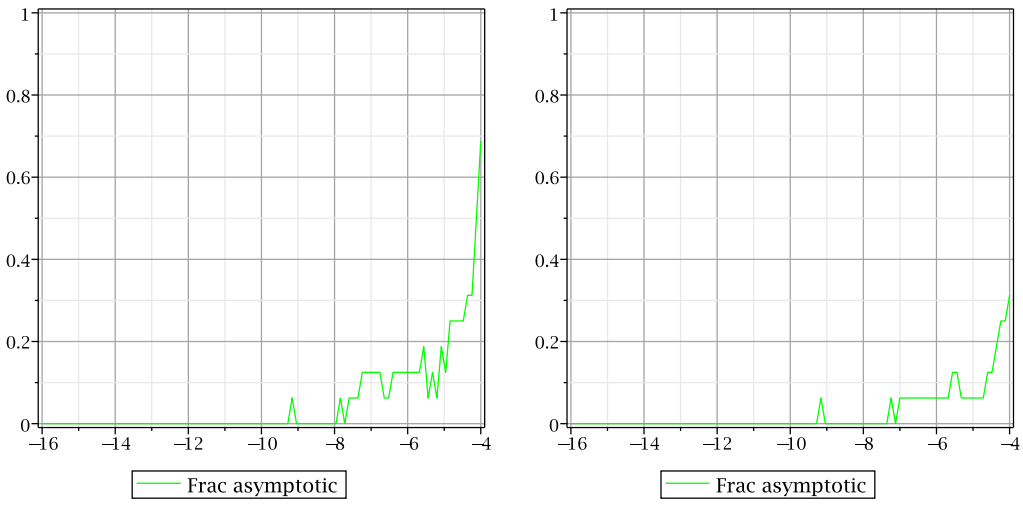


Figure 3.13: The fraction of isotopes in the network being treated asymptotically at any given time for the calculations depicted in Figures 3.7 and 3.8. The left illustrates the case from 3.7 and the right depicts the case from 3.8.

of mass conservation as the calculation continues (Figure 3.14). The increasingly equilibrated conditions in the network affect the 1×10^{-8} mass parameter case very little as far as accuracy is concerned, while the 1×10^{-6} case has, by the end of the calculation, accumulated 5% more mass in the network than when it started.

Figure 3.15 shows an alpha network computed with a mass parameter of 1×10^{-8} integrated out to a time of one second, where it has been in equilibrium for a considerable amount of time. The timestep taken by the network ceased to increase once it became in equilibrium, as shown in Figure 3.16. Figure 3.17 illustrates that this is because the network is unable to treat more populations as being asymptotic; it can no longer remove more stiffness from the system.

The lesson learned here is that while the asymptotic approximation as implemented in tandem with the FLFD method to create a asymptotic flux-limiting algorithm can relax stiffness from two different sources and enable a network to take timesteps several orders of magnitude larger than what is possible with a normal explicit technique, such as Euler's method, there is a danger in the undersirable loss of accuracy in the calculation due to taking timesteps which are too large. For

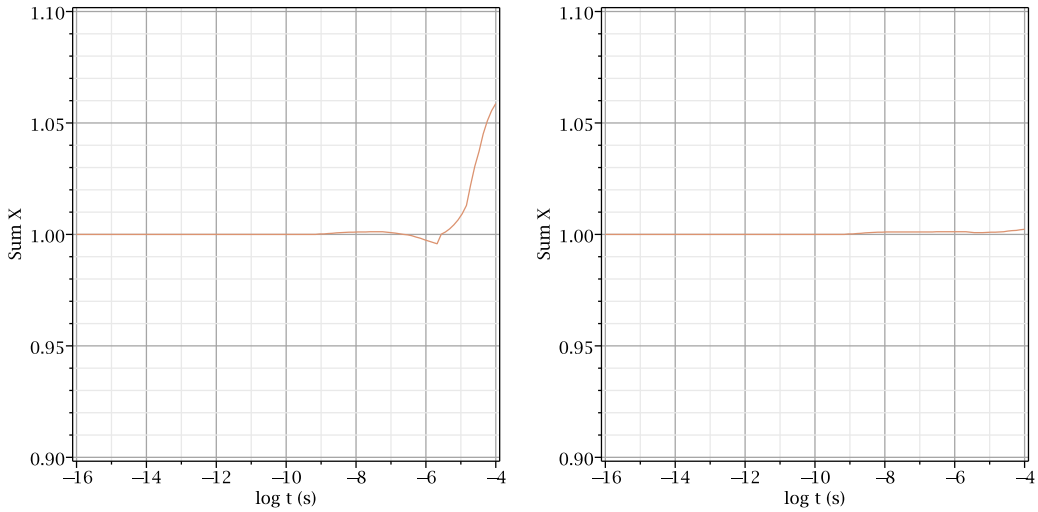


Figure 3.14: Sum of all the mass fractions for the calculations depicted in Figures 3.7 and 3.8, which should total to 1. The larger mass parameter case (1×10^{-6}) on the left deviates much more strongly from mass conservation than the 1×10^{-8} case on the right.

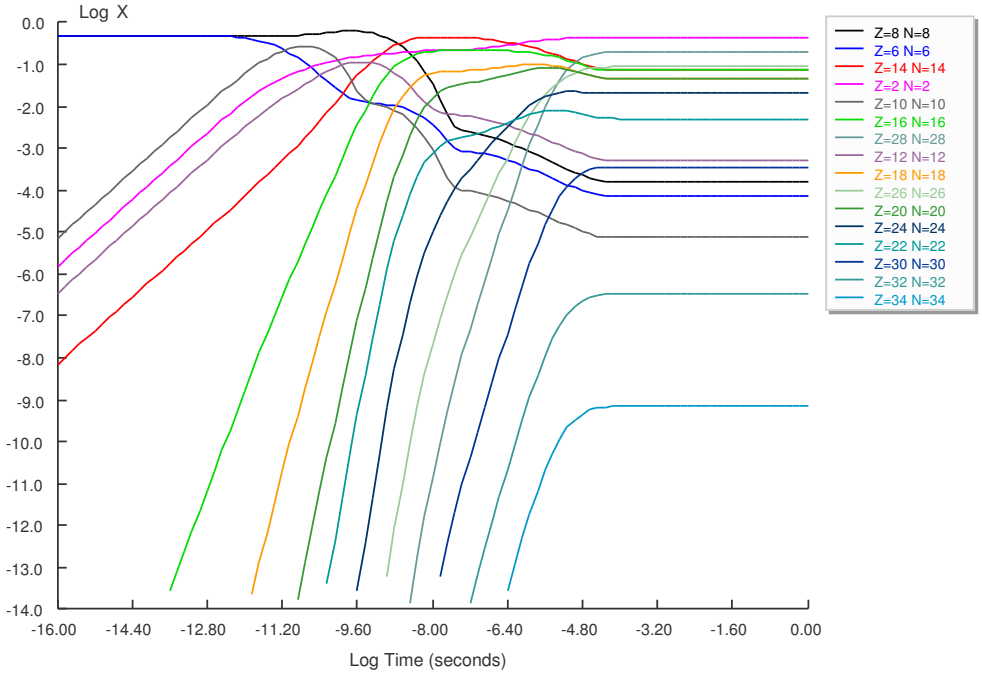


Figure 3.15: Mass fractions for the same calculation as in Figure 3.8 (mass parameter of 1×10^{-8}). By a time of 1 second, the network has been in equilibrium for a much longer time than it spent out of equilibrium.

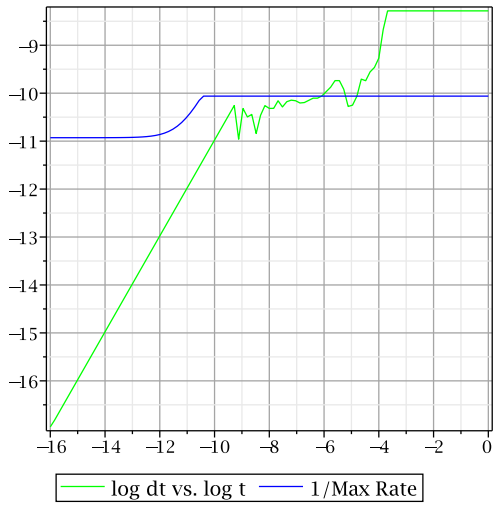


Figure 3.16: The timestep dt versus time for the same case as Figure 3.15.

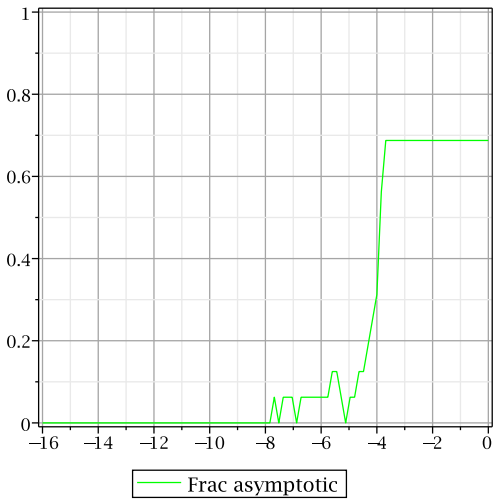


Figure 3.17: The fraction of isotopes in the alpha network for Figure 3.15 which are treated asymptotically in any timestep.

some applications, this asymptotic approach may be enough. Chapter 4 examines the behavior of the asymptotic flux-limiting algorithm in a variety of astrophysical problems, coupling the network calculation to realistic thermodynamic profiles.

3.3 Partial Equilibrium

As demonstrated in previous sections, the flux-limited forward differencing and asymptotic approximation help to relieve stiffness. However, with the loss of accuracy implied by Figure 3.14, equilibrium and the approach to it remain difficult for an explicit method to deal with. An increase in timestep size of one order of magnitude by relaxing timestep constraints caused the system at the end of the run to stop conserving particle number in the system and gain mass. Any results from such a calculation are thus immediately suspect. A technique that can more ably deal with equilibrium conditions is required to remove instability from the system, preserve accuracy, and increase the size of timesteps.

David Mott wrote a doctoral dissertation on applying quasi-steady state and partial equilibrium methods to chemical reaction networks . Included in this work is an attempt to solve an alpha network [19], which inspired the development here. These ideas have been around for some years in both the physics [20] and chemistry [21] literatures.

A partial equilibrium approach would interrogate individual reactions, determine if the reaction is in equilibrium with its inverse (if α-capture on a ^{24}Mg nucleus to produce ^{28}Si nucleus is the 'forward' reaction, then photodisintegration of ^{28}Si back into an α-particle and a ^{24}Mg nucleus would be the 'inverse'), and if it is, then remove the term from the equations being integrated numerically and use a simple algebraic formula to carry the information about that reaction back to the remainder of the network. This both simultaneously removes some of the stiffest terms from the numerical calculation while not reducing the size of the reaction network. All reactions are still taken into account, the only change is which ones are calculated

with an explicit numerical integration and which are solved algebraically. Much of the following discussion is adapted from Ref. [18].

For a generic two-body reaction of the form

$$a + b \rightleftharpoons c + d \tag{3.16}$$

the term contributed to the system of equations takes the form of

$$F_{a+b\rightleftharpoons c+d} = \pm(k_f Y_a Y_b - k_f Y_c Y_d), \tag{3.17}$$

where the Y_i are the populations for species i and $F_{a+b\rightleftharpoons c+d}$ is of course the flux. Using the above equation, the system of differential equations for these four isotopes a, b, c, d can be written as follows:

$$\frac{dY_a}{dt} = F_{a+b\rightleftharpoons c+d} + \text{other terms changing } Y_a \tag{3.18}$$
$$\frac{dY_b}{dt} = F_{a+b\rightleftharpoons c+d} + \text{other terms changing } Y_b \tag{3.19}$$
$$\frac{dY_c}{dt} = -F_{a+b\rightleftharpoons c+d} + \text{other terms changing } Y_c \tag{3.20}$$
$$\frac{dY_d}{dt} = -F_{a+b\rightleftharpoons c+d} + \text{other terms changing } Y_d. \tag{3.21}$$

Removing this one source term from the network influences multiple populations, potentially alleviating the stiffness of multiple equations and the whole system.

Partial equilibrium (PE) then occurs when a single pair of reactions such as that illustrated in Eqn. (3.16) is in equilibrium with itself ($F = 0$) while other pairs of reactions in the network are not in equilibrium ($F \neq 0$). The non-equilibrium reactions will perturb the equilibrium reaction pairs so that $F = 0$ becomes $F \sim 0$. Therefore consider a system to be in PE when it has two sets of reactions: those near equilibrium ($F \sim 0$) and those not in equilibrium ($F \neq 0$). The members of either set of reactions may change over time as the system evolves.

This poses the challenge then of systematically identifying reactions that are in or near equilibrium, and then removing them from the numerical integration and treating them algebraically. This decouples the equilibrated reactions from the network, and given that a pair of reactions in equilibrium introduces a timescale that approaches infinity, this should cause the stiffness in the numerically integrated portion of the system to decrease dramatically. (As the net flux of the reactions approaches zero, the "effective" rate governing the pair of reactions approaches zero; 1/(a very small number) is very large, and the situation only gets worse the closer the reactions become to being in equilibrium.)

Before any of this equilibrium stiffness can be removed, there first needs to be a test or measure to understand quantitatively what $F \sim 0$ means for a reaction pair. Just as in the asymptotic approximation we required that the total flux for an isotope be small compared to either the flux in or out of the isotope, we can analogously test whether the net flux of the pair of reactions is small to the sum of the forward and inverse components of the flux, which using Eqn. (3.17) can be written as:

$$\frac{|F_{a+b\rightleftharpoons c+d}|}{(k_f Y_a Y_b + k_f Y_c Y d)} = \frac{(k_f Y_a Y_b - k_f Y_c Y d)}{(k_f Y_a Y_b + k_f Y_c Y d)} < \varepsilon, \tag{3.22}$$

where ε is an arbitrarily small quantity.

For the reaction pair $a + b \rightleftharpoons c + d$ there are four distinct timescales based on the rate of change of the four populations Y_a, Y_b, Y_c, and Y_d. Consider the pair of reactions $a + b \rightleftharpoons c + d$ and the populations Y_a, Y_b, Y_c, and Y_d in isolation from the rest of the reaction network. The differential equation governing the population of a is then given by

$$\frac{dY_a}{dt} = F_{a+b\rightleftharpoons c+d}. \tag{3.23}$$

Substituting the source term from Eqn. (3.17) into this we get,

$$\frac{dY_a}{dt} = k_r Y_c Y_d - k_f Y_a Y_b. \tag{3.24}$$

Let us introduce a timescale τ_a. Since a flux must have units of particles per time, the terms on the right side of Eqn. (3.24) must have the same units. Therefore since Y_a has units of particles, $k_f Y_b$ must have units of inverse time. Let

$$\tau_a = \frac{1}{k_f Y_b}, \tag{3.25}$$

so that Eqn. (3.24) can be written as

$$\frac{dY_a}{dt} = k_r Y_c Y_d - \frac{Y_a}{\tau_a}. \tag{3.26}$$

To simplify the notation, let $F_a^+ = k_r Y_c Y_d$ so that we can write

$$\frac{dY_a}{dt} = F_a^+ - \frac{Y_a}{\tau_a}, \tag{3.27}$$

where F_a^+ represents the flux into species a for this reaction pair.

Assuming that the rate parameters and the populations of species b, c and d remain approximately constant over the timestep being taken in the numerical integration then τ_a is a measure of the rate of change for the species a. We can similarly write differential equations for Y_b, Y_c, and Y_d and derive timescales for them as well:

$$\tau_b = \frac{1}{k_f Y_a} \quad \tau_c = \frac{1}{k_r Y_d} \quad \tau_b = \frac{1}{k_r Y_c}. \tag{3.28}$$

This gives us four different ways to characterize the time for the reaction pair $a + b \rightleftharpoons c + d$ to reach equilibrium from its disturbed near-equilibrium state, but it isn't intuitive to understand which of the four, or what combination of them, best characterizes the timescale for the reaction pair to reach equilibrium. To determine a natural timescale for the reaction pair, we will use conserved scalars, which are linear combinations of dependent variables that do not vary over time.

3.3.1 Conserved Scalars

We begin with an example of this idea. Consider the reaction pair

$$a \rightleftharpoons 2b, \tag{3.29}$$

which has for its source term

$$F_{a \rightleftharpoons 2b} = k_f Y_a - k_r Y_b Y_b = k_f Y_a - k_r Y_b^2. \tag{3.30}$$

Considered in isolation from the rest of the system this gives the following differential equations:

$$\frac{dY_a}{dt} = -F_{a \rightleftharpoons 2b} \tag{3.31}$$

and

$$\frac{dY_b}{dt} = 2F_{a \rightleftharpoons 2b}. \tag{3.32}$$

Manipulating Eqn. (3.32) gives

$$\frac{dY_b}{dt} - 2F_{a \rightleftharpoons 2b} = 0. \tag{3.33}$$

Eqn. (3.31) can be solved for $F_{a \rightleftharpoons 2b}$, and then substituting that into Eqn. (3.33) gives

$$\frac{2dY_a}{dt} + \frac{dY_b}{dt} = 0. \tag{3.34}$$

Eqn. (3.34) can be simply integrated, resulting in

$$2Y_a + Y_b = C, \tag{3.35}$$

where C is the constant of integration. This quantity then is a time independent value, and an example of a conserved scalar. The conservation arises solely from the nature of the reaction pair in Eqn. (3.29) and not any effect of the dynamics of

the system. Assuming there is nothing else in the systen to create or destroy a or b isotopes, the quantity in Eqn. (3.35) is conserved regardless of whether the system is in equilibrium or not; which should be expected because Eqn. (3.35) is a statement that nucleon number is conserved by the reaction.

To develop these ideas further and apply them to the more general $ab \rightleftharpoons cd$, it is useful to define variables which are the difference between the initial and current values of the populations

$$\delta Y_a \equiv Y_a - Y_a(t=0) \qquad \delta Y_b \equiv Y_b - Y_b(t=0), \tag{3.36}$$

where $Y_i(t=0)$ is the initial population for species i. The time derivative of the left-hand equation is

$$\frac{d\delta Y_a}{dt} = \frac{dY_a}{dt} - \frac{dY_a(t=0)}{dt}, \tag{3.37}$$

but since $dY_a(t=0)$ is a constant the right hand term drops out to give

$$\frac{d\delta Y_a}{dt} = \frac{dY_a}{dt}. \tag{3.38}$$

We know from Eqn. (3.31) what $\frac{dY_a}{dt}$ is, and if we substitute that into Eqn. (3.38) we find

$$\frac{d\delta Y_a}{dt} = -F_{a \rightleftharpoons 2b}. \tag{3.39}$$

By analogous construction we can do the same for $d\delta Y_b$,

$$\frac{d\delta Y_b}{dt} = 2F_{a \rightleftharpoons 2b}. \tag{3.40}$$

Note that the forms of Eqns. (3.39) and (3.40) are the same as seen in Eqns. (3.31) and (3.32). It is to be expected that since the difference between δY_i and Y_i is only an additive constant that the differential equations for them should be essentially the same. However, this hints that defining a variable λ which characterizes the "progress" of the entire reaction as one population converts into another may be

useful. Hence let us define λ as the progress variable for the reaction $a \rightleftharpoons 2b$ such that λ satisfies the following equations:

$$\frac{d\lambda}{dt} = F_{a \rightleftharpoons 2b} \qquad \lambda(t=0) = 0, \tag{3.41}$$

where $\lambda(t=0)$ is the initial value for λ. Substituting Eqn. (3.41) into Eqns. (3.39) and (3.40) and solving them (noting that $\delta Y_i = 0$) results in

$$\lambda = -\delta Y_a \qquad \lambda = \frac{1}{2}\delta Y_b. \tag{3.42}$$

Substituting this result into the definitions for δY_a and δY_b given in Eqn. (3.36) gives

$$Y_a = \delta Y_a + Y_a(t=0) = -\lambda + Y_a(t=0) \tag{3.43}$$

$$Y_b = \delta Y_b + Y_b(t=0) = 2\lambda + Y_b(t=0). \tag{3.44}$$

Because of the conserved scalar concept, the system of two differential equations with two unknown variables has been reduced to a single differential equation in terms of the variable λ from which the populations Y_a and Y_b can be computed.

Using all of these tools, let's apply the conserved scalar concept to the more general case of the two-body reaction $ab \rightleftharpoons cd$. In the absence of outside terms contributing to the Y_i, Eqns. (3.18) through (3.21) become

$$\frac{dY_a}{dt} = -F_{ab \rightleftharpoons cd} \tag{3.45}$$

$$\frac{dY_b}{dt} = -F_{ab \rightleftharpoons cd} \tag{3.46}$$

$$\frac{dY_c}{dt} = F_{ab \rightleftharpoons cd} \tag{3.47}$$

$$\frac{dY_d}{dt} = F_{ab \rightleftharpoons cd}. \tag{3.48}$$

Subtract Eqn. (3.46) from Eqn. (3.45) to get

$$\frac{dY_a}{dt} - \frac{dY_b}{dt} = -F_{ab\rightleftharpoons cd} - (-F_{ab\rightleftharpoons cd}) = -F_{ab\rightleftharpoons cd} + F_{ab\rightleftharpoons cd} = 0. \tag{3.49}$$

Integrate Eqn. (3.49) to find

$$Y_a - Y_b = c_1, \tag{3.50}$$

where c_1 is the constant of integration. Add Eqn. (3.45) and Eqn. (3.47),

$$\frac{dY_a}{dt} + \frac{dY_c}{dt} = -F_{ab\rightleftharpoons cd} + F_{ab\rightleftharpoons cd} = 0, \tag{3.51}$$

and then integrate the result to get

$$Y_a + Y_c = c_2, \tag{3.52}$$

where c_2 is the constant of integration. Similarly, add Eqn. (3.45) and Eqn. (3.48) to find

$$\frac{dY_a}{dt} + \frac{dY_d}{dt} = -F_{ab\rightleftharpoons cd} + F_{ab\rightleftharpoons cd}) = 0, \tag{3.53}$$

which can be integrated to find

$$Y_a + Y_d = c_3, \tag{3.54}$$

where c_3 is the constant of integration. Eqns. (3.50), (3.52), and (3.54) then define a trio of constants, the conserved scalars for this reaction pair. Again, these constants emerge solely from the structure of the reaction pair and not from any dynamical consideration. Assuming that the initial conditions of the problem are known, these constants can be evaluated by substituting the initial populations into the above

equation,

$$c_1 = Y_a(t=0) - Y_b(t=0) \tag{3.55}$$

$$c_2 = Y_a(t=0) + Y_c(t=0) \tag{3.56}$$

$$c_3 = Y_a(t=0) + Y_d(t=0). \tag{3.57}$$

The source term for this reaction is

$$F_{ab\rightleftharpoons cd} = k_f Y_a Y_b - k_r Y_c Y_d. \tag{3.58}$$

The ordinary differential equation (ODE) corresponding to the population Y_a is then

$$\frac{dY_a}{dt} = -F_{ab\rightleftharpoons cd} = -k_f Y_a Y_b - k_r Y_c Y_d. \tag{3.59}$$

Using our conserved scalars c_1, c_2, and c_3 we can define a trio of new variables $a = k_r - k_f$, $b = -k_r(c_2 + c_3) + k_f c_1$, and $c = k_r c_2 c_3$ and rewrite Eqn. (3.59) as

$$\frac{dY_a}{dt} = aY_a^2 + bY_a + c. \tag{3.60}$$

This rewritten ODE can be solved in the following manner. Start by isolating dt on the right hand side

$$\frac{dY_a}{aY_a^2 + bY_a + c} = dt. \tag{3.61}$$

Then integrate both sides. This equation is a standard form and the solution is readily available in many standard integral tables, such as Ref. [22]. If the quantity $q = 4ac - b^2$ satisfies $q < 0$ then the solution to Eqn. (3.61) is

$$\int \frac{dY_a}{aY_a^2 + bY_a + c} = \frac{-2}{\sqrt{-q}} \tanh^{-1}\left(\frac{2cY_a + b}{\sqrt{-q}}\right). \tag{3.62}$$

The proof that $q < 0$ can be found in Ref. [19]. Doing this integration from $t = 0$ to the current time t gives

$$\frac{-2}{\sqrt{-q}} \tanh^{-1}\left(\frac{2cY_a + b}{\sqrt{-q}}\right) - \frac{-2}{\sqrt{-q}} \tanh^{-1}\left(\frac{2cY_a(t=0) + b}{\sqrt{-q}}\right) = t \tag{3.63}$$

which can be rewritten slightly as

$$\tanh^{-1}\left(\frac{2aY_a + b}{\sqrt{-q}}\right) - \tanh^{-1}\left(\frac{2aY_a(t=0) + b}{\sqrt{-q}}\right) = -\frac{\sqrt{-q}}{2}t. \tag{3.64}$$

We continue to rearrange the terms so that

$$\tanh^{-1}\left(\frac{2aY_a + b}{\sqrt{-q}}\right) = -\frac{\sqrt{-q}}{2}t + \tanh^{-1}\left(\frac{2aY_a(t=0) + b}{\sqrt{-q}}\right). \tag{3.65}$$

Note that the right most term is a constant. Let $A = \tanh^{-1}\left(\frac{2aY_a(t=0)+b}{\sqrt{-q}}\right)$ be that constant for the sake of compacting the notation. If we take the hyperbolic tangent of both sides of Eqn. (3.65) it becomes

$$\frac{2aY_a + b}{\sqrt{-q}} = \tanh\left(-\frac{\sqrt{-q}t}{2} + A\right), \tag{3.66}$$

which using the definition of hyperbolic tangent can be transformed into

$$\frac{2aY_a + b}{\sqrt{-q}} = \frac{e^{-\sqrt{q}t+2A} + 1}{e^{-\sqrt{q}t+2A} - 1}, \tag{3.67}$$

and

$$\frac{2aY_a + b}{\sqrt{-q}} = \frac{e^{2A}e^{-\sqrt{q}t} + 1}{e^{2A}e^{-\sqrt{q}t} - 1}. \tag{3.68}$$

The term e^{2A} may be cumbersome to calculate as it involves multiple exponential functions. It can be simplified as follows. First let $X = 2aY_a(t=0) + b/\sqrt{-q}$, so that

$$e^{2A} = e^{2\tanh^{-1}(X)}. \tag{3.69}$$

Note that $\tanh^{-1}(X) = \frac{1}{2}\ln\frac{1+X}{1-X}$, so Eqn. (3.69) becomes

$$e^{2A} = e^{\ln\frac{1+X}{1-X}} = \frac{1+X}{1-X}. \tag{3.70}$$

Substitute X back into Eqn. (3.70) and simplify the result to find

$$e^{2A} = -\left(\frac{\sqrt{q} + 2aY_a(t=0) + b}{-\sqrt{q} + 2aY_a(t=0) + b}\right). \tag{3.71}$$

For the sake of notational compactness later, define

$$\phi = \frac{\sqrt{q} + 2aY_a(t=0) + b}{-\sqrt{q} + 2aY_a(t=0) + b}, \tag{3.72}$$

which means that $e^{2A} = -\phi$. Substituting ϕ into Eqn. (3.68) gives

$$\frac{2aY_a + b}{\sqrt{-q}} = \frac{-\phi e^{-\sqrt{-q}t} - 1}{-\phi e^{-\sqrt{-q}t} + a}, \tag{3.73}$$

which is readily solved for Y_a

$$Y_a(t) = \frac{-1}{2a}\left(b + \sqrt{-q}\frac{1 + \phi e^{-\sqrt{-q}t}}{1 - \phi e^{-\sqrt{-q}t}}\right). \tag{3.74}$$

The equilibrium solution of Eqn. (3.74) corresponds to the limit when the time goes to infinity ($t \to \infty$):

$$\bar{Y_a} = -\frac{1}{2a}(b + \sqrt{-q}), \tag{3.75}$$

where $\bar{Y_a}$ is the equilibrium solution for Y_a. Let us define a progress variable λ such that it satisfies

$$\frac{d\lambda}{dt} = F_{ab\rightleftharpoons cd} \qquad \lambda(t=0) = 0. \tag{3.76}$$

The source term for this pair of reactions is

$$F_{ab\rightleftharpoons cd} = -k_f Y_a Y_b + k_r Y_c Y_d \tag{3.77}$$

and the differential equations governing this pair of reactions are

$$\frac{dY_a}{dt} = F_{ab\rightleftharpoons cd} = \frac{d\lambda}{dt} \tag{3.78}$$

$$\frac{dY_b}{dt} = F_{ab\rightleftharpoons cd} = \frac{d\lambda}{dt} \tag{3.79}$$

$$\frac{dY_c}{dt} = -F_{ab\rightleftharpoons cd} = -\frac{d\lambda}{dt} \tag{3.80}$$

$$\frac{dY_d}{dt} = -F_{ab\rightleftharpoons cd} = -\frac{d\lambda}{dt}. \tag{3.81}$$

If we solve for Y_a we find that

$$Y_a(t) - Y_a(t=0) = -\lambda(t). \tag{3.82}$$

Solving for the other variables returns a similar result,

$$Y_b(t) - Y_b(t=0) = -\lambda(t) \tag{3.83}$$

$$Y_c(t) - Y_c(t=0) = \lambda(t) \tag{3.84}$$

$$Y_d(t) - Y_d(t=0) = \lambda(t). \tag{3.85}$$

Just as in the $a \rightleftharpoons 2b$ example above, we have reduced the solution of several ODEs for several variables to the solution of one ODE for one variable from which we can compute the remaining quantities. If we solve for Y_a then we can use the constants c_1, c_2, and c_3 from Eqns. (3.50), (3.52 and (3.54) to compute Y_b, Y_c and Y_d,

$$Y_b(t) = Y_a(t) - c_1 \tag{3.86}$$

$$Y_c(t) = c_2 - Y_a(t) \tag{3.87}$$

$$Y_d(t) = c_3 - Y_a(t). \tag{3.88}$$

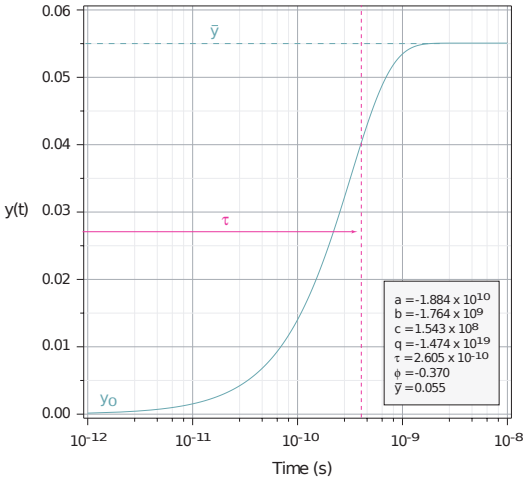


Figure 3.18: The time evolution of Eqn. (3.74) for fixed constants a, b, c. τ is the timescale to reach equilibrium and $\bar{y}$ is the equilibrium value for $Y_a(t)$ with an assumed initial value of $Y_a(t=0) = 0$. From Ref. [18].

Since we can solve the system of ODEs for the $ab \rightleftharpoons cd$ reaction pair by solving a single ODE for a single variable, we can define a characteristic timescale,

$$\tau = \frac{1}{\sqrt{-q}}, \tag{3.89}$$

for the reaction pair to reach equilibrium. Figures 3.18 and 3.19 show the time behavior of the solution (Eqn. (3.74)) for the ODES in Eqns. (3.78-3.81). Notice how the exponential term displayed in Figure 3.19 goes to 0 very quickly, and is rather negligible after time τ. Likewise, the solution in Figure 3.18 saturates to the equilibrium value $\bar{Y}_a$ quickly after reaching τ. Our criteria in Eqn. (3.22) is still valid here, however we expect that

$$\frac{|Y_i(t) - \bar{Y}_i|}{\bar{Y}_i} < \varepsilon_i, \tag{3.90}$$

where $Y_i(t)$ is the actual population of species i, $\bar{Y}_i$ is the equilibrium value computed with Eqn. (3.74) and ε_i is an arbitrary user-specified small quantity, will provide a test which is approximately as good.

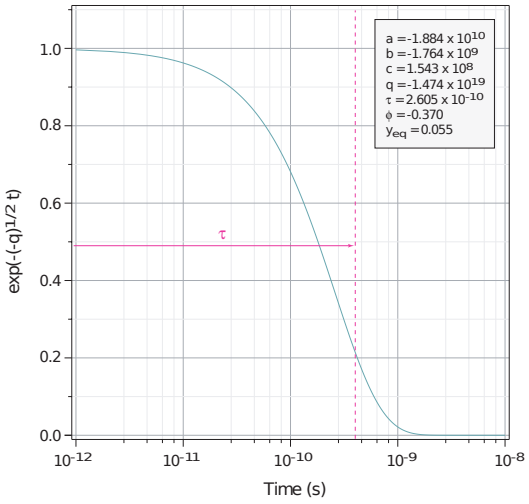


Figure 3.19: Time evolution of the exponential portion of Eqn. (3.74) for fixed constants a, b, c. τ is the timescale to reach equilibrium. From Ref. [18]

3.3.2 Reaction Vectors

In a realistic network, there may be thousands or even tens of thousands of reactions participating in a nuclear network consisting of hundreds or thousands of isotopes. To implement the partial equilibrium approximation outlined in the previous section each of those reactions must be examined to determine whether they are in equilibrium at every timestep. This is a considerable amount of detailed bookkeeping which requires a systematic formalism to simplify and regulate. David Mott in his thesis outlines just such a formalism, which is adapted here. In short, we will devise a scheme where we create an analog between the nuclear reaction network and a linear vector space.

Start by defining a vector to represent the composition of the network at any given time

$$\vec{y} = (y_1, y_2, ..., y_n), \tag{3.91}$$

where the y_n are the individual populations for each species n in the nuclear reaction network. The vector $\vec{y}$ is in an n-dimensional vector space Φ, where Φ is the vector space which contains all possible composition vectors for the nuclear reaction network. We use a small y here as opposed to a capital Y because these elements of $\vec{y}$ can be

any numbers as long as they remain proportional to the populations Y_i. We can represent any reaction in the network as

$$\sum_{i=1}^{n} a_i y_i \rightleftharpoons \sum_{i=1}^{n} b_i y_i \tag{3.92}$$

where a_i and b_i are a set of coefficients that define how the reaction affects the populations of the nuclei. For example, consider the simple network of ^{4}He, ^{12}C, ^{16}O, and ^{20}Ne. Order the elements of the composition vector such that

$$\vec{y} = \Big(y(^4\mathrm{He}), y(^{12}\mathrm{C}), y(^{16}\mathrm{O}), y(^{20}\mathrm{Ne}) \Big). \tag{3.93}$$

One reaction in the network is $^4\mathrm{He} + {}^{12}\mathrm{C} \rightarrow {}^{16}\mathrm{O}$. For this reaction the coefficients are

$$a = \{1, 1, 0, 0\} \qquad b = \{0, 0, 1, 0\}. \tag{3.94}$$

From these coefficients we can define a vector $\vec{r} \in \Phi$ which represents how the reaction can alter the composition $\vec{y}$. $\vec{r}$ has the form

$$\vec{r} = (b_1 - a_1, ..., b_n - a_n). \tag{3.95}$$

For our example reaction $\vec{r}$ is

$$\vec{r} = (-1, -1, 1, 0). \tag{3.96}$$

This shows that every time the example reaction occurs, it subtracts one atom from the ^{4}He and ^{12}C populations and add one to ^{16}O. Every reaction in the network can have its action parameterized this way. The final composition vector is written as a sum of the initial composition vector $\vec{y}_0$ and a linear sum of reaction vectors

$$\vec{y} = \vec{y}_0 + \sum_{i=1}^{m} \alpha_i \vec{r}_i \tag{3.97}$$

where the α_i are scalar quantities representing how many times reaction i has occurred in a system with m reactions in it.

3.3.3 Conservation Laws

With derived conserved scalar quantities, the reaction vector formalism is allowed to define conserved vector quantities which are in the vector space Φ. Define a time-independent vector $\vec{c} = (c_1, c_2, ..., c_n) \in \Phi$ which is orthogonal to every $\vec{r}_i$ in the network. From Eqn. (3.97) we have

$$\vec{y} - \vec{y}_0 = \sum_{i=1}^{m} \alpha_i \vec{r}_i. \tag{3.98}$$

Take the inner product of $\vec{c}$ against both sides to get

$$\vec{c} \cdot (\vec{y} - \vec{y}_0) = \vec{c} \cdot \sum_{i=1}^{m} \alpha_i \vec{r}_i. \tag{3.99}$$

Recall that $\vec{c}$ is orthogonal to any individual $\vec{r}_i$, then by the properties of the Euclidean inner product (see any good linear algebra text for the details of inner products, such as Ref. [23])

$$\vec{c} \cdot \vec{r}_i = 0. \tag{3.100}$$

Then since we can expand the sum of the $\alpha_i \vec{r}_i$ as

$$\sum_{i=1}^{m} \alpha_i \vec{r}_i = \alpha_1 \vec{r}_1 + \alpha_2 \vec{r}_2 + ... + \alpha_m \vec{r}_m, \tag{3.101}$$

the right hand side of Eqn. (3.99) becomes

$$\vec{c} \cdot \sum_{i=1}^{m} \alpha_i \vec{r}_i = \vec{c} \cdot (\alpha_1 \vec{r}_1 + \alpha_2 \vec{r}_2 + ... + \alpha_m \vec{r}_m) \tag{3.102}$$

which then becomes

$$\vec{c} \cdot \sum_{i=1}^{m} \alpha_i \vec{r}_i = \vec{c} \cdot \alpha_1 \vec{r}_1 + \vec{c} \cdot \alpha_2 \vec{r}_2 + ... + \vec{c} \cdot \alpha_m \vec{r}_m. \tag{3.103}$$

Since scalars are unaffected by inner products this gives

$$\vec{c} \cdot \sum_{i=1}^{m} \alpha_i \vec{r}_i = \alpha_1 (\vec{c} \cdot \vec{r}_1) + \alpha_2 (\vec{c} \cdot \vec{r}_2) + ... + \alpha_m (\vec{c} \cdot \vec{r}_m), \tag{3.104}$$

which means that since $\vec{c}$ is orthogonal to every r_i that

$$\vec{c} \cdot \sum_{i=1}^{m} \alpha_i \vec{r}_i = 0. \tag{3.105}$$

Eqn. (3.99) says that

$$\vec{c} \cdot \vec{y} = \vec{c} \cdot \vec{y}_0, \tag{3.106}$$

or alternately

$$\sum_{i=1}^{m} c_i y_i = \sum_{i=1}^{m} c_i y_0^i, \tag{3.107}$$

where the index on the elements of y_0 has been moved to a superscript to reduce notational clutter. This means that vector $\vec{c}$ gives a linear combination of species abundances which are completely invariant under the reactions in the network as defined by the reaction vectors $\vec{r}_i$. Eqn. (3.107) is a conservation law for the nuclear reaction network which arises solely because of the structure of the network and not from any dynamics at play. For example, if $c_i = A_i$, where A_i are the atomic masses, then Eqn. (3.107) is a statement that mass is conserved in the system.

3.3.4 Reaction Groups

In the commonly used astrophysical nuclear reaction rate library REACLIB, individual reactions are classified into one of eight different categories, depending on the number of isotopes involved either on the input or output side of the reaction.

Table 3.1: REACLIB Library Reaction Classes

Class	Reaction	Description or Example
1	$a \rightarrow b$	β decays
2	$a \rightarrow b + c$	Photodisintegration to a lighter nucleus $+\alpha$
3	$a \rightarrow b + c + d$	Example: $^{12}\mathrm{C} \rightarrow 3\alpha$
4	$a + b \rightarrow c$	Capture reactions
5	$a + b \rightarrow c + d$	Exchange reactions
6	$a + b \rightarrow c + d + e$	$^{2}\mathrm{H} + {}^{7}\mathrm{Be} \rightarrow {}^{1}\mathrm{H} + 2^{4}\mathrm{He}$
7	$a + b \rightarrow c + d + e + f$	$^{3}\mathrm{He} + {}^{7}\mathrm{Be} \rightarrow 2^{1}\mathrm{H} + 2^{4}\mathrm{He}$
8	$a + b + c \rightarrow d$	Effective 3-body reactions
8	$a + b + c \rightarrow d + e$	Effective 3-body reactions

Table 3.1 lists the standard REACLIB classifications, illustrates the structure of the reactions in each reaction class, and gives a short description of, or example from, that class. Reaction Class 8 is split into two lines in the table because it encompasses reactions with two separate structures, though in both cases they are reactions which require three nuclei on the input-side. Table 3.2 maps the REACLIB classes onto five classes of reaction groups, which are opposing pairs of reactions as demanded by the PE machinery developed in previous sections.

REACLIB class 8 reactions appear in both reaction group C and E because class 8 encompasses two different sets of output products. It is interesting to note that reaction group D is composed of only class 5 reactions paired with other class 5 reactions. Reaction class 7 doesn't appear in Table 3.2 as there are no important astrophysical reactions which require four bodies as inputs, however such reactions can be present in a system and can interfere with the system reaching equilibrium. Reactions which are catalytic in nature, that is reactions which have one of the species on both sides of the reaction, don't always fit these observations.

The PE formalism developed above is well-suited to handle the majority of astrophysical reactions. An exception are those reactions which are three-body reactions from reaction class 8. For those the differential equation governing them has the form

$$\frac{dY}{dt} = \alpha Y^3 + \beta Y^2 + \gamma Y + \epsilon. \tag{3.108}$$

Table 3.2: Reaction Group Classification

Class	Reaction Pair	Corresponding REACLIB Classes
A	$a \rightleftharpoons b$	1 with 1
B	$a + b \rightleftharpoons c$	2 with 4
C	$a + b + c \rightleftharpoons d$	3 with part of 8
D	$a + b \rightleftharpoons c + d$	5 with 5
E	$a + b \rightleftharpoons c + d + e$	6 with part of 8

To employ the machinery developed above, we can choose to approximate that

$$Y(t)^3 = Y(t_{\text{initial}})Y(t)^2, \tag{3.109}$$

where $Y(t_{\text{initial}})$ is the value of $Y(t)$ evaluated at the beginning of the timestep. As justification for this approximation, when near equilibrium conditions, then $Y(t_{\text{initial}}) \sim Y(t)$. Let $a = \alpha Y(t_{\text{initial}}) + \beta$, $b = \gamma$, and $c = \epsilon$. Substituting those into Eqn. (3.108) gives the familiar second-order equation from Eqn. (3.60) whose solution we already know from Eqn. (3.74). Thus we have turned a difficult equation with a cubic term into a problem we've already solved.

Recall that in earlier examples we had source terms of the form

$$F_{ab \rightleftharpoons cd} = k_f Y_a Y_b - k_r Y_c Y_d. \tag{3.110}$$

If the reaction pair $ab \rightleftharpoons cd$ is in equilibrium, then the forward and reverse reactions have fluxes that are approximately equal

$$F_{ab \rightleftharpoons cd} = F^+ - F^- = k_f Y_a Y_b - k_r Y_c Y_d = 0. \tag{3.111}$$

Thus equilibrium conditions imply a constraint of the form

$$k_f Y_a Y_b = k_r Y_c Y_d \tag{3.112}$$

or

$$\frac{Y_a Y_b}{Y_c Y_d} = \frac{k_r}{k_f}. \tag{3.113}$$

If detailed balance is applied to the above equation, then the equations of quasi-equilibrium from Ref [20] are recovered.

It is easy to generalize from this case for the various reaction groups, but the constraint always has the form of a ratio of products of Y_i on one side and rate constants k on the other side. In general for a reaction with reaction vector $\vec{r}$ such that

$$\sum_{i=1}^{n} a_i y_i = \sum_{i=1}^{n} b_i y_i \tag{3.114}$$

equilibrium implies a constant of the form

$$\prod_{i=1}^{n} y_i^{(b_i - a_i)} = K \tag{3.115}$$

where is the time and temperature dependent ratio of rate parameters $K = k_r/k_f$.

3.3.5 Summary of the Partial Equilibrium Approach

In this section the two pieces needed to implement the PE approximation are presented: (1) an overview of the approach, and (2) the various equations derived from the previous several sections needed to test for equilibrium and compute the equilibrium value. Applying the PE method alongside the asymptotic flux-limiting algorithm combats stiffness with as many tools as possible. This approach will be a two-step process, (1) applying the asymptotic approximation and an explicit forward differencing technique to all populations in the network, and (2) using the partial equilibrium framework to remove source terms for reactions which are in equilibrium from the differential equations.

Before we present an outline of how to implement a combined partial equilibrium and asymptotic approximation scheme, we summarize the equations and constants

for reactions group classes A-E (see Table 3.2) that are derived with techniques outlined in the preceding several sections. The summary below of equations is specific to astrophysical nuclear reaction networks and is based upon the classification and parameterization of reactions according to the REACLIB library.

Reaction Group Class A $(a \rightleftharpoons b)$

Source term: $\frac{dy_a}{dt} = -k_f y_a + k_r y_b$

Constraints: $y_a + y_b \equiv c_1 = y_a^0 + y_b^0$

Equation: $\frac{dy_a}{dt} = by_a + c \quad (b = -k_f c = k_r)$

Solution: $y_a(t) = y_a^0 e^{bt} - \frac{c}{b}$

Equilibrium solution: $\bar{y_a} = -\frac{c}{b} = \frac{k_r}{k_f}$

Equilibrium timescale: $\tau = \frac{1}{b} = \frac{1}{k_f}$

Equilibrium tests: $\frac{|y_i - \bar{y_i}|}{\bar{y_i}} < \varepsilon_i \ (i = a, b)$

Equilibrium constraint: $\frac{y_a}{y_b} = \frac{k_r}{k_f}$

Other variables: $y_b = c_1 - y_a$

Progress variable: $\lambda \equiv y_a^0 - y_a \quad y_a = y_a^0 - \lambda \quad y_b = y_b^0 + \lambda$

Reaction Group Class B $(a + b \rightleftharpoons c)$

Source term: $\frac{dy_a}{dt} = -k_f y_a y_b + k_r y_c$

Constraints: $y_b - y_a \equiv c_1 = y_b^0 - y_a^0 \quad y_b + y_c \equiv c_2 = y_b^0 + y_c^0$

Equation: $\frac{dy_a}{dt} = ay_a^2 + by_a + c \quad (a = -k_f b = -(c_1 k_f + k_b)c = k_r(c_2 - c_1))$

Solution: $y_a(t) = \frac{-1}{2a}\left(b + \sqrt{-q}\frac{1+\phi e^{-\sqrt{-q}t}}{1-\phi e^{-\sqrt{-q}t}}\right)$

Equilibrium solution: $\bar{y_a} = -\frac{1}{2a}(b + \sqrt{-q})$

Equilibrium timescale: $\tau = \frac{1}{\sqrt{-q}}$

Equilibrium tests: $\frac{|y_i - \bar{y_i}|}{\bar{y_i}} < \varepsilon_i \ (i = a, b, c)$

Equilibrium constraint: $\frac{y_a y_b}{y_c} = \frac{k_r}{k_f}$

Other variables: $y_b = c_1 + y_a \quad y_c = c2 - y_b$

Progress variable: $\lambda \equiv y_a^0 - y_a \quad y_a = y_a^0 - \lambda \quad y_b = y_b^0 - \lambda$

Reaction Group Class C $(a + b + c \rightleftharpoons d)$

Source term: $\frac{dy_a}{dt} = -k_f y_a y_b y_c + k_r y_d$

Constraints: $y_a - y_b \equiv c_1 = y_a^0 - y_b^0$

$y_a - y_c \equiv c_2 = y_a^0 - y_c^0$

$\frac{1}{3}(y_a + y_b + y_c) + y_d \equiv c_3 = \frac{1}{3}(y_a^0 + y_b^0 + y_c^0) + y_d$

Equation: $\frac{dy_a}{dt} = ay_a^2 + by_a + c$

$a = -k_f y_a^0 + k_f(c_1 + c_2) \;\; b = -(k_f c_1 c_2 + k_r) \;\; c = (c_3 + \frac{1}{3}c_1 + \frac{1}{3}c_2)k_r$

Solution: $y_a(t) = \frac{-1}{2a}\left(b + \sqrt{-q}\frac{1+\phi e^{-\sqrt{-q}t}}{1-\phi e^{-\sqrt{-q}t}}\right)$

Equilibrium solution: $\bar{y_a} = -\frac{1}{2a}(b + \sqrt{-q})$

Equilibrium timescale: $\tau = \frac{1}{\sqrt{-q}}$

Equilibrium tests: $\frac{|y_i - \bar{y_i}|}{\bar{y_i}} < \varepsilon_i \;\; (i = a, b, c, d)$

Equilibrium constraint: $\frac{y_a y_b y_c}{y_d} = \frac{k_r}{k_f}$

Other variables: $y_b = y_a - c1 \;\; y_c = y_a - c_2 \;\; y_d = c_3 - y_a + \frac{1}{3}(c_1 + c_2)$

Progress variable: $\lambda \equiv y_a^0 - y_a \;\; y_a = y_a^0 - \lambda \;\; y_b = y_b^0 - \lambda$

$y_c = y_c^- - \lambda \;\; y_d = \lambda + y_d^0$

Reaction Group Class D $(a + b \rightleftharpoons c + d)$

Source term: $\frac{dy_a}{dt} = -k_f y_a y_b + k_r y_c y_d$

Constraints: $y_a - y_b \equiv c_1 = y_a^0 - y_b^0$

$y_a + y_c \equiv c_2 = y_a^0 + y_c^0$

$y_a + y_d \equiv c_3 = y_a^0 + y_d^0$

Equation: $\frac{dy_a}{dt} = ay_a^2 + by_a + c$

$a = k_f - k_r \;\; b = -k_r(c_2 + c_3) + k_f c_1 \;\; c = k_r c_2 c_3$

Solution: $y_a(t) = \frac{-1}{2a}\left(b + \sqrt{-q}\frac{1+\phi e^{-\sqrt{-q}t}}{1-\phi e^{-\sqrt{-q}t}}\right)$

Equilibrium solution: $\bar{y_a} = -\frac{1}{2a}(b + \sqrt{-q})$

Equilibrium timescale: $\tau = \frac{1}{\sqrt{-q}}$

Equilibrium tests: $\frac{|y_i - \bar{y_i}|}{\bar{y_i}} < \varepsilon_i \;\; (i = a, b, c, d)$

Equilibrium constraint: $\frac{y_a y_b}{y_c y_d} = \frac{k_r}{k_f}$

Other variables: $y_b = y_a - c_1 \;\; y_c = c_2 - y_a \;\; y_d = c_3 - y_a$

Progress variable: $\lambda \equiv y_a^0 - y_a \;\; y_a = y_a^0 - \lambda \;\; y_b = y_b^0 - \lambda$

$y_c = y_c^0 + \lambda \;\; y_d = y_d^0 + \lambda$

Reaction Group Class E $(a + b \rightleftharpoons c + d + e)$

Source term: $\frac{dy_a}{dt} = -k_f y_a y_b + k_r y_c y_d y_e$

Constraints: $y_a + \frac{1}{3}(y_c + y_d + y_e) \equiv c_1 = y_a^0 + \frac{1}{3}(y_c^0 + y_d^0 + y_e^0)$

$y_a - y_b \equiv c_2 = y_a^0 - y_b^0 \;\; y_c - y_d \equiv c_3 = y_c^0 - y_d^0 \;\; y_c - y_e \equiv c_4 = y_c^0 - y_e^0$

Equation: $\frac{dy_a}{dt} = a y_a^2 + b y_a + c$

$a = (3c_1 - y_a^0)k_r - k_f \;\; b = c_2 k_f - (\alpha\beta + \alpha\gamma + \beta\gamma)k_r \;\; c = k_r \alpha\beta\gamma$

$\alpha = c_1 + \frac{1}{3}(c_3 + c_4) \;\; \beta = c_1 - \frac{2}{3}c_3 + \frac{1}{3}c_4 \;\; \gamma = c_1 + \frac{1}{3}c_3 - \frac{2}{3}c_4$

Solution: $y_a(t) = \frac{-1}{2a}\left(b + \sqrt{-q}\frac{1+\phi e^{-\sqrt{-q}t}}{1-\phi e^{-\sqrt{-q}t}}\right)$

Equilibrium solution: $\bar{y_a} = -\frac{1}{2a}(b + \sqrt{-q})$

Equilibrium timescale: $\tau = \frac{1}{\sqrt{-q}}$

Equilibrium tests: $\frac{|y_i - \bar{y_i}|}{\bar{y_i}} < \varepsilon_i \;\; (i = a, b, c, d, e)$

Equilibrium constraint: $\frac{y_a y_b}{y_c y_d y_e} = \frac{k_r}{k_f}$

Other variables: $y_b = y_a - c_2 \;\; y_c = \alpha - y_a \;\; y_d = \beta - y_a$

Progress variable: $\lambda \equiv y_a^0 - y_a \;\; y_a = y_a^0 - \lambda \;\; y_b = y_b^0 - \lambda$

$y_c = y_c^0 + \lambda \;\; y_d = y_d^0 + \lambda \;\; y_e = y_e^0 + \lambda$

First, compute for all reactions the source terms and derivatives. Then identify which reaction pairs are in equilibrium and set the source terms for the reactions which belong to that reaction pair identically to zero; note which isotopes have their population affected by doing this. Choose a timestep Δt and determine which species in the network are asymptotic. Evolve the non-asymptotic populations with explicit forward differencing, such as FLFD; evolve populations which satisfy $k\Delta t$ with the asymptotic approximation. Finally for all those isotopes which had reactions which were in equilibrium, assume that the non-equilibrium reactions have disturbed

the populations away from equilibrium during the timestep and then restore the equilibrium, with the constraint that the total mass of the system is conserved.

3.3.6 Restoring Equilibrium

The partial equilibrium method gives special treatment to isotopes affected by equilibrated reactions. After the effect of non-equilibrated reaction pairs is computed, those isotopes in reaction groups that are in equilibrium need to have that equilibrium restored since the equilibrium of a reaction pair will generally be slightly disturbed during a timestep by reactions in the network not in equilibrium. Our first implementation of a restoration scheme involved performing Newton-Raphson iteration to reimpose equilibrium abundance ratios. A second method was similar, using Newton-Raphson iterations to set all of the abundances for these isotopes to equilibrium values. Both of these approaches have the downside of introducing iterations on Jacobian matrixes $\mathbf{J} \equiv \frac{\partial \mathbf{F}}{\partial \mathbf{Y}}$, where $\mathbf{F}$ is a vector containing the fluxes and $\mathbf{Y}$ is a vector containing the abundances for the populations. To avoid the cost of these Jacobian based schemes, we have investigated simpler schemes to restore equilibrium for an isotope, by simply averaging the equilibrium population values for that isotope from every involved reaction group.

The first sort of averaging to try is also the simplest, just a flat averaging scheme where each reaction group contributes equally to the re-imposition of equilibrium on the disturbed populations. Weighted averages of the equilibrium populations from each reaction group are another approach, and both weighting based on the net flux and on the mass fraction were investigated. Finally we explored extremely simple schemes where either the equilibrium value closest to or furthest away from the value for the population returned from the numerical integration was used.

Let $\bar{Y}_i$ here represent the averaged population of species i which we use to reimpose equilibrium on the system. Then if i participates in M equilibrated reaction groups,

its restored equilibrium value is

$$\bar{Y}_i = \frac{1}{M} \sum_j^M \bar{Y_i^j} \tag{3.116}$$

where $\bar{Y_i^j}$ is the equilibrium solution computed for species i in reaction group j. As a note, while the formalism developed previously identifies pairs of forward and inverse reactions, as we use reaction rate libraries which sometimes treat the resonant and non-resonant components of each reaction as separate "reactions" these reaction groups can commonly have four "reactions" as members though these represent two physical reactions. The equilibrium constraints for each reaction group class are

$$\text{Reaction Group Class A:} \quad \frac{y_a}{y_b} = \frac{k_r}{k_f}, \tag{3.117}$$

$$\text{Reaction Group Class B:} \quad \frac{y_a y_b}{y_c} = \frac{k_r}{k_f}, \tag{3.118}$$

$$\text{Reaction Group Class C:} \quad \frac{y_a y_b y_c}{y_d} = \frac{k_r}{k_f}, \tag{3.119}$$

$$\text{Reaction Group Class D:} \quad \frac{y_a y_b}{y_c y_d} = \frac{k_r}{k_f}, \tag{3.120}$$

$$\text{Reaction Group Class E:} \quad \frac{y_a y_b}{y_c y_d y_e} = \frac{k_r}{k_f}. \tag{3.121}$$

From which we expect that each Y_i^j for any species i should be close to each other, but not exactly the same. Making the constraint more stringent with a smaller ε_j can reduce the difference between the Y_i^j. The averaging scheme in Eqn. (3.116) will be referred to as "flat averaging" in this discussion.

The weighted schemes are similar to the above. For the flux-weighted scheme, the restored equilibrium value has the form

$$\bar{Y}_i = \sum_j^M \bar{Y_i^j} f_i^j, \tag{3.122}$$

where f_i^j is the flux normalized so that $f_i^j = F_i^j / \sum_j^M F_i^j$ and the F_i^j are the largest component of the fluxes for species i in reaction group j. Recall that we are only imposing equilibrium to an arbitrary accuracy with the constraint ε_j. This means that the fluxes in either direction (forward or backward) are not completely equal to each other. We take the larger of the forward and backward flux components and assign that to F_i^j. This attempts to treat those reactions which have the largest derivatives (i.e. the most population being moved to and from) as more important than less active reactions; though the net effect of any of this activity should be small since the reaction groups being treated with PE are in or near equilibrium.

The mass-weighted scheme is similar, with a restored equilibrium value of

$$\bar{Y}_i = \sum_j^M \bar{Y}_i^j x_i^{(j)}. \tag{3.123}$$

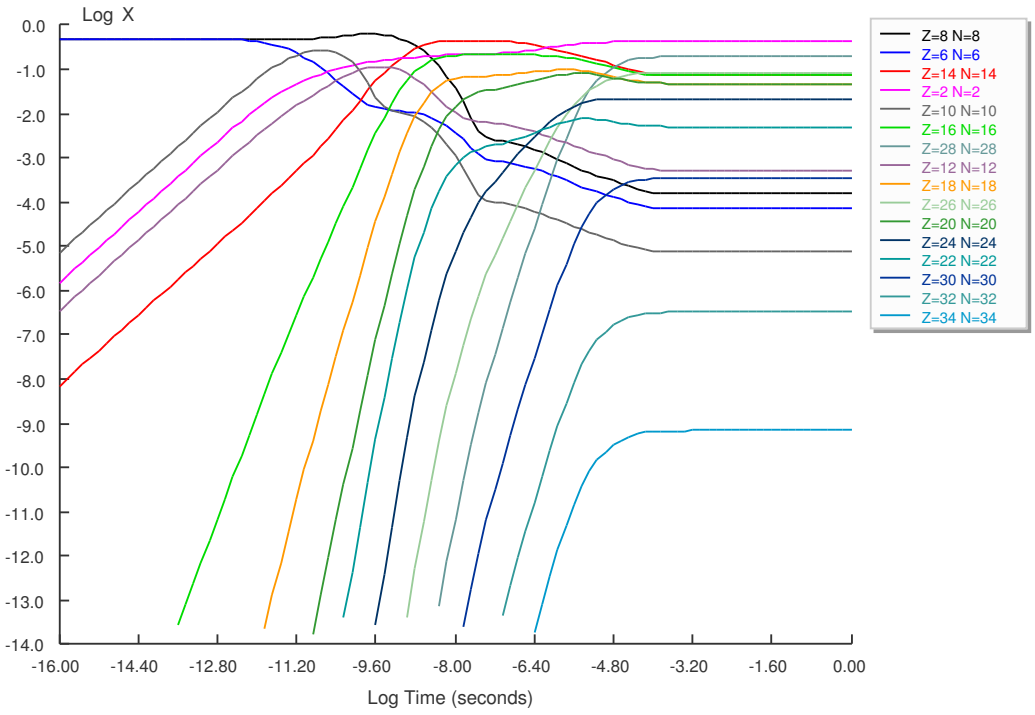
Let $X^{(j)}$ be the total sum of the mass fractions for the n species participating in reaction group j, so that $X^{(j)} = \sum_{i=1}^n X_i$. Further, define X_i^{total} as the sum of the masses of all reaction groups which affect isotope i, so if M reaction groups affect isotope i it becomes $X_i^{\text{total}} = \sum_j^M X^{(j)}$. Next, define our normalized reaction group mass fractions for isotope i as $x^{(j)} = X^{(j)} / X_i^{\text{total}}$. The idea behind this case is to allow those reactions which involve larger populations to skew the equilibrium of restoration more heavily towards their value. We argued during the development of the FLFD method that larger populations were generally more important for the system and that smaller populations are less important for the accuracy of the system. By analogy to that, it may be that the reaction groups which affect larger populations in the system at equilibrium should take priority over small populations.

The last two simple restoration schemes discussed here replace the averaging step with the simple choice of choosing either the equilibrium value closest to or furthest away from the results returned from the equilibrium value. If the idea that the $\bar{Y}_i^j$ should be close to the same across the different reaction groups j is true, it may be the case that the choice of which way to restore equilibrium for population i is not

important as long as the $\bar{Y}_i$ chosen is somewhere in or near the range of values of equilibrium $\bar{Y}_i^j$ or the numerical solution. These two ways of choosing $\bar{Y}_i$ are included to test that idea.

For each of the first three schemes, the flat average (Figure 3.20) and the two weighted averages (Figures 3.21 and 3.22), the system is well behaved. The flat average and mass-weighted averaging scheme produce results and take timesteps with curves that lie on top of each other. The flat average and the flux-weighted average disagree slightly. From $\log t = -6$ to $\log t = -4$, the flux-weighted curve begins to deviate from the flat average as shown in Figure 3.23. In this time range, flux-weighted averaging results in differences of about 20% from the flat average for populations of isotopes such as ^{12}C and ^{16}O. It is not apparent from the timestep plot shown in Figure 3.21, but the flux-weighted scheme takes 39% more timesteps as the flat average: 3979 timesteps for the flat average compared to 5527 timesteps for the flux-weighted average. The mass-weighted average took 3977 timesteps total. The final two averaging schemes, shown in Figures 3.24 and 3.25, give such blatantly bad results that we will not discuss them further. The basic arithmetic mean, what we called the flat average, equals or beats the other choices for restoring equilibrium that were tested.

The strikingly different result between the mass-weighted, flux-weight, and flat averaging schemes when compared to the simpler minimum and maximum scheme is difficult to explain. The typical spread of $\bar{Y}_i^j$ for a given species i is only on the order of a few percent. An arbitrary example from a recent calculation has a $Y_{^4\mathrm{He}}$ from the numeric integration of $Y_{^4\mathrm{He}}^{\mathrm{numeric}} = 5.062 \times 10^{-4}$. The largest $\bar{Y}_{^4\mathrm{He}}^{\bar{j}}$ is for the reaction group $^4\mathrm{He} + {}^{24}\mathrm{Mg} \rightleftharpoons {}^{28}\mathrm{Si}$ ($Y_{^4\mathrm{He}}^{\mathrm{biggest}} = 5.0745 \times 10^{-4}$) and the smallest is for $^4\mathrm{He} + {}^{28}\mathrm{Si} \rightleftharpoons {}^{32}\mathrm{S}$ ($Y_{^4\mathrm{He}}^{\mathrm{smallest}} = 4.9878 \times 10^{-4}$). The flat average for the four equilibrated reaction groups contributing to ^{4}He is $Y_{^4\mathrm{He}}^{\mathrm{flat}} = 5.0209 \times 10^{-4}$. The largest and smallest values are thus only 1.06% different from the mean, and the largest value is only 1.74% larger than the smallest equilibrium $Y_{^4\mathrm{He}}^j$. Spreads of this scale, on the 1% level are fairly typical.



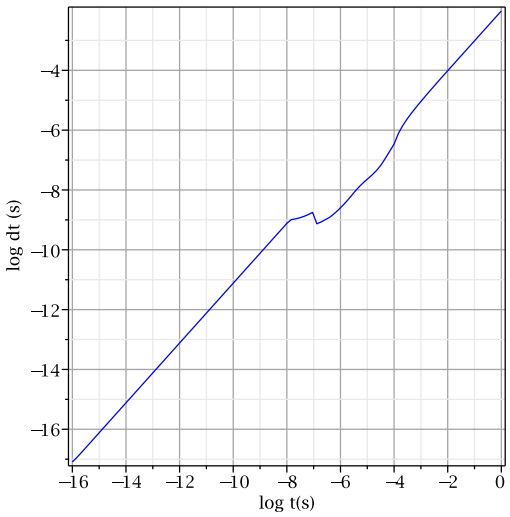


Figure 3.20: Time evolution of mass fractions (top) for an alpha network using PE with flat averaging and the timestep dt taken (bottom).

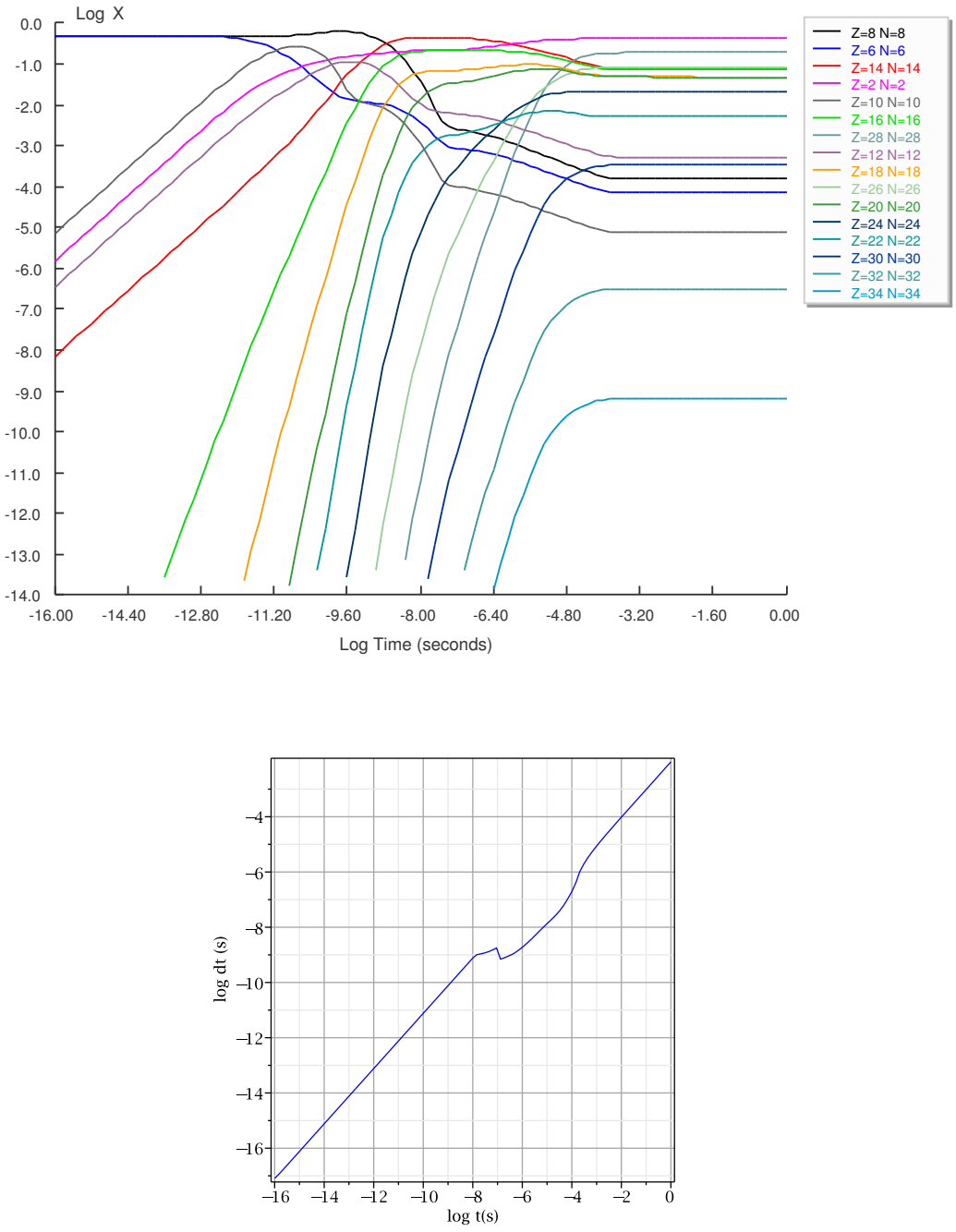


Figure 3.21: Time evolution of mass fractions (top) for an alpha network using PE with flux-weighted averaging and the timestep dt taken (bottom).

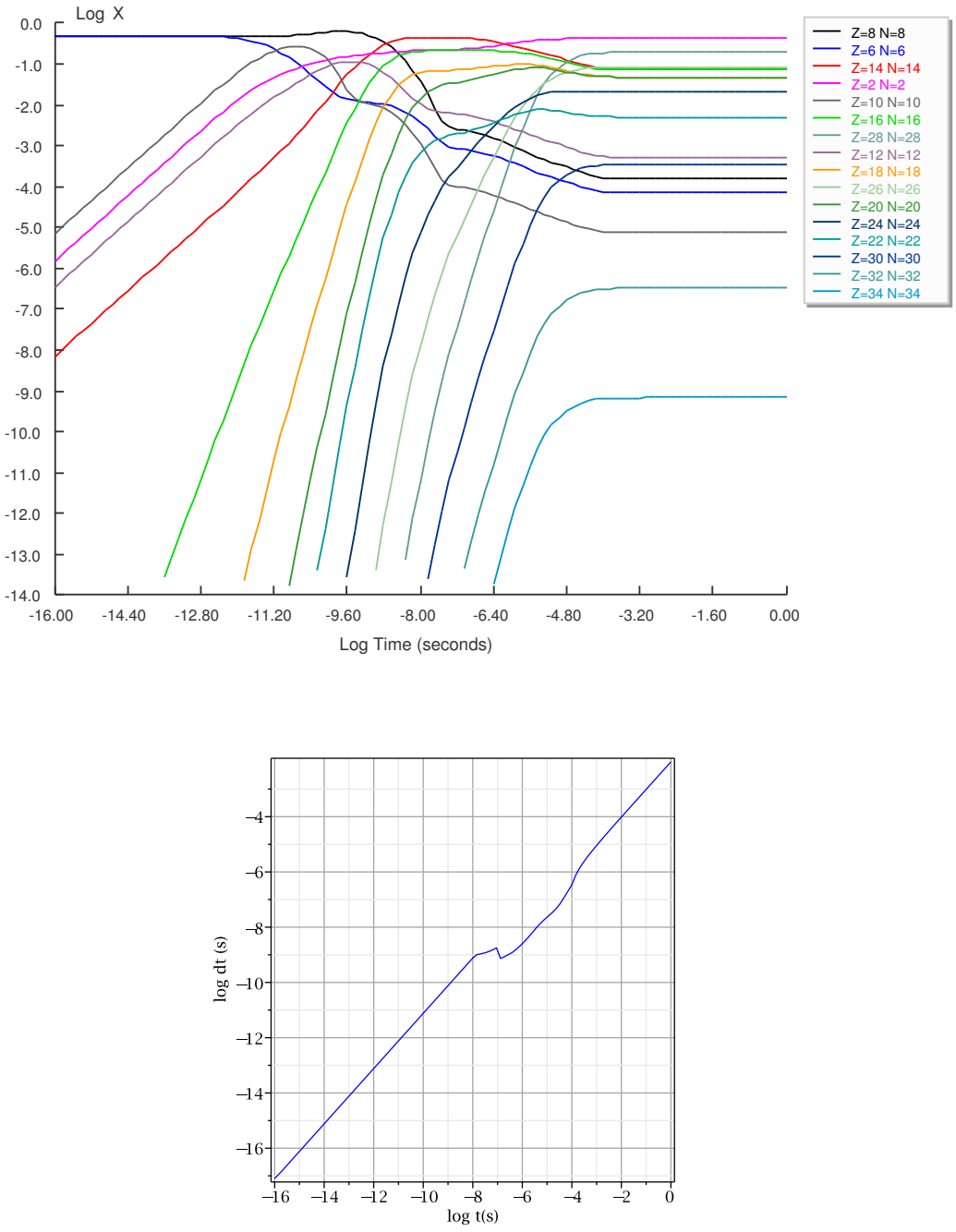


Figure 3.22: Time evolution of mass fractions (top) for an alpha network using PE with mass-weighted averaging and the timestep dt taken (bottom).

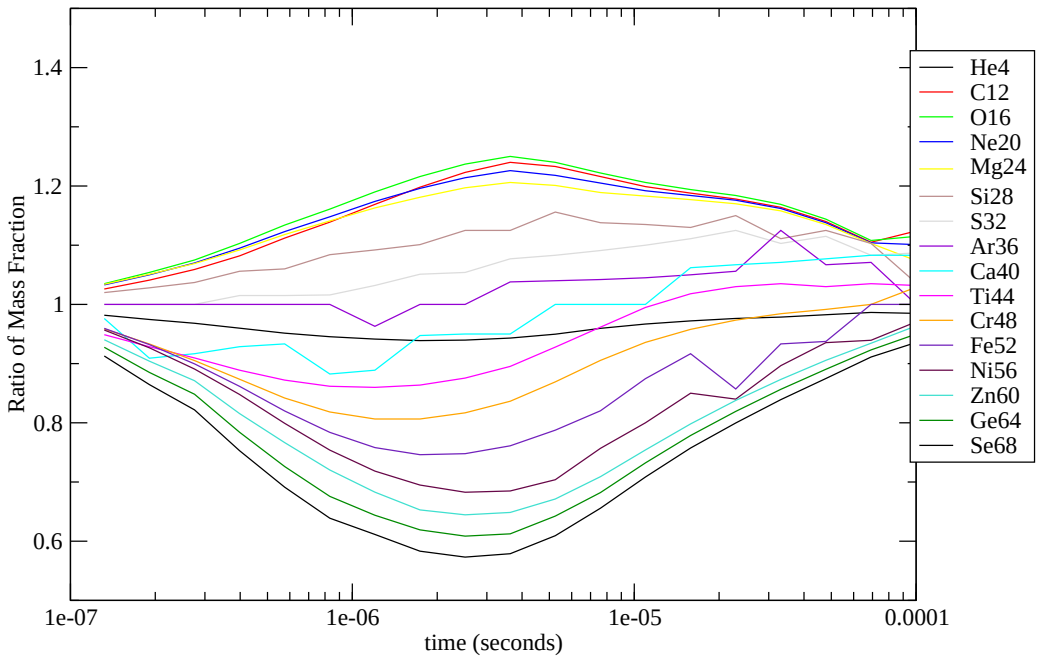


Figure 3.23: The ratio of mass fractions for the flux-weighted average and flat average ($X_{\text{flux-weighted}}/X_{\text{flat}}$) for the isotopes in the alpha network.

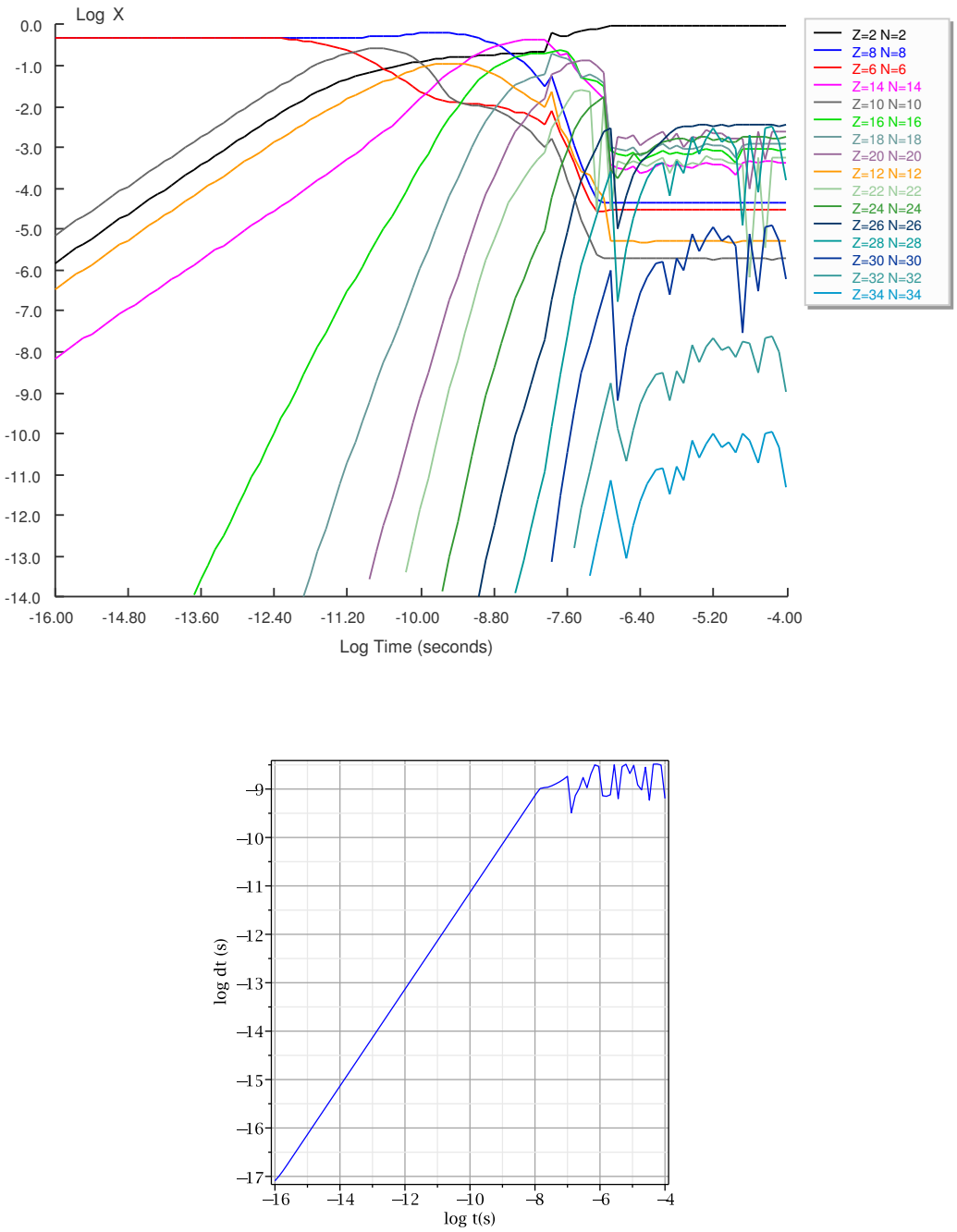


Figure 3.24: Time evolution of mass fractions (top) for an alpha network using PE choosing the equilibrium value closest to the numerical integration's result for reimposing equilibrium and the timestep dt taken (bottom).

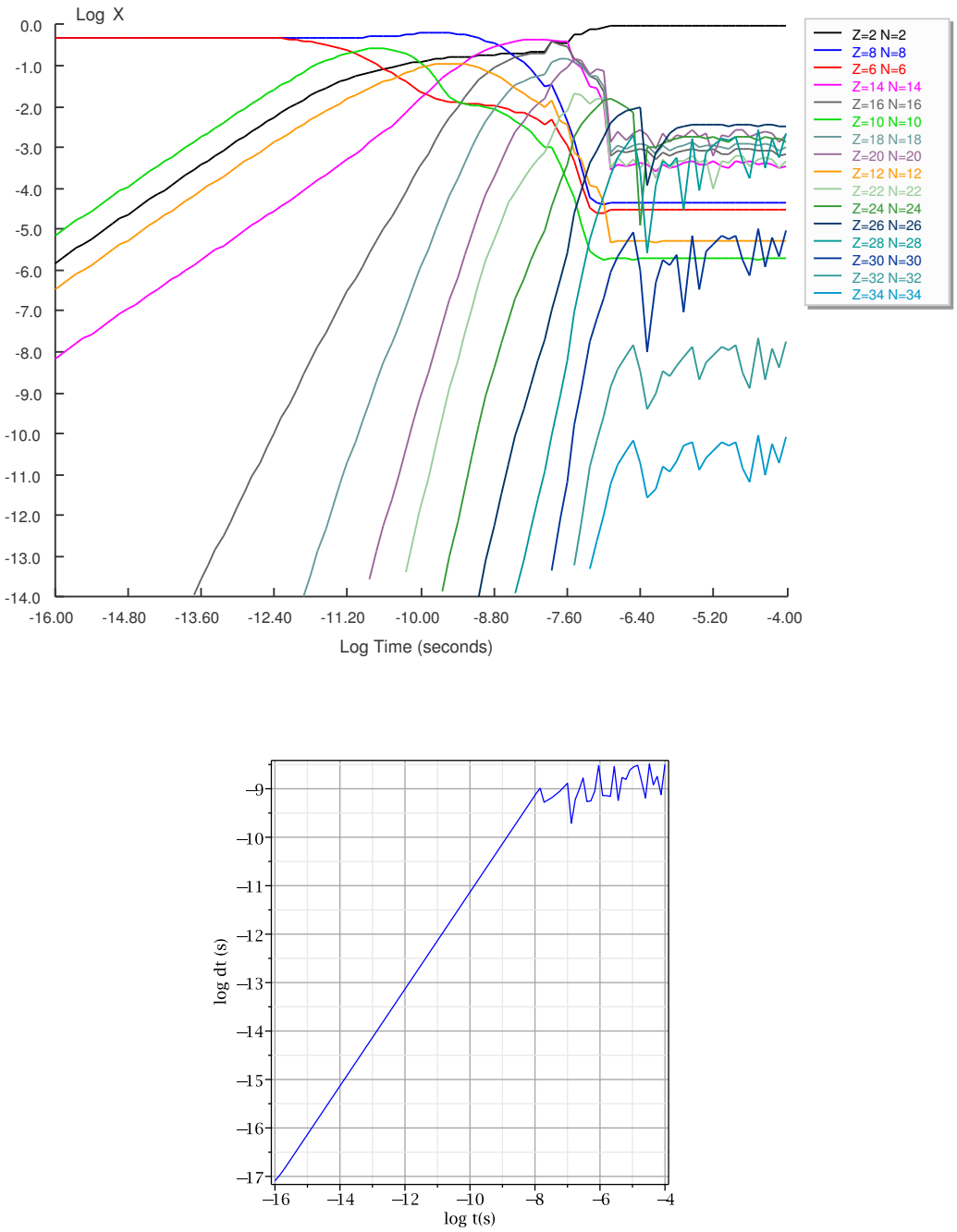


Figure 3.25: Time evolution of mass fractions (top) for an alpha network using PE choosing the equilibrium value furthest from the numerical integration's result for reimposing equilibrium and the timestep *dt* taken (bottom).

3.3.7 Partial Equilibrium Examples

We will present four examples of calculations performed with the partial equilibrium algorithm. The first example will be the same test case used earlier as an example for the asymptotic approximation: an alpha network at a constant temperature and density. The second example is the same network, but with the calculation extended out to a time of 10^{18} seconds to illustrate the timestep behavior of the system when it becomes heavily equilibrated. The third is a comparison to the calculation of an alpha network performed in Mott's thesis (Ref. [19]). The fourth example is a small network with which the partial equilibrium method as currently implemented has difficulty obtaining accurate results.

For the first example, we again use an alpha network, as illustrated in Figure 3.3, which encompasses isotopes up to ^{68}Se. The temperature is a constant $T = 7 \times 10^9$ Kelvin and the density is a constant $\rho = 10^8$ g cm^{-3}. The time evolution of the mass fractions is given in Figure 3.26 and timestep behavior is shown by Figure 3.27.

The solid curves in Figure 3.26 were calculated using PE combined with the asymptotic approximation and the dashed curves were calculated with only the asymptotic approximation. For most of the calculation the results are essentially indistinguishable from each other, especially when the system moves into equilibrium beyond a time of about $t = 1 \times 10^{-4}$ seconds. There is a deviation of about 10% shortly before the the system equilibrates and all reactions go into equilibrium with each other in several curves (^{60}Co, ^{16}O, ^{12}C, ^{20}Ne, and ^{68}Se). The largest deviation is in the population of ^{1}2C, where the partial equilibrium combined with asymptotic calculation differs from the purely asymptotic by 14.3% at $t = 1.15 \times 10^{-5}$ seconds ($\log(t) = -4.94$). However, the calculation quickly recovers and the PE plus asymptotic calculation begins agreeing with the asymptotic approximation-only calculation again.

For the entire integration range, the PE calculation outperforms the asymptotic calculation. The sudden dip in the timestep taken by PE at $t = 1 \times 10^{-8}$ (see Figure

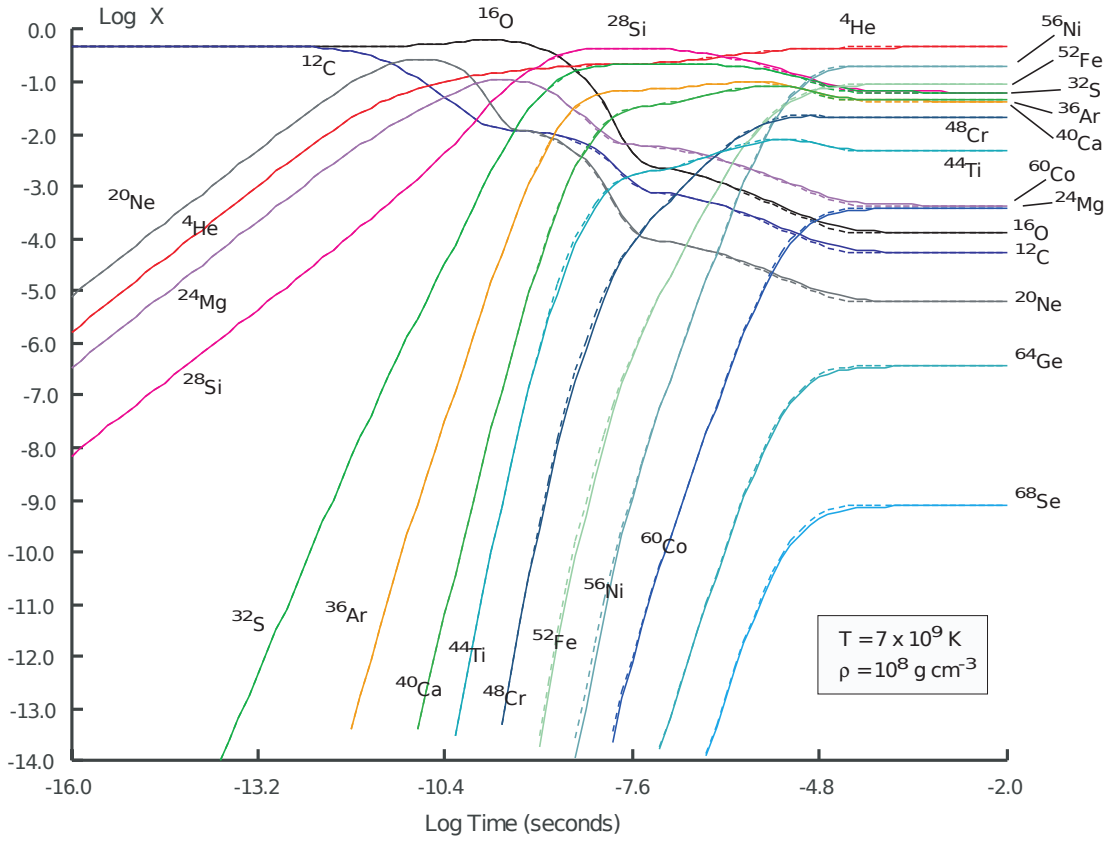


Figure 3.26: Mass fractions of an alpha network with constant temperature of $T = 7 \times 10^9$ Kelvin and density of $\rho = 10^8$ g cm^{-3}. The solid curves were calculated using the partial equilibrium method combined with the asymptotic approximation. The dashed curves were calculated using only the asymptotic flux-limiting algorithm. From Ref. [18].

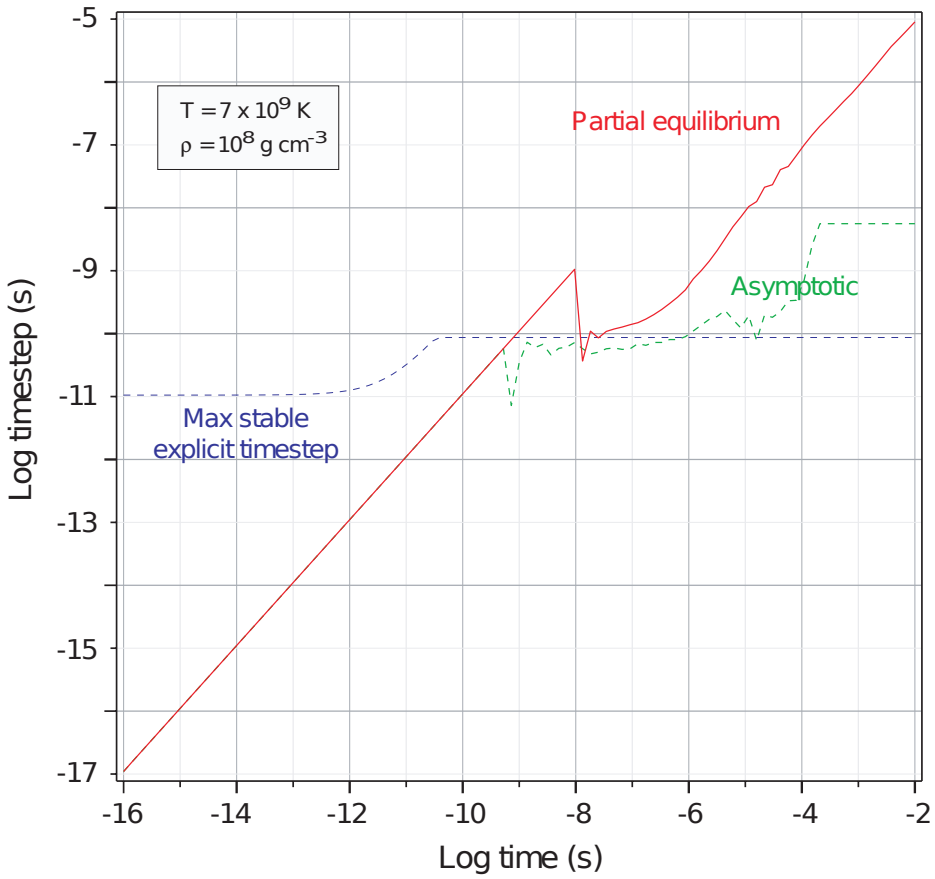


Figure 3.27: Comparison of timesteps taken by the PE plus asymptotic calculation illustrated in Figure 3.26 with the timesteps taken by the asymptotic approximation on its own. Also shown is the maximum explicit timestep. From Ref. [18].

3.27) is artificial and caused by a change in one of the parameters which controls the timestepping. We have yet to create a good automatic timestep control algorithm for PE and currently have to hand-tune parameters, which is large omission from having a complete method. It should be possible to get even better dt vs t performance from the algorithm, but as Figure 3.27 shows we can already greatly exceed the performance of the asymptotic approximation on its own. Figure 3.28 shows what portion of the network's reaction groups are being treated as in equilibrium and what portion of the isotopes in the system are being computed with the asymptotic approximation.

Next, to illustrate the power of using the partial equilibrium technique, especially combined with the asymptotic approximation, we provide an example of an alpha

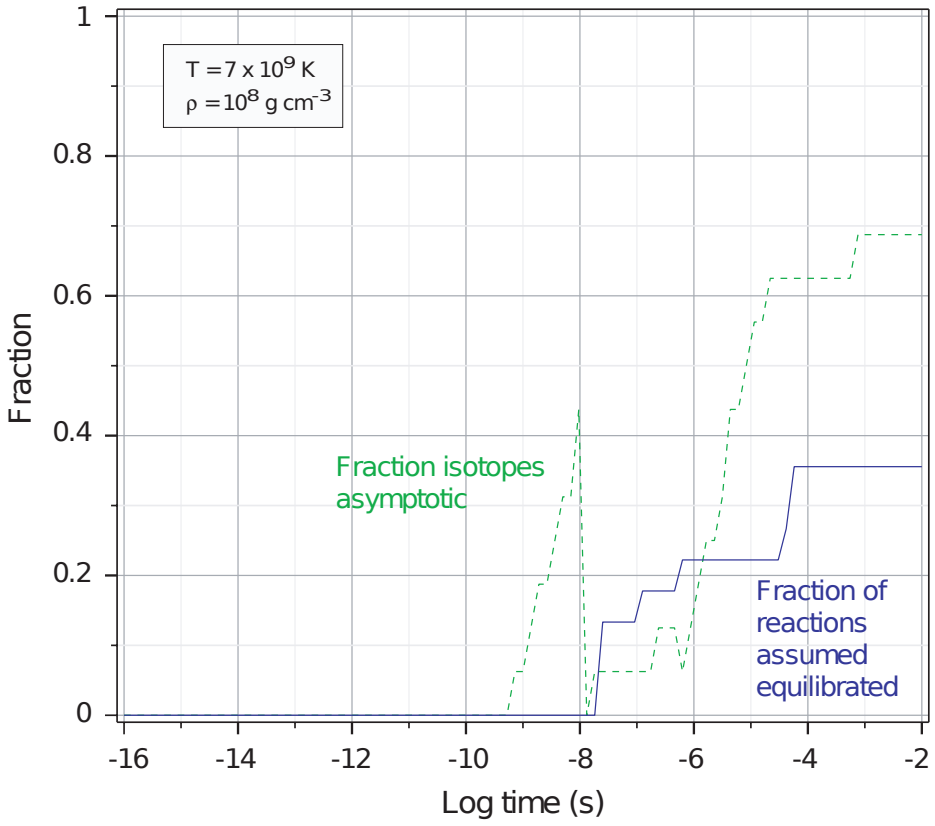


Figure 3.28: An illustration of what fraction of the isotopes in the network from Figure 3.27 are being treated asymptotically at any time (green dashed curve) and which reaction groups are actually being treated as in equilibrium (solid blue curve). From Ref. [18].

Table 3.3: Alpha Network Reaction Groups

Group	Class	Reactions	Members
1	C	$3\,^4\mathrm{He} \rightleftharpoons {}^{12}\mathrm{C}$	4
2	B	$^4\mathrm{He} + {}^{12}\mathrm{C} \rightleftharpoons {}^{16}\mathrm{O}$	4
3	D	$^{12}\mathrm{C} + {}^{12}\mathrm{C} \rightleftharpoons {}^4\mathrm{He} + {}^{20}\mathrm{Ne}$	2
4	B	$^4\mathrm{He} + {}^{16}\mathrm{O} \rightleftharpoons {}^{20}\mathrm{Ne}$	4
5	D	$^{12}\mathrm{C} + {}^{16}\mathrm{O} \rightleftharpoons {}^4\mathrm{He} + {}^{24}\mathrm{Mg}$	2
6	D	$^{16}\mathrm{O} + {}^{16}\mathrm{C} \rightleftharpoons {}^4\mathrm{He} + {}^{28}\mathrm{Si}$	2
7	B	$^4\mathrm{He} + {}^{20}\mathrm{Ne} \rightleftharpoons {}^{24}\mathrm{Mg}$	4
8	C	$^{12}\mathrm{C} + {}^{20}\mathrm{Ne} \rightleftharpoons {}^4\mathrm{He} + {}^{28}\mathrm{Si}$	2
9	B	$^4\mathrm{He} + {}^{24}\mathrm{Mg} \rightleftharpoons {}^{28}\mathrm{Si}$	4
10	B	$^4\mathrm{He} + {}^{28}\mathrm{Si} \rightleftharpoons {}^{32}\mathrm{S}$	2
11	B	$^4\mathrm{He} + {}^{32}\mathrm{S} \rightleftharpoons {}^{36}\mathrm{Ar}$	2
12	B	$^4\mathrm{He} + {}^{36}\mathrm{Ar} \rightleftharpoons {}^{40}\mathrm{Ca}$	2
13	B	$^4\mathrm{He} + {}^{40}\mathrm{Ca} \rightleftharpoons {}^{44}\mathrm{Ti}$	2
14	B	$^4\mathrm{He} + {}^{44}\mathrm{Ti} \rightleftharpoons {}^{48}\mathrm{Cr}$	2
15	B	$^4\mathrm{He} + {}^{48}\mathrm{Cr} \rightleftharpoons {}^{52}\mathrm{Fe}$	2
16	B	$^4\mathrm{He} + {}^{52}\mathrm{Fe} \rightleftharpoons {}^{56}\mathrm{Ni}$	2
17	B	$^4\mathrm{He} + {}^{56}\mathrm{Ni} \rightleftharpoons {}^{60}\mathrm{Zn}$	2
18	B	$^4\mathrm{He} + {}^{60}\mathrm{Zn} \rightleftharpoons {}^{64}\mathrm{Ge}$	2
19	B	$^4\mathrm{He} + {}^{64}\mathrm{Ge} \rightleftharpoons {}^{68}\mathrm{Se}$	2

network with constant temperature of $T = 5 \times 10^9$ Kelvin and constant density of $\rho = 1 \times 10^7$ g cm^{-3}. We use the reaction network in Table 3.3, which can fully equilibrate. Those reaction groups which have 4 members count the resonant and non-resonant terms of the reaction separately. So, for example, the single reaction $^4\mathrm{He} + {}^{12}\mathrm{C} \rightarrow {}^{16}\mathrm{O}$ is parameterized with two different components which are counted as separate reactions, even though both components combined represent a single physical process.

Figure 3.29 shows the mass fractions, timestep behavior, and fraction of reactions equilibrated during the calculation. Once the system becomes fully equilibrated near $t = 1 \times 10^3$ seconds the timesteps essentially become 10% of the time, which is the maximum timestep size allowed in our implementation for accuracy reasons. With every reaction group being treated in equilibrium all stiffness in the system has been lifted and timesteps of arbitrary size can be taken (at constant temperature). This

allows for integration of this alpha network out to a time which is approximately the age of the universe over a much shorter timescale (tens of seconds). During the calculation, the solver is still testing for equilibrium conditions and will remove any reaction group from equilibrium if it fails the test. As the graph shows, the system, once fully equilibrated and undisturbed, remains in equilibrium as would be expected. This good result suggests that as long as the system can become fully equilibrated in our formalism, we can treat instabilities arising from equilibrium conditions very effectively.

In his thesis, Mott tried to integrate a thermonuclear reaction network consisting of an alpha network, similar to the examples above. The conditions in the network were a constant temperature of $T = 5 \times 10^9$ K and a constant density of $\rho = 1 \times 10^7$ g cm^{-3}. Figure 3.30 shows the evolution of the abundances with both the partial equilibrium combined with asymptotic approximation calculation and the asymptotic approximation on its own. Figure 3.31 shows the number of isotopes treated asymptotically and the number of reaction groups which are treated in equilibrium.

The timesteps taken by the partial equilibrium plus asymptotic solver are quite competitive with a semi-implicit solver as shown in Figure 3.32. The YASS solver is a first-order semi-implicit solver used by Mott (descended from Khokhlov's work in Ref. [24]) as a comparison against the other techniques he was using. As Figure 3.32 shows, the curve labeled ASY+PE, which is the partial equilibrium combined with asymptotic approximation, is at worst taking timesteps an order of magnitude smaller than the timesteps taken by the implicit YASS solver. At several points ASY+PE is taking timesteps comparable to YASS and in both the earliest region ($log(t) < -5$) and in the time between $log(t) = 0$ and $log(t) = 1$ ASY+PE takes larger steps than YASS. Mott's quasi-steady state plus partial equilibrium (QSS+PE) solver takes timesteps several orders of magnitudes smaller than our ASY+PE. The asymptotic approximation on its own fares better than a direct explicit method at late times, but it is far out of the running for being competitive.

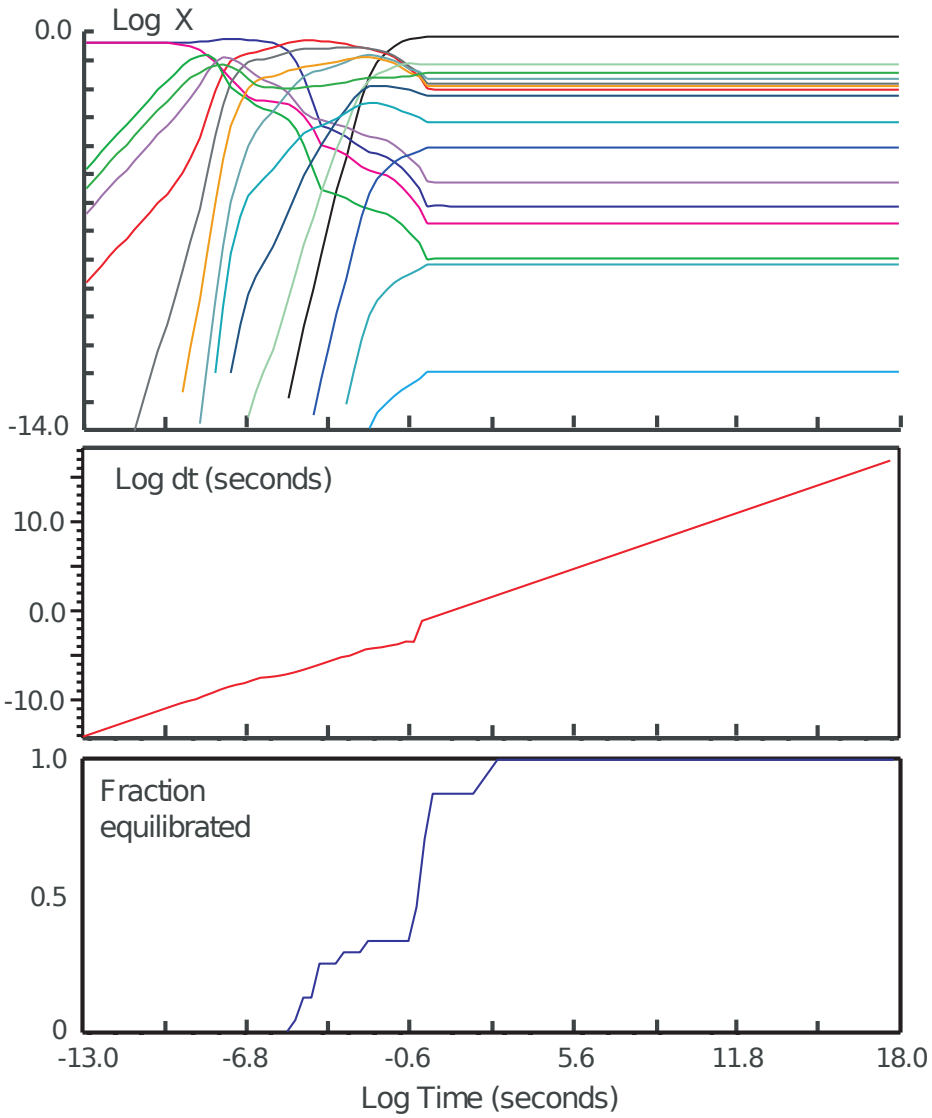


Figure 3.29: Alpha network with $T = 5 \times 10^9$ Kelvin and $\rho = 1 \times 10^7$ g cm^{-3} run out to $t = 1 \times 10^{18}$ seconds. From Ref. [18].

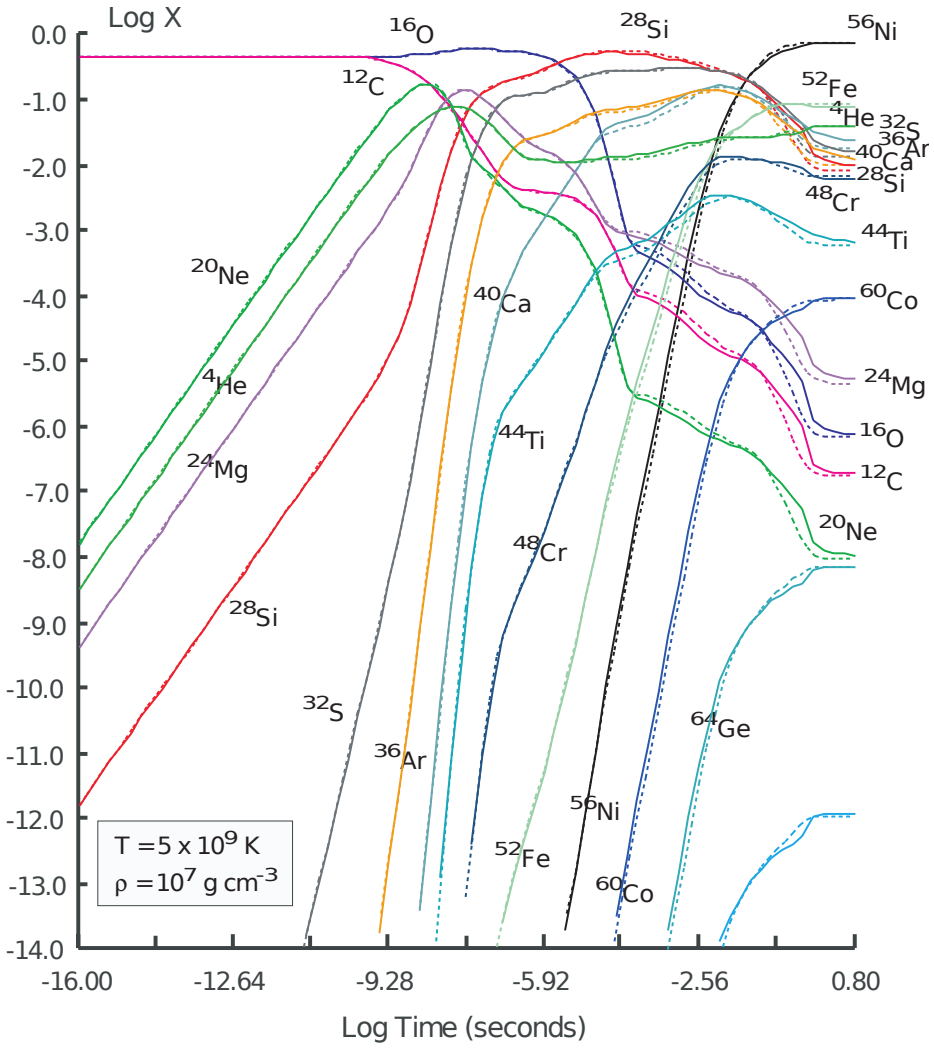


Figure 3.30: Alpha network with $T = 5 \times 10^9$ Kelvin and $\rho = 1 \times 10^7$ g cm^{-3}. The solid curves are calculated with partial equilibrium plus asymptotic approximation while the dashed curves are calculated with only the asymptotic approximation. From Ref. [18].

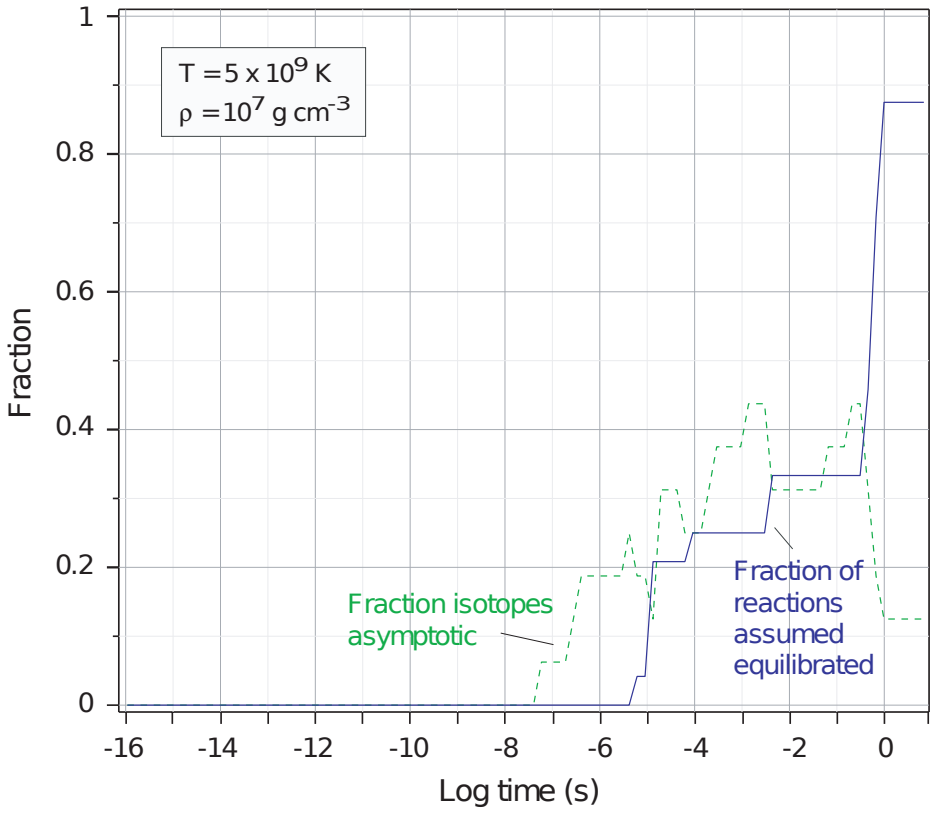


Figure 3.31: Fraction of isotopes being treated asymptotically and reaction groups being treated in equilibrium for an Alpha network with $T = 5 \times 10^9$ Kelvin and $\rho = 1 \times 10^7$ g cm^{-3}. From Ref. [18].

The exact details of the calculation performed by Mott aren't clear from reading the text, but his Figure 1.3 shows that his asymptotic approximation was unable to accurately model the behavior of an alpha network. Whereas in our tests, see Figure 3.8 for example, the asymptotic approximation is quite capable of returning results essentially identical to that from an explicit forward integration with timesteps small enough to remain stable. Further, Mott uses other techniques and improvements on the algorithm, such as predictor-corrector methods and further approximations such as their quasi-steady state approximation, but find that they still don't recover accurate results (see Figure 3.13 in Mott's thesis).

A likely cause of this problem Mott experiences is a failure to conserve particle number as strictly needed for the calculation. In our experience running these problems, conservation to about the 1% level by the end of the calculation is a requirement to get an answer with enough accuracy. That means for every timestep, particle number needs to be conserved to a very strict degree; if the calculation is expected to take 10^6 timesteps, then particle number must be conserved in every timestep at least to a tolerance of at least $.01 \times \frac{1}{1000000} = 1 \times 10^{-8}$, which is one part per hundred million. It is possible that Mott's thermonuclear reaction solver did not have such a strict constraint on particle conservation built into it, in which case the failure of the asymptotic method and poor showing of partial equilibrium in his thesis is to be expected. However this is speculation since we don't have the details the Mott calculation.

Here, we find that for the partial equilibrium plus asymptotic method that our timesteps are, everywhere, within a factor of ten or better than the YASS implicit solver and in several regions of time takes even better timesteps than YASS. There are enough potential reaction rate differences from using different libraries and Coulomb corrections that a in-depth comparison against YASS isn't justified. This result is quite promising. Explicit methods like, PE scale, linearly with the network size while implicit techniques scale quadratically or cubically. If PE is just as successful taking such large timesteps in larger networks (yet to be demonstrated) as it is in alpha

Table 3.4: 4-isotope plus Protons Reaction Groups

Group	Class	Reactions
1	C	$3^4\text{He} \rightleftharpoons {}^{12}\text{C}$
2	B	${}^4\text{He} + {}^{12}\text{C} \rightleftharpoons {}^{16}\text{O}$
3	D	${}^{12}\text{C} + {}^{12}\text{C} \rightleftharpoons {}^4\text{He} + {}^{20}\text{Ne}$
4	B	${}^1\text{H} + {}^{16}\text{O} \rightleftharpoons {}^{17}\text{F}$
5	D	${}^4\text{He} + {}^{16}\text{O} \rightleftharpoons {}^{20}\text{Ne}$
6	D	${}^4\text{He} + {}^{17}\text{F} \rightleftharpoons {}^1\text{H} + {}^{20}\text{Ne}$

networks, then PE would gain a speed increase over implicit method as network size increased.

3.3.8 Applying Partial Equilibrium to Larger Networks

The goal of these methods isn't to solve simple alpha networks, but to allow for larger and more realistic networks to be calculated in an efficient manner, especially when coupled to a hydrodynamics code in a large scale simulation. Larger networks will be much stiffer than the alpha networks investigated so far, especially because they contain protons and neutrons which couple to many or most isotopes in a large network and have very fast rates. The same fast neutron and proton reactions that bring much of the stiffness to large networks do so because they quickly equilibrate.

The simplest model to investigate the effects of protons and neutrons on networks with partial equilibrium is to take a very simple four isotope alpha network and add protons (or neutrons) and one other isotope to it. The truncated alpha network consists of ^{4}He, ^{12}C, ^{16}O, ^{20}Ne. For testing protons, add protons and ^{17}F to the network. If there are any complications which arise from the presence of fast proton reaction rates, this simple network with only 6 isotopes and 12 reactions organized into 6 reaction groups should hopefully illustrate them.

Tables 3.4 shows the breakdown of reaction groups for the proton-added network. The reduced network includes both α-capture reactions, photodisintegrations, and the proton reactions which are much stiffer than reactions in alpha networks.

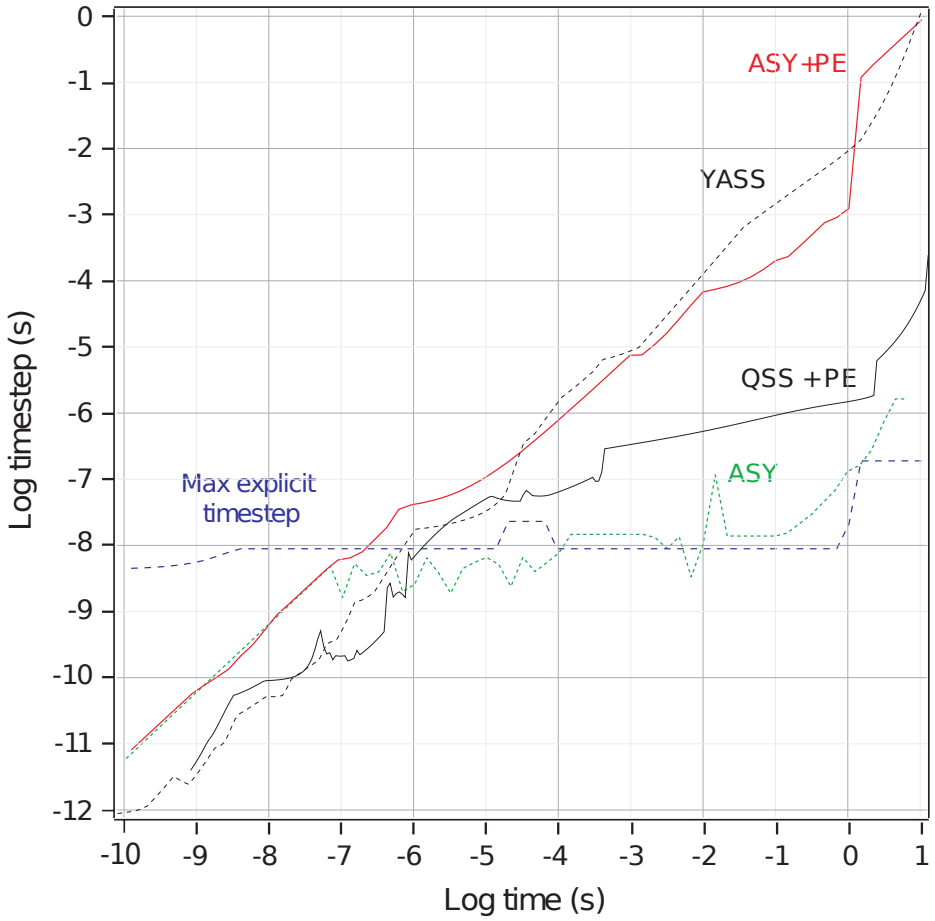


Figure 3.32: Comparison of timesteps for an Alpha network with $T = 5 \times 10^9$ Kelvin and $\rho = 1 \times 10^7$ g cm^{-3}. YASS is an implicit solver and its dashed grey curve is taken from data in Mott's thesis (Ref. [19]). The red curve (ASY+PE) is the calculation presented in Figure 3.30 for the partial equilibrium plus asymptotic curve. The solid black line (QSS+PE) is from Mott's thesis and is his implementation of a quasi-steady state and partial equilibrium solver. The dotted green curve (ASY) is the asymptotic flux-limiting algorithm presented in this work. The dashed blue line indicates the maximum stable explicit timestep. This figure taken from Ref. [18].

Figure 3.33 shows a comparison between the abundances for the proton-added network, computed with a fixed temperature $T = 5 \times 10^9$ K and density of $\rho = 10^8$ g cm^{-3}. For the timesteps shown in the bottom of Figure 3.34 the purely asymptotic and the partial equilibrium plus asymptotic methods give essentially identical results with partial equilibrium plus asymptotic taking timesteps that are several orders of magnitude larger than the purely asymptotic calculation.

Stiffness arising from additional equilibrated reactions, no matter how fast the reaction, is removed by the partial equilibrium method. Figure 3.35 shows various timescales in the proton-added network, when the network becomes fully equilibrated the timescale set by partial equilibrium conditions ($\frac{Y}{(F_{\text{forward}} - F_{\text{reverse}})}$) increases above above the timescale set by the slowest reactions in the network adding additional orders of magnitude of stiffness to the system. At later times, when the purely asymptotic calculation is forced to take timesteps only slightly larger than $dt = 10^{-10}$ seconds the partial equilibrium combined with asymptotic calculation is able to fully equilibrate the network and take arbitrarily large timesteps (limited here to 10% of the time).

Without protons added to the network (see the top of Figure 3.34), the maximum stable explicit timestep is on the order of $dt = 1 \times 10^{-8}$ seconds. With protons added the maximum stable explicit timestep is $dt = 1 \times 10^{-13}$ seconds. The difference between the pure asymptotic and partial equilibrium plus asymptotic timestep arises from the increased stiffness of the system which depresses the maximum timestep the asymptotic method alone can take, resulting in a factor of 10^6 difference in timesteps at the end of the calculation. This stiffness comes from individual reactions in the network approaching equilibrium. As the proton-influenced reaction pairs become equilibrated, the asymptotic solver is unable to increase its timestep. The partial equilibrium plus asymptotic solver has no difficulty with them, removing the stiff source terms from its equations and with them the equilibrium stiffness. It's clear that the purely asymptotic timesteps are simply not competitive against the partial

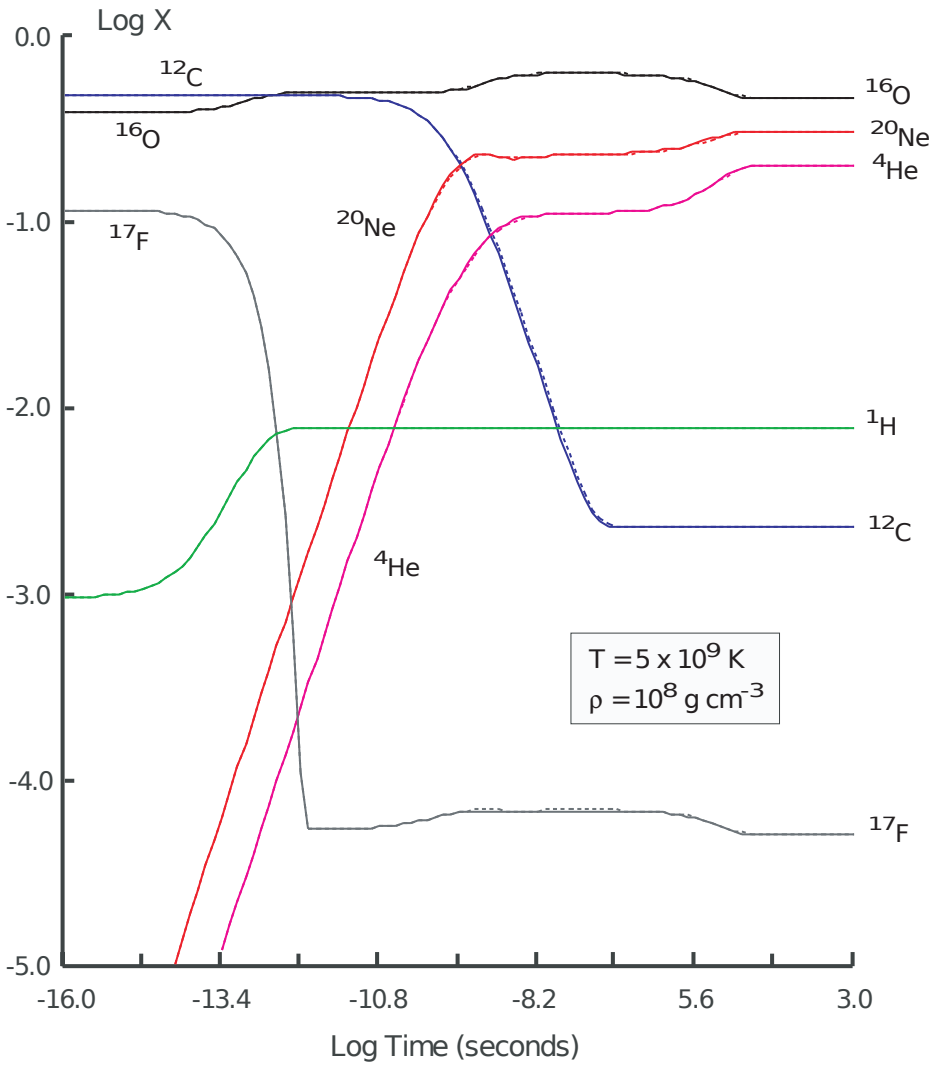


Figure 3.33: Time evolution of mass fractions for a network of Protons, ^{4}He, ^{12}C, ^{16}O, ^{17}F, and ^{20}Ne computed with a constant temperature and density of $T = 5 \times 10^9$ Kelvin and $\rho = 1 \times 10^8$ g cm^{-3}. The solid curves are calculated with partial equilibrium plus asymptotic method, the dashed curves are an asymptotic approximation only calculation.

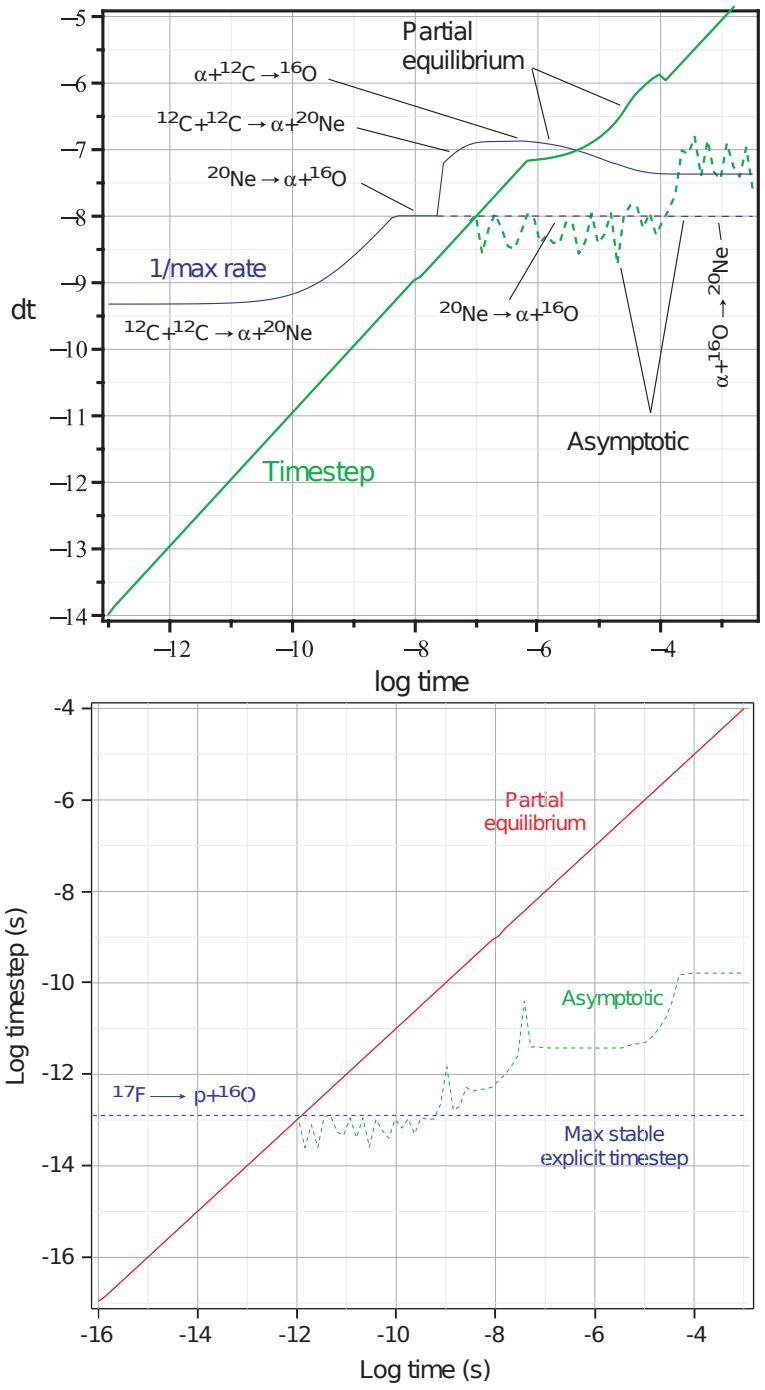


Figure 3.34: Top: Timesteps *dt* vs. time for the evolution of the 4-isotope alpha network with nothing added. Bottom: Timestep *dt* vs. time for the evolution of the 4-isotope alpha network plus protons and ^{17}F. From Ref. [18].

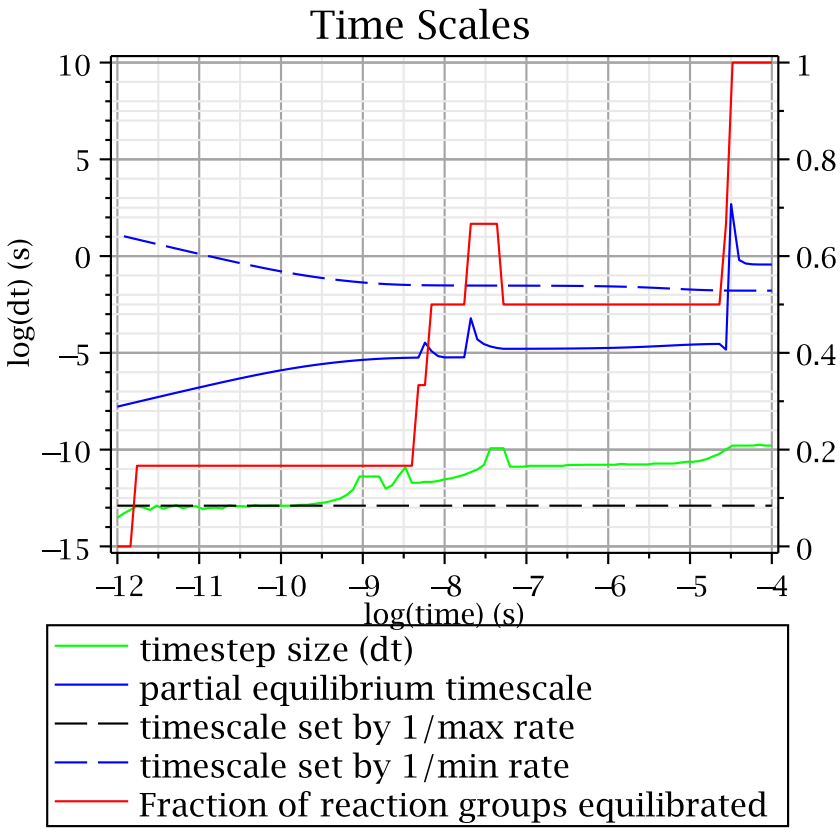


Figure 3.35: Comparison of various timescales in the 4-isotope plus protons test problem. The green curve is the actual timestep taken by the asymptotic solver. The solid blue curve is the timescale set by reaction pairs coming into (near) equilibrium. The dashed black curve is the timescale set by the fastest reaction in the system. The dashed blue curve is the timescale set by the slowest reaction in the system. The red curve, which uses the vertical axis on the right hand of the graph, shows the fraction of reaction groups which are in equilibrium.

equilibrium combined with asymptotic timestep, while both methods give the same numerical result for the populations of the isotopes in the system.

The simple 4-isotope plus protons model examined in this section illustrates that the partial equilibrium method (when combined with the asymptotic approximation) can take timesteps 10^8 times larger than the maximum stable explicit timestep. Combined with results from the examples in Section 3.3.7 where partial equilibrium plus asymptotic was able to take timesteps comparable to that of the semi-implicit YASS solver, there is great hope that the partial equilibrium method will be able to competitively solve larger realistic reaction networks.

Chapter 4

Network Calculation Comparison

The asymptotic algorithm presented in Chapter 3 is a promising new technique for solving coupled sets of ODEs, but it has competition from well-established and well-tested codes using implicit and semi-implicit methods. Presented here is a suite of test problems which test the asymptotic algorithm against an implicit method (backward Euler) and a semi-implicit method (Bader-Deuflhard).

Initially it was proposed to test the asymptotic algorithm with a suite of six test problems, which were a stellar evolution simulation, a classical nova simulation, a simulation of the r-process, a type Ia supernova simulation, a type II supernova simulation, and a x-ray burst simulation. It proved difficult to find a temperature and density profile for a stellar evolution problem, so a tidally induced supernova simulation was substituted instead. Tables 4.1 through 4.4 show an overview of the physical and computational parameters of the test problems. The test problems and the results of the asymptotic calculation are described below in their own sections below.

4.1 Astrophysical Reaction Rates

To calculate reaction rates and fluxes in astrophysical systems, it is necessary to know the interaction cross section for the nuclear processes involved. The discussion on this

Table 4.1: Summary of times for test problems

Test Problem	Start Time (s)	Stop Time (s)
Nova	0	$3.578311183457 \times 10^{10}$
r-process	6.89×10^{-2}	4.378×10^{-1}
Supernova Type Ia	2.0×10^{-2}	2.0
Supernova Type II	3.42×10^{-1}	5.0
Tidally Induced Supernova	6.5	6.86×10^{1}
X-ray Burst	1.0×10^{1}	2.35×10^{4}

Table 4.2: Summary of temperatures for test problems

Test Problem	T_{min} (K)	T_{max} (K)
Nova	1.43×10^{6}	3.92×10^{8}
r-process	2.5×10^{9}	9.99×10^{9}
Supernova Type Ia	1.3×10^{7}	4.765×10^{9}
Supernova Type II	1.04×10^{9}	9.9×10^{9}
Tidally Induced Supernova	1.05×10^{7}	2.59×10^{9}
X-ray Burst	7.80×10^{6}	9.94×10^{8}

Table 4.3: Summary of densities for test problems

Test Problem	ρ_{min} (g cm^{-3})	ρ_{max} (g cm^{-3})
Nova	1.09×10^{-5}	2.70×10^{4}
r-process	5.52×10^{4}	3.81×10^{6}
Supernova Type Ia	4.48×10^{6}	9.45×10^{8}
Supernova Type II	4.60×10^{3}	1.03×10^{7}
Tidally Induced Supernova	4.16×10^{-1}	7.09×10^{5}
X-ray Burst	1.39×10^{3}	1.14×10^{6}

Table 4.4: Summary of network size for test networks

Network	Number of Isotopes	Number of Reactions
Alpha	14	41
Nova	169	1950
r-process	2184	25902
Si02_fix	299	3463
SN150	150	1609
Supernova Type II	1072	11996
X-ray Burst	304	3314

topic parallels that given in Ref. [25]. For a particle i interacting with a target particle j the reaction cross section is given by

$$\sigma = \frac{\text{number of reactions per target per second}}{\text{flux of projectile particles}} = \frac{r/n_j}{n_i v} \tag{4.1}$$

where the relative velocity v between the targets j and the projectiles i is constant, n_j is the number density of the targets and n_i is the number density of the projectiles. r is the number of reactions per cm^3 per second, and in the simplest case (constant relative velocity) is given by

$$r = \sigma v n_j n_i, \tag{4.2}$$

while a more general expression is

$$r_{j,i} = \int \sigma(|\overrightarrow{v}_j - \overrightarrow{v}_i|)|\overrightarrow{v}_j - \overrightarrow{v}_i| d^3 n_j d^3 n_i. \tag{4.3}$$

Using Maxwell-Boltzmann statistics we can write the number densities in the form

$$d^3 n = n\Big(\frac{m}{2\pi k_B T}\Big)^{3/2} \exp\Big(-\frac{mv^2}{2k_B T}\Big) d^3 v. \tag{4.4}$$

k_B is Boltzmann's constant and T is the temperature. Using Eqn. (4.4), the n_i and n_j can be pulled out of the integral in Eqn. (4.3). If $\langle \sigma v \rangle$ is defined as the velocity integrated cross section, then we can express Eqn. (4.3) as

$$r_{j,i} = \langle \sigma v \rangle_{j,i} n_j n_i. \tag{4.5}$$

Define $\langle j, i \rangle$ to be the integrated cross section, then for these nuclear reactions in very hot environments we can write

$$\langle j, i \rangle = \langle \sigma v \rangle_{j,i} = \Big(\frac{8}{\mu\pi}\Big)^{1/2} (k_B T)^{-3/2} \int_0^\infty E\sigma(E) \exp(E/k_B T) dE, \tag{4.6}$$

where μ is the reduced mass of the system made of particles i and j and E the center of mass energy for the system.

For computing rates based on these sorts of equations there are a variety of data compilations available. Caughlan and Fowler published a compilation of reaction rates for astrophysically important thermonuclear reactions in 1988 [26]. There is also the REACLIB library of rates, which is available online or partially given in publications such as Ref. [14]. Much of this data has been collected and collated in places online such as Ref. [27] or Ref. [28].

4.2 On the Implicit and Semi-Implicit Solvers

The implicit (backward Euler) and semi-implicit (Bader-Deuflhard) calculations were performed with XNET, which is a general-purpose reaction network used in Ref. [29]. Both methods can be used with one of a number of different linear algebra packages: LAPACK [30], MA28 [31], and PARDISO [32] [33] [34] [35]. PARDISO and MA28 are packages designed to solve sparse systems of equations, while LAPACK is a dense solver.

Before comparisons were made between the asymptotic solver and the implicit and semi-implicit solvers, tests were run to determine which of the three linear algebra packages performed the best. In addition to three choices of linear algebra package, XNET has two methods of determining convergence as it does Newton-Raphson iteration to solve the system of equations. First is the relative change of abundances ($\Delta Y/Y$). If $\sum_{i=1}^{m} \Delta Y_i/Y_i$ over all m species in the network is smaller than some tolerance, the iteration loop will consider the system converged for that timestep. The second method is based on mass conservation. The system of equations is considered converged when the total mass in any timestep is conserved to within some tolerance.

Tables 4.5, 4.6, 4.7 show the calculation time for each of the test problems, run in XNET using backward Euler on a 3 GHz processor for all three linear

algebra packages and both convergence tests. Some trends become immediately clear. LAPACK is not the package of choice for "large" problems. The nuclear network used in the simulation of the r-process contains 2184 isotopes. LAPACK using mass conservation to determine convergence takes 2399 seconds to integrate the network. In comparison the PARDISO package takes only 100.6 seconds to perform the same set of calculations.

There is a general trend that using mass conservation as the iteration convergence criteria provides superior performance to using the relative change of abundances, however this is not always true. The tidally induced supernova cases in particular do not fit the pattern. For the 299-isotope network calculated with LAPACK, mass conservation (26.95 seconds) provided better performance than the relative change of abundance test (59.43 seconds). However for the same test problem, MA28 and PARDISO give the result that the abundance test gives somewhat superior performance: a difference of 8.77 seconds in the MA28 case and a difference of 3.17 seconds in the PARDISO case. Tests with a 150-isotope network in the same test problem show that for LAPACK the relative change of abundance test is faster by about 37% than the mass conservation test. The relative change of abundance test in MA28 with a 150-isotope network is 35% faster than the mass conservation test, while with PARDISO mass conservation is about 24% faster than the change of abundance test.

For the 14-isotope alpha network all three solvers give roughly comparable performance results; in this situation much of the computation time is spent on start up costs. If you increase the number of zones performing the computation to 100, which has the effect of making XNET compute the same network under the same conditions 100 times, then all three solvers with both convergence conditions still give fairly comparable run times of between 20 and 30 seconds (for both the type Ia and tidally induced supernovae). An exception is the PARDISO solver using the relative change of abundances as its convergence test, which takes 57.9 seconds for the

Table 4.5: Backward Euler Timing in XNET with LAPACK

Test Problem	Mass Convergence (s)	Abundance Convergence (s)
Alpha Network - Tidally Induced SN	1.72	1.53
Alpha Network - Type Ia SN	1.79	2.54
150 Network - Tidally Induced SN	16.49	10.31
150 Network - Type Ia SN	19.66	12.91
299 Network - Tidally Induced SN	26.95	59.43
299 Network - Type Ia SN	22.40	60.05
Classical Nova	7.79	13.55
Type II SN	388.30	1308.80
X-ray Burst	30.65	158.00
r-process	2399.00	61979.90

type Ia supernova simulation with 100 zones and 39.5 seconds for the tidally induced supernova simulation with 100 zones.

As network size increases, PARDISO seems to generally be a better choice for speed than MA28. The fastest run times for the 150-isotope network in the type Ia simulation are 7.62 seconds (with mass conservation) for MA28, and 6.37 seconds (with mass conservation) for PARDISO. For the 299-isotope network in the type Ia simulation, the fastest run times are 9.79 seconds (with mass conservation) and 28.15 seconds (with mass conservation). The best run times for the 150-isotope network in the tidally induced supernova simulation are 8.17 seconds (MA28 with the relative change of abundances test), and 7.52 seconds (PARDISO with mass conservation). MA28 is faster for the 299-isotope network in the tidally induced supernova case (16.36 seconds) than PARDISO (20.49 seconds). In the 169-element classical nova simulation MA28 has a shorter run time than PARDISO. However, the two largest networks (the 1072-isotope network used in the type II supernova simulation and the 2184-element network used in the r-process simulation) are computed the fastest by using PARDISO with mass conservation, which is most strongly illustrated by the r-process case where PARDISO is over 2.5 times faster than MA28 (both using mass conservation as the convergence criteria).

Table 4.6: Backward Euler Timing in XNET with MA28

Test Problem	Mass Convergence (s)	Abundance Convergence (s)
Alpha Network - Tidally Induced SN	1.15	1.44
Alpha Network - Type Ia SN	2.03	2.33
150 Network - Tidally Induced SN	12.56	8.17
150 Network - Type Ia SN	7.62	32.58
299 Network - Tidally Induced SN	25.13	16.36
299 Network - Type Ia SN	9.79	62.66
Classical Nova	6.04	20.21
Type II SN	82.38	750.30
X-ray Burst	46.46	77.50
r-process	272.30	4519.40

Table 4.7: Backward Euler Timing in XNET with PARDISO

Test Problem	Mass Convergence (s)	Abundance Convergence (s)
Alpha Network - Tidally Induced SN	1.97	2.84
Alpha Network - Type Ia SN	2.96	1.98
150 Network - Tidally Induced SN	7.52	9.87
150 Network - Type Ia SN	6.37	15.60
299 Network - Tidally Induced SN	23.66	20.49
299 Network - Type Ia SN	28.15	39.40
Classical Nova	18.70	16.10
Type II SN	59.89	141.20
X-ray Burst	42.20	98.90
r-process	100.60	264.30

No matter which linear algebra package is chosen the result of the calculation and the timesteps taken were virtually identical. In some cases the number of timesteps taken would differ by about 1% (10 timesteps out of about 1000). The choice of convergence criteria has a stronger effect on the number of timesteps taken. While generally the convergence criteria has the same level of effect on the number of timesteps as the choice of linear algebra package, for the r-process and type II supernova simulations the effect was much stronger. For the r-process test problem using LAPACK, mass conservation took 1131 timesteps while the change of relative abundances for the convergence criteria caused XNET to take 4490 timesteps. The r-process problem with MA28 shows a similar pattern with the mass conservation condition resulting in 1125 timesteps and the change of relative abundances criteria resulting in the network taking 2925 timesteps. Interestingly the PARDISO solver took 1131 timesteps for both choices of convergence criteria. In the type II supernova test problem using MA28 the mass conservation test causes the network to take 1047 timesteps, with the change of relative abundances test the network takes 2404 timesteps (PARDISO takes 1047 and 1063 timesteps for the two different tests).

These observations of the number of timesteps taken partially explain why PARDISO is able to outperform MA28 in larger networks. Even if PARDISO and MA28 had the same cost per timestep, for the type II supernova case using the change of relative abundances convergence test, the difference in the number of timesteps would cause MA28 to take 2.26 times longer than PARDISO. However, the costs are not the same. For these larger networks, MA28 takes longer per timestep. Again, for the type II supernova example, with the change of relative abundances convergence test, MA28 has an average real time per timestep of $750.3/2404 = 0.312$ seconds per timestep while PARDISO only takes $141.2/1063 = 0.133$ seconds per timestep. Comparing for the same test problem with the mass conservation criteria yields a time per timestep of $82.38/1047 = 0.079$ seconds per timestep for MA28 and $59.89/1047 = 0.057$ seconds per timestep. The r-process test problem yields real time per timestep costs of $272.3/1125 = 0.242$ seconds for MA28 with mass conservation,

4519.4/2925 = 1.55 seconds for MA28 with the relative change of abundances condition, 100.6/1131 = 0.089 seconds for PARDISO with mass conservation, and 264.3/1131 = 0.234 for PARDISO with the relative change of abundances condition.

The relative change of abundances convergence criteria is a bit more "strict" than using mass conservation to determine convergence. In mass conservation the total nucleon number of the system must be conserved, and thus a slight overabundance of one isotope can be "compensated" for by a slight deficiency in other isotopes. With the relative change of abundances test, the sum of all the $\|dY/Y\|$ in the system has to be smaller than some tolerance. None of the errors can effectively mask each other due to taking the absolute value of the relative changes when they are summed. This difference in the two convergence criteria available in XNET provides some explanation for why the very large networks calculated using MA28 (which were examined in the previous paragraphs) take so many more timesteps for the relative change in abundance criteria than for the mass conservation case: if a Newton-Raphson iteration fails to converge, XNET tries to integrate the equations again using a smaller timestep, making the network take more timesteps total and also take more time per timestep as it takes more trial timesteps and performs more Newton-Raphson iterations. PARDISO is a more robust linear algebra package for these large networks as it doesn't exhibit this behavior. The XNET backward Euler calculation with the best timestepping behavior, i.e. the fewest total timesteps, is used in the comparisons of the test problems below.

Bader-Deuflhard was tested less thoroughly than backward Euler. In the comparisons with the asymptotic algorithm in the test problems, the Bader-Deuflhard calculations were all performed using LAPACK for the linear algebra package and the abundance convergence condition as the convergence criteria. The MA28 solver proved unreliable when combined with Bader-Deuflhard in XNET, and given that Bader-Deuflhard calculations with LAPACK appeared to have very similar timestepping behavior as the best backward Euler calculations we decided to proceed with a reliable calculation as a benchmark.

One last consideration is the scaling of the cost per timestep taken by the implicit solvers. Recall from Chapter 2 that backward Euler and Bader-Deuflhard both require matrix inversions to solve the system of equations at each timestep. One of the things repeatedly stated as motivation for using explicit methods to treat these problems is that they scale linearly while implicit methods scale quadratically or cubically. Indeed, the cost for matrix inversions LAPACK is stated in Ref [36] as scaling as $\frac{4}{3}n^3$ where n is the size of the matrix, which here is analogous to the number of isotopes in the matrix. Sparse solvers scale better than a dense matrix solver like LAPACK which gives MA28 and PARDISO a clear advantage when solving the larger networks as seen in Table 4.5 through Table 4.7.

4.3 Methodology

Due to differences in the code bases between the asymptotic and standard implicit methods direct comparisons of run times are not very useful. The asymptotic approximation has been implemented in a Java-based program while backward Euler and Bader-Deuflhard are implemented in XNET which is written in FORTRAN. Little effort has been made towards optimizing the asymptotic calculations to this point. XNET has had many people contribute to it, adding features or improving speed, over the years. So there is not only a difference in speed that will arise between comparing Java against a compiled FORTRAN code , but there are wildly differing levels of optimization to deal with as well.

Comparing the time needed to run each calculation between the two codes will not give a useful picture of the capabilities of the algorithms. Looking at the size of the timesteps taken can give some idea of which algorithm is the best to use in a given situation.

Both the asymptotic approximation and the implicit methods need to calculate the derivatives used to advance the system of equations. For the asymptotic method this is the most expensive computation step. Both backward Euler and Bader-Deuflhard

then need to build the Jacobian matrix and then invert it to advance to find a solution and advance to the next timestep. For larger networks the Jacobian build and inversion can take 70% or more of the program's calculation time. If XNET spends 70% of its time inverting matrices, then it is only spending 30% of its time finding derivatives. The asymptotic algorithm can then afford to take 3 timesteps where the implicit method only took a single timestep.

That suggests that counting the number and size of timesteps taken by the algorithms can lead to a general idea of whether the asymptotic approximation has the possibility of being competitive against the standard implicit techniques currently in use. This isn't a definitive assessment of the asymptotic algorithm, there are too many differences between its implementation and that of backward Euler or Bader-Deuflhard, rather it is a first look at the asymptotic algorithm's behavior to determine if it is worth the time and effort needed for a deeper look or if it is grossly uncompetitive.

4.4 Test Problem: Nova

Novae are astronomical events where a star suddenly brightens and gives off a large amount of energy for a period of time. Novae are associated with binary stellar systems and are often recurrent. A typical nova outburst occurs when a larger companion star accretes hydrogen-rich matter onto a smaller white dwarf companion [16]. The accreted matter forms a degenerate gas shell on the white dwarf's surface and eventually ignites with hydrogen burning via the CNO cycle. CNO burning burns hydrogen into helium, utilizing a set of reactions where carbon, nitrogen, and oxygen serve as catalysts in a cycle reaching relatively steady state while hydrogen decreases and helium increases [37]. CNO burning in the nova eventually heats the accreted matter to a high enough temperature that the "hot" CNO cycle becomes the predominant source of energy production.

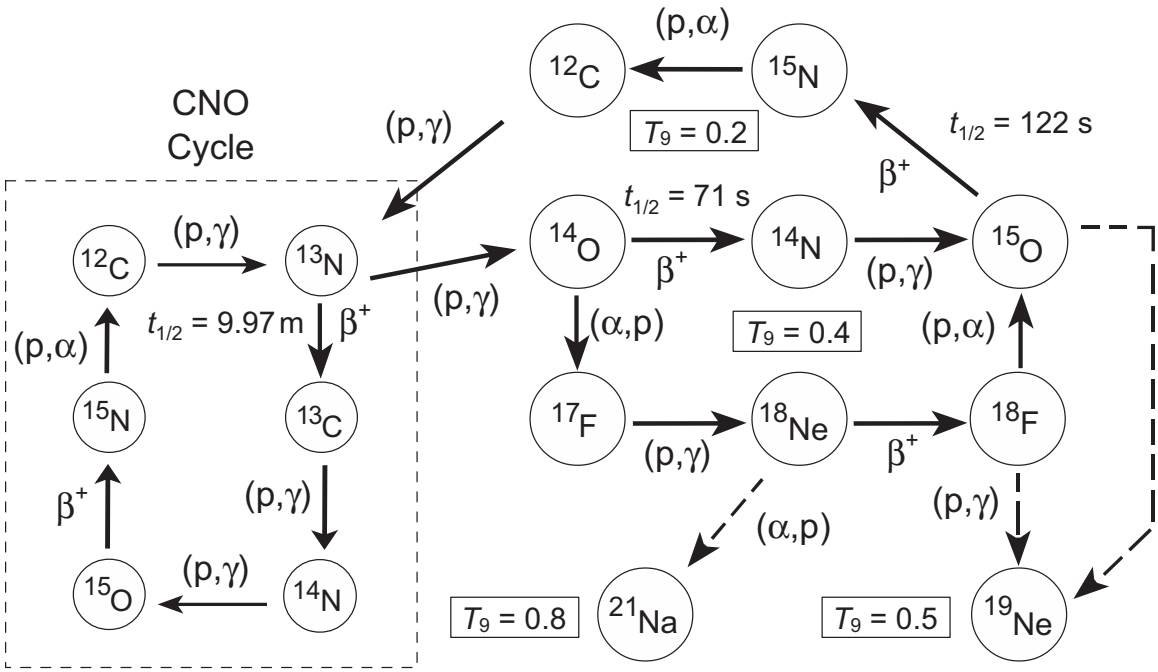


Figure 4.1: This diagram shows the regular CNO cycle (the boxed region) and its connection to the hot CNO cycle, and how the different isotopes are connected together via reactions. From [16].

The hot CNO cycle includes the normal CNO cycle as its foundation, but the high temperatures and pressures under which hot CNO occurs, $10^8 - 10^9$K [38], allow for additional sets of reactions to occur as illustrated in Figure 4.1. With sufficient temperatures it becomes possible to produce isotopes which "escape" the hot CNO cycle, such as ^{15}O capturing an α-particle and producing a ^{19}Ne nucleus [39][40].

As the accreted material envelope around the white dwarf continues to heat up, the highly degenerate hydrogen envelope leads to a thermonuclear runaway. Degeneracy is lifted, the gas to expands and is ejected from the white dwarf [41]. The expansion cools the gas off and quenches the thermonuclear runaway.

Figure 4.2 shows the temperature and density profiles for this calculation. The peak temperature is $T = 3.92 \times 10^8$ Kelvin. The entire calculation over over a range of 3.58×10^{10} seconds, the accretion time, though the nova outburst where the

Table 4.8: Initial composition of the Nova

Isotope	Mass Fraction
^{1}H	0.365
^{4}He	0.133
^{16}O	0.150
^{20}Ne	0.249
^{24}Mg	0.100

temperature rises and density decreases rapidly is very narrow in comparison (see Figure 4.3), lasting for about 100 seconds.

The nuclear reaction network for this system has 169 isotopes in it and is illustrated in Figure 4.4. The initial composition comes from Ref. [41] and is a mixture of half solar abundances and half Oxygen-Neon-Magnesium to represent the mixing of material between the accreted gas and the (ONeMg) white dwarf. The major components of the initial composition are given in Table 4.8

Until the large temperature rise at about $t = 3.578 \times 10^{10}$ seconds the nuclear populations remain relatively constant. As the temperature begins to slowly rise before the "spike" in temperature, some populations begin to evolve dramatically, such as ^{12}C decreasing, which is visible in Figure 4.5. Substantial change does not occur until the spike occurs and the temperature rises dramatically to its peak of $T = 3.92 \times 10^{8}$ K. During this relatively short period of time many populations undergo rapid evolution as illustrated in Figure 4.6.

Table 4.9 shows a selection of the mass fractions for ten arbitrarily chosen isotopes at the end of the simulation. The final asymptotic populations for ^{1}H and ^{4}He are within 1% of the backward Euler and Bader-Deuflhard result. For the other selected isotopes, the asymptotic approximation returns final mass fractions which are within 10% of the results from the implicit solvers.

Figures 4.7 and 4.8 show the size of the timestep taken at each time for each of the three methods. The number of timesteps taken by each solver in several thermodynamic regions are shown in Table 4.10. The accretion region takes place

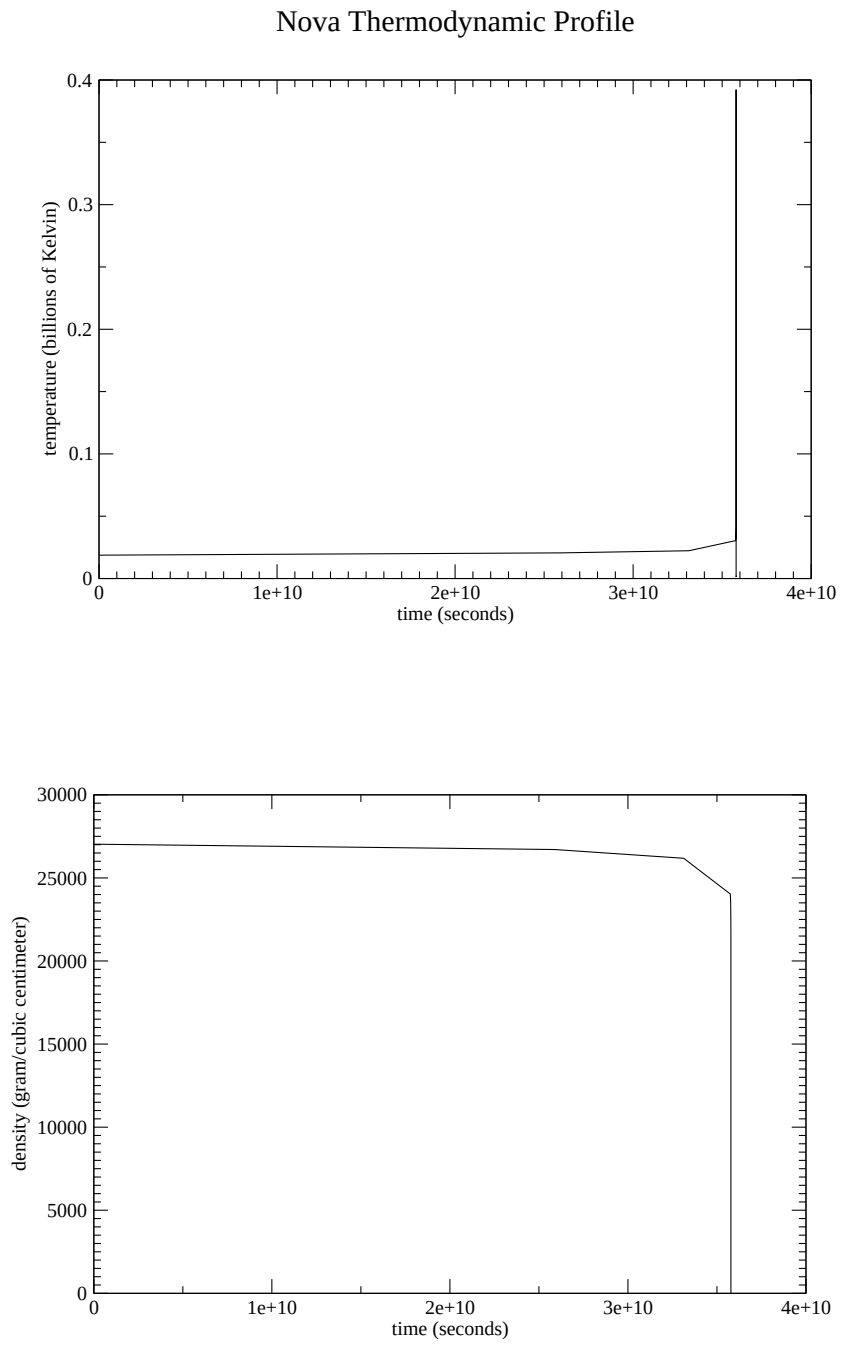


Figure 4.2: The temperature and density profiles for the Nova problem.

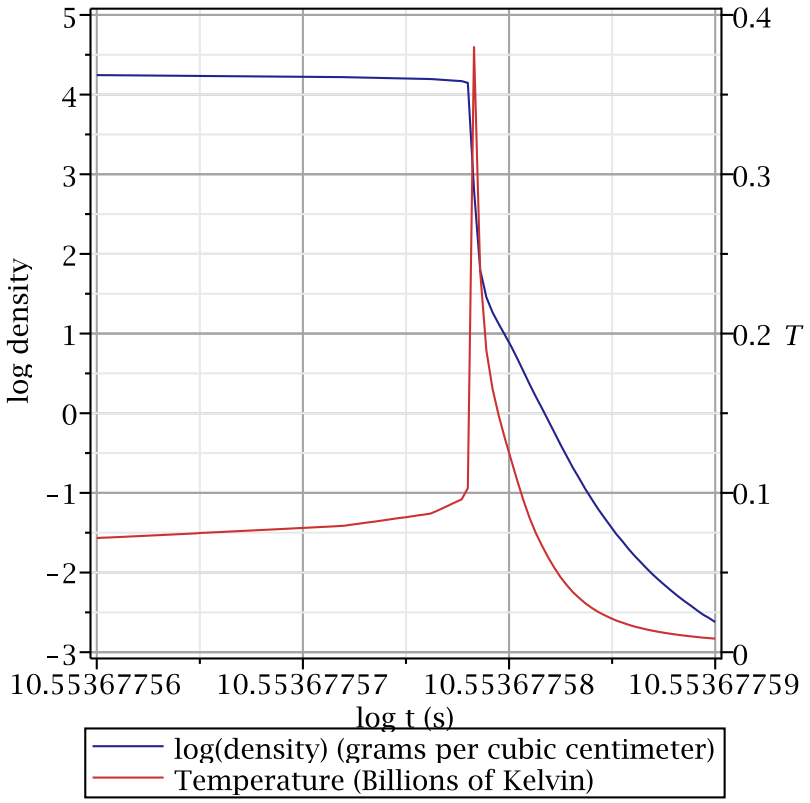


Figure 4.3: Nova temperature profile zoomed in around the sudden rise and subsequent drop in temperature. The red curve is the temperature as measured in billions of Kelvin (using the axis labeled T_9) and the blue curve is the logarithm of the density.

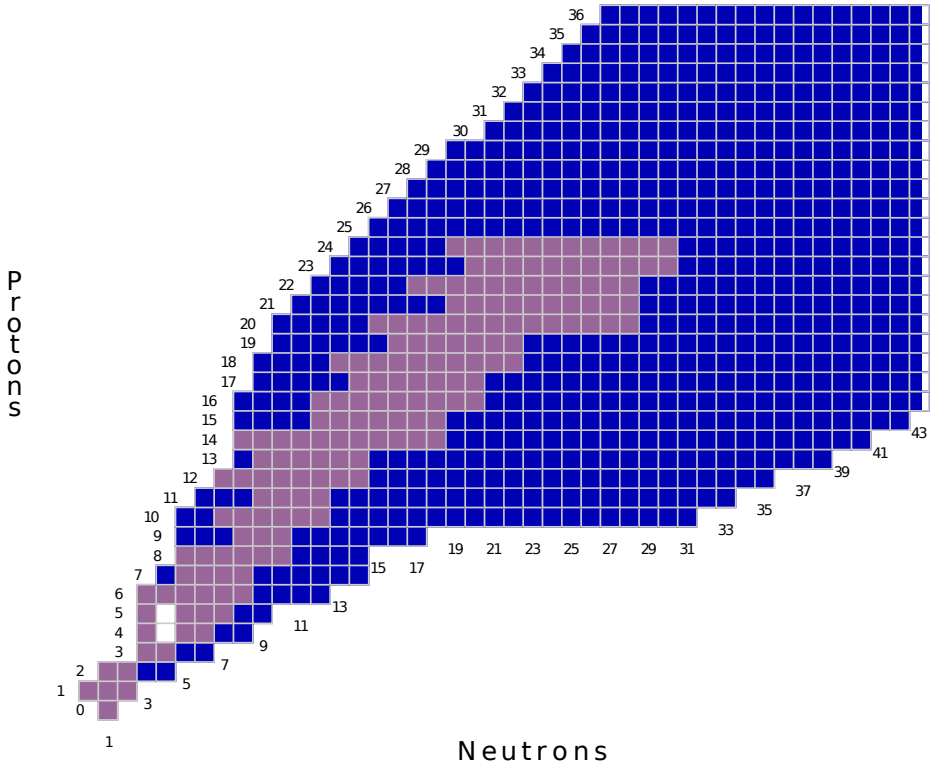


Figure 4.4: Overview of the 169 isotope nuclear network for the nova calculation.

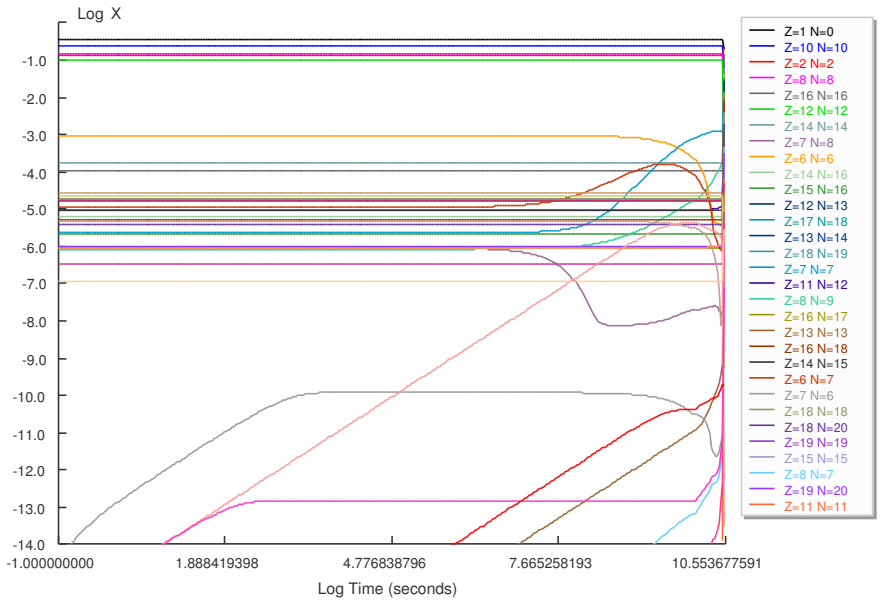


Figure 4.5: Time evolution of mass fractions for the classical nova simulation.

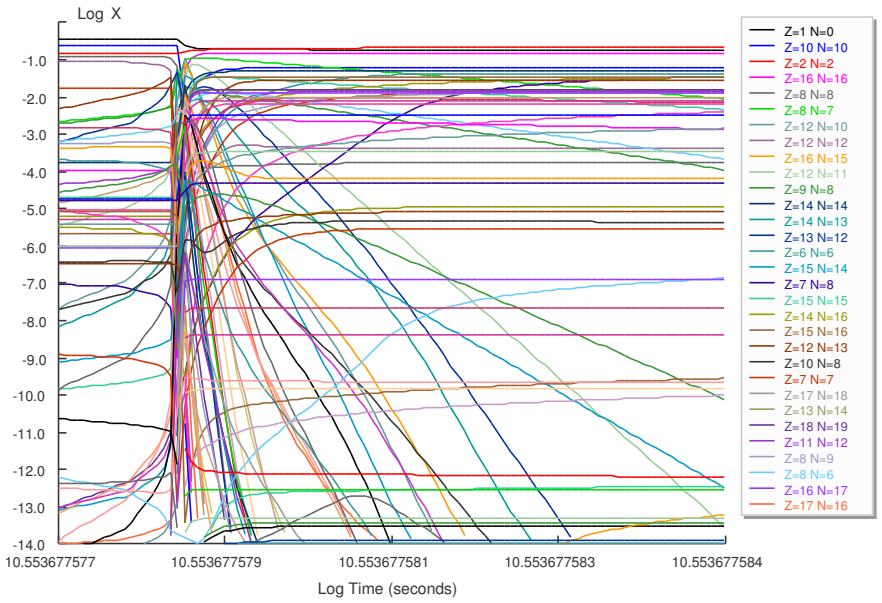


Figure 4.6: Time evolution of mass fractions for the classical nova simulation zoomed into the region around the rapid temperature increase.

Table 4.9: Selected final mass fractions for the nova test problem

Isotope	Asymptotic	Backward Euler	Bader-Deuflhard
^{1}H	1.88×10^{-1}	1.89×10^{-1}	1.89×10^{-1}
^{4}He	2.12×10^{-1}	2.14×10^{-1}	2.14×10^{-1}
^{12}C	4.32×10^{-2}	4.38×10^{-2}	4.39×10^{-2}
^{15}N	4.05×10^{-2}	4.52×10^{-2}	4.52×10^{-2}
^{16}O	1.77×10^{-4}	1.82×10^{-4}	1.77×10^{-4}
^{20}Ne	6.20×10^{-2}	6.29×10^{-2}	6.28×10^{-2}
^{24}Mg	4.33×10^{-4}	4.53×10^{-4}	4.47×10^{-4}
^{25}Mg	3.00×10^{-2}	2.78×10^{-2}	2.79×10^{-2}
^{27}Al	1.59×10^{-2}	1.76×10^{-2}	1.77×10^{-2}
^{28}Si	5.04×10^{-2}	5.39×10^{-2}	5.41×10^{-2}

Table 4.10: Timesteps in each region for nova simulation

Region	End Time (10^{10} s)	Asymptotic	Backward Euler	Bader-Deuflhard
Build Up	3.578306835786	992	405	397
Outburst	3.578306837331	609	454	454
Early Expansion	3.578306856187	1414	400	398
Late Expansion	3.578311183457	884	887	924

over the longest amount of time, from $t = 0$ to $t = 3.578306835786 \times 10^{10}$ seconds ($\log(t) = 10.55367757809$). The outburst region begins when the temperature rises above 10^8 Kelvin at the end of the accretion phase and continues to the temperature peak at $t = 3.578306837331 \times 10^{10}$ seconds ($\log(t) = 10.55367757828$). The early expansion region is from the temperature peak to when the temperature drops back below 10^8 Kelvin at $t = 3.578306856187 \times 10^{10}$ seconds ($\log(t) = 10.55367758057$). The late expansion phase lasts from there to the end time of the calculation.

For the 169-isotope network used in th nova calculation backward Euler spends 86.8% of its time building and inverting matrices. To be competitive with the implicit methods asymptotic needs to take fewer than $\frac{1}{1-.868} = 7.58$ timesteps for each backward Euler timestep. The asymptotic solver beats this criteria in every thermodynamic region shown in Table 4.10. The asymptotic solver is competitive with the implicit solvers for the classical nova test problem.

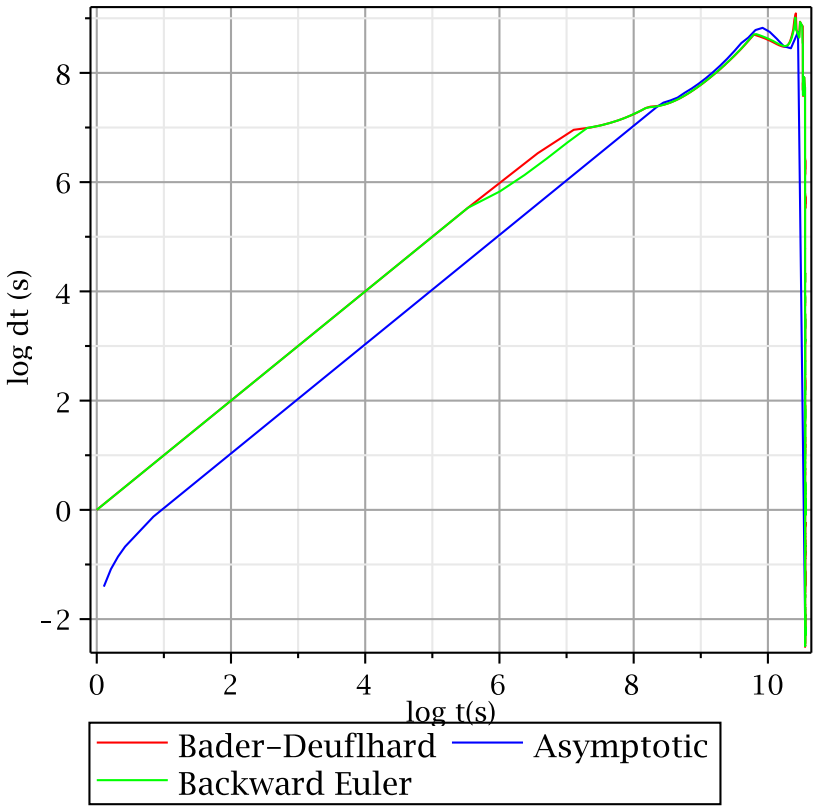


Figure 4.7: Classical nova timestep comparison.

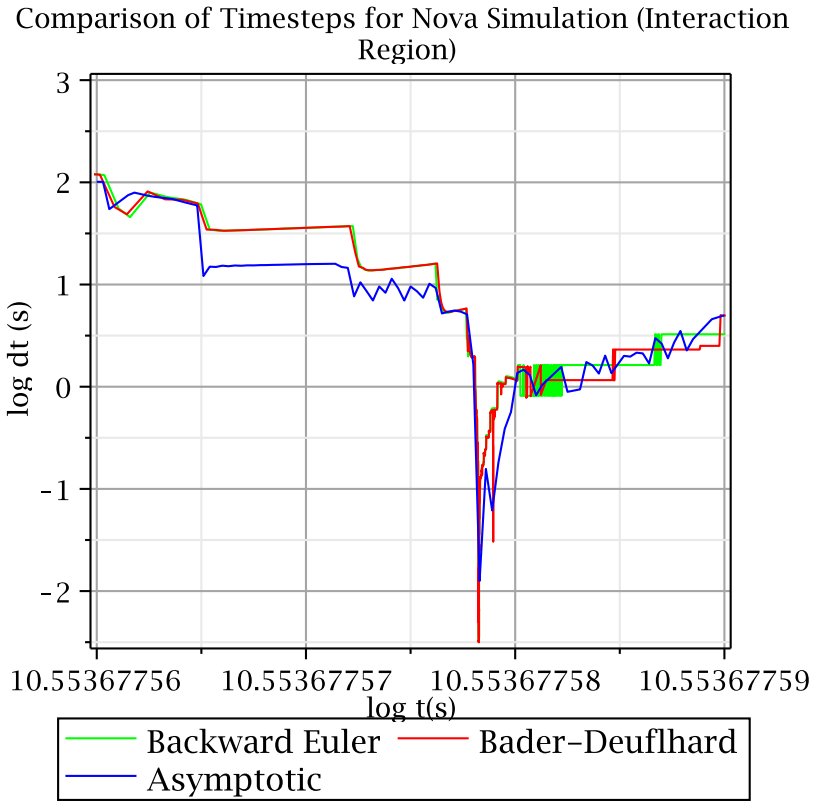


Figure 4.8: Nova timestep comparison focused on the region near the sudden temperature rise.

4.5 Test Problem: Type Ia Supernova

Supernovae are classified into several types. The most overarching categories are type I and type II supernovae, which are classified based on the absence (type I) or presence (type II) of hydrogen lines in their spectra. If a type I supernova also possesses Silicon spectral lines, it is considered a type Ia supernova [42].

The progenitor star of a type Ia supernova is believed to be a carbon-oxygen white dwarf locked in a binary star system with a companion which accretes matter onto the white dwarf [43], either through an active stellar wind or the companion star overfilling one of its Roche lobes [16]. With accretion occurring at a higher rate than in novae, the accreted matter burns steadily while accreting. Thus instead of a hydrogen rich accreted layer, the accreted matter is carbon and oxygen, growing the size of the white dwarf up to the limiting Chandrasekhar mass. Once the white dwarf has accreted enough matter from its companion, a runaway thermonuclear explosion is initiated that causes the supernova [44]. An alternative model triggers the thermonuclear runaway in degenerate matter by the merger of two white dwarfs [45].

Supernova type Ia are of special interest in astronomy because they serve as standardizable candles for measuring distances [46]. Much effort has been put forth to understand the mechanism of type Ia events. Various models are summarized in Ref. [47]. The thermodynamic profile shown in Figure 4.9 is based on the work in Ref. [48].

The first network we examine is the 14-isotope alpha network, which includes ^{4}He, ^{12}C, ^{16}O, ^{20}Ne, ^{24}Mg, ^{28}Si, ^{32}S, ^{36}Ar, ^{40}Ca, ^{44}Ti, ^{48}Cr, ^{52}Fe, ^{56}Ni, and ^{60}Zn. Figure 4.10 shows the time evolution of the nuclear populations from $\log(t) = .1$ to $\log(t) = .3$, highlighting the region near the temperature spike shown in Figure 4.9. Figure 4.11 compares the timesteps taken by the asymptotic, backward Euler, and Bader-Deuflhard solvers for the alpha network in this simulation. The initial mass fractions for the system are $X(^{12}\mathrm{C}) = 0.5$ and $X(^{16}\mathrm{O}) = 0.5$.

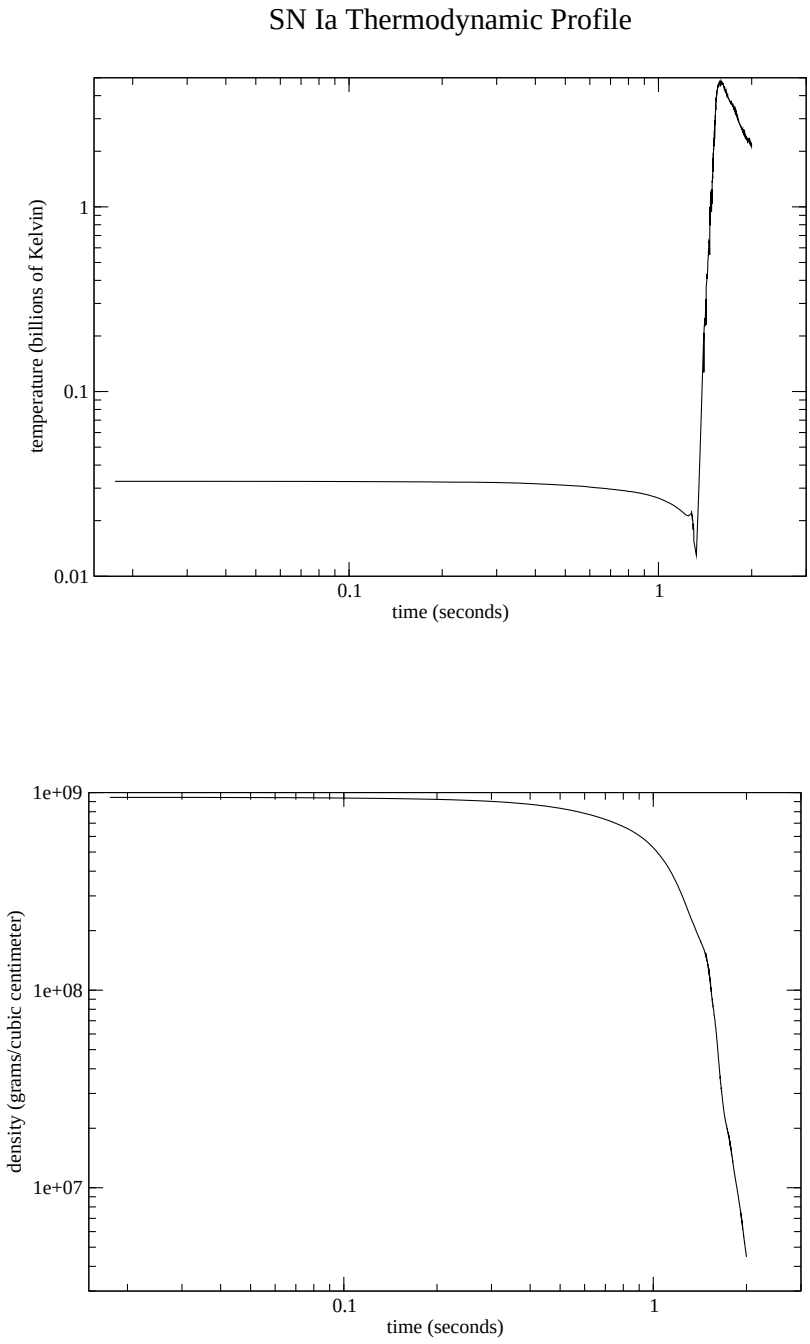


Figure 4.9: The temperature and density profiles for the Type Ia Supernova problem.

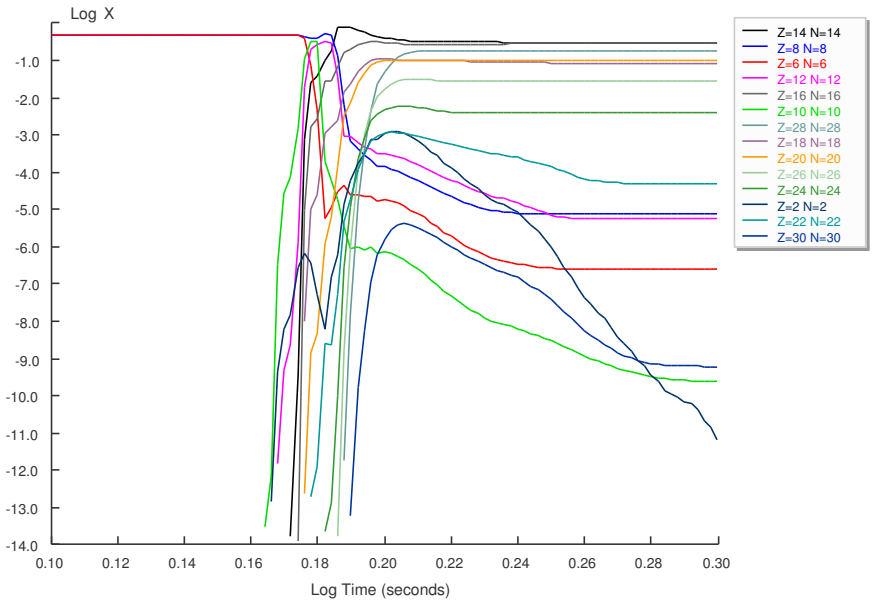


Figure 4.10: Time evolution of the mass fractions for a nuclear reaction network using a 14-isotope alpha network under type Ia supernova conditions.

The final mass fractions for this test problem using each of the three methods is shown in Table 4.11. For the highest population isotopes, those with mass fractions greater than 10^{-3}, the asymptotic algorithm returns results that are within about 10% of the implicit methods. This may seem like a large deviation, however the inputs are known to have errors of 10%-15% (for parameterized reaction rates derived from experimental results) [49]. With that in mind, getting errors of the 10% level or smaller for populations seems acceptable. Less acceptable is the loss in accuracy for the smaller populations. The single largest difference between the asymptotic and the implicit methods is in the mass fraction of ^{44}Ti. The asymptotic mass fraction differs by 32% compared to the backward Euler and 34% compared to the Bader-Deuflhard result. The remaining mass fractions deviate by less than 20%.

For much of the integration the asymptotic solver takes timesteps of the same order of magnitude as the implicit methods. However, in the interaction region between $\log(t) = 0.18$ and $\log(t) = 0.26$ the asymptotic solver lags behind the other two solvers by factors of between 10^3 and 10^5, resulting in the asymptotic taking

Table 4.11: Final mass fractions for the type Ia supernova with an alpha network

Isotope	Asymptotic	Backward Euler	Bader-Deuflhard
^{4}He	6.69×10^{-12}	5.71×10^{-12}	5.72×10^{-12}
^{12}C	2.45×10^{-7}	2.88×10^{-7}	2.74×10^{-7}
^{16}O	7.22×10^{-6}	8.49×10^{-6}	8.15×10^{-6}
^{20}Ne	2.64×10^{-10}	2.65×10^{-10}	2.50×10^{-10}
^{24}Mg	5.06×10^{-6}	6.12×10^{-6}	5.92×10^{-6}
^{28}Si	2.88×10^{-1}	3.15×10^{-1}	3.11×10^{-1}
^{32}S	2.97×10^{-1}	2.95×10^{-1}	2.92×10^{-1}
^{36}Ar	9.00×10^{-2}	8.41×10^{-2}	8.36×10^{-2}
^{40}Ca	1.12×10^{-1}	1.01×10^{-1}	1.01×10^{-1}
^{44}Ti	6.31×10^{-5}	4.77×10^{-5}	4.70×10^{-5}
^{48}Cr	3.90×10^{-3}	3.78×10^{-3}	3.80×10^{-3}
^{52}Fe	2.81×10^{-2}	2.74×10^{-2}	2.79×10^{-2}
^{56}Ni	1.84×10^{-1}	1.74×10^{-1}	1.80×10^{-1}
^{60}Zn	7.07×10^{-10}	5.78×10^{-10}	5.85×10^{-10}

tens of thousands of timesteps for every implicit step. Backward Euler takes 636 timesteps in the interaction region (27% of total timesteps), while the asymptotic takes 4,810,018 timesteps which is more than 99.9% of all timesteps taken by the asymptotic solver. The asymptotic approximation is simply not competitive for this thermonuclear profile.

Next a larger network comprising 150 isotopes (illustrated in Figure 4.12) is computed using the thermodynamic profile shown in Figure 4.9. This larger network includes protons (^{1}H) and neutrons, which typically have faster reaction rates than the reactions in the alpha network used previously. Figure 4.13 shows the time evolution of the mass fractions for this network with initial conditions of $X(^{12}\mathrm{C}) = 0.5$ and $X(^{16}\mathrm{O}) = 0.5$.

The timesteps taken by the asymptotic method are shown in Figure 4.14, along with the timesteps from the backward Euler and Bader-Deuflhard calculations. Backward Euler typically takes timesteps of order $dt = 10^{-3}$ or $dt = 10^{-4}$ seconds. The Bader-Deuflhard solver was quite noisy with its timesteps fluctuating between $dt = 10^{-3}$ and $dt = 10^{-10}$ seconds, but the smallest of timesteps are rare with

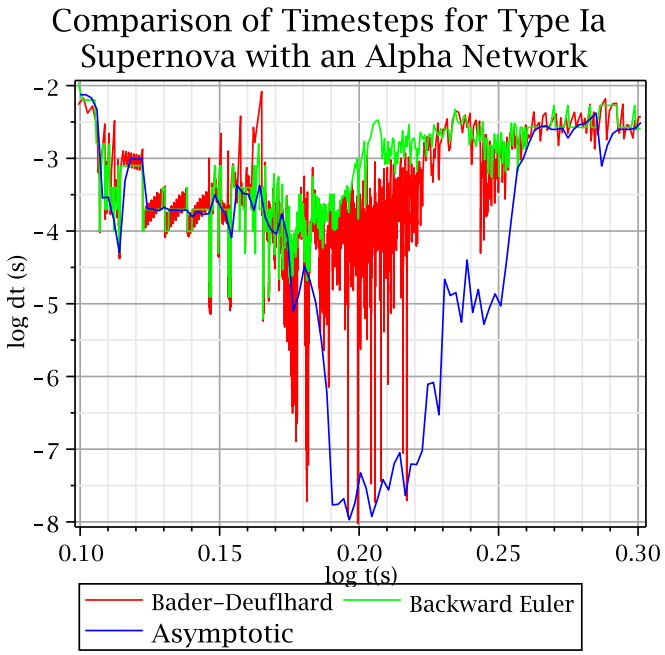


Figure 4.11: Comparison of the timesteps taken by the asymptotic solver, Bader-Deuflhard, and backward Euler for the 14-isotope alpha network under type Ia supernova conditions.

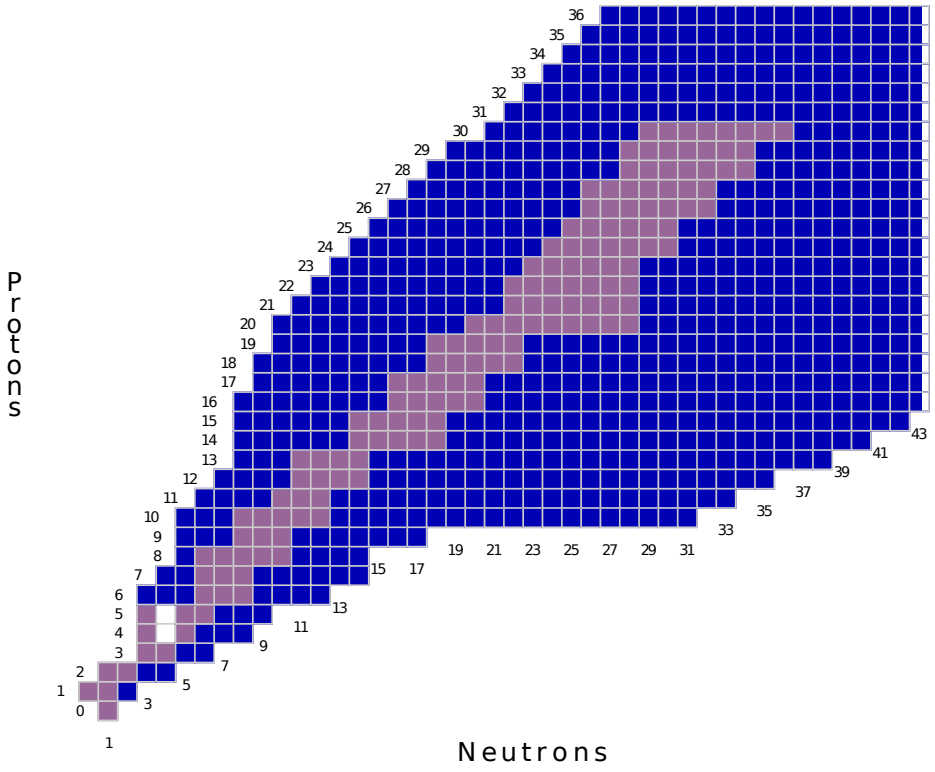


Figure 4.12: Overview of the 150 isotope nuclear network named "SN150" in Table 4.4 and used for the tidally induced supernova and type Ia supernova test problems.

timesteps of $dt = 10^{-4}$ being the norm. As the Bader-Deuflhard solver was newly implemented for this work, we are less familiar with its behavior and less certain that the implementation is optimal. The asymptotic algorithm took timesteps typically five to seven magnitudes smaller than the backward Euler solver in the region near the temperature spike. Where the implicit solvers take thousands of timesteps (2579 for backward Euler and 6140 for Bader-Deuflhard), the asymptotic solver takes hundreds of millions of timesteps. The asymptotic solver is not competitive for the type Ia supernova problem with a 150-isotope network.

The results are much the same for a 299-isotope network (illustrated in Figure 4.15) using the same type Ia thermodynamic profile as in Figure 4.9. This case has initial conditions of $X(^{12}\mathrm{C}) = 0.495$, $X(^{16}\mathrm{O}) = 0.495$ and $X(^{22}\mathrm{Ne}) = 0.01$. The mass fractions are shown in Figure 4.16.

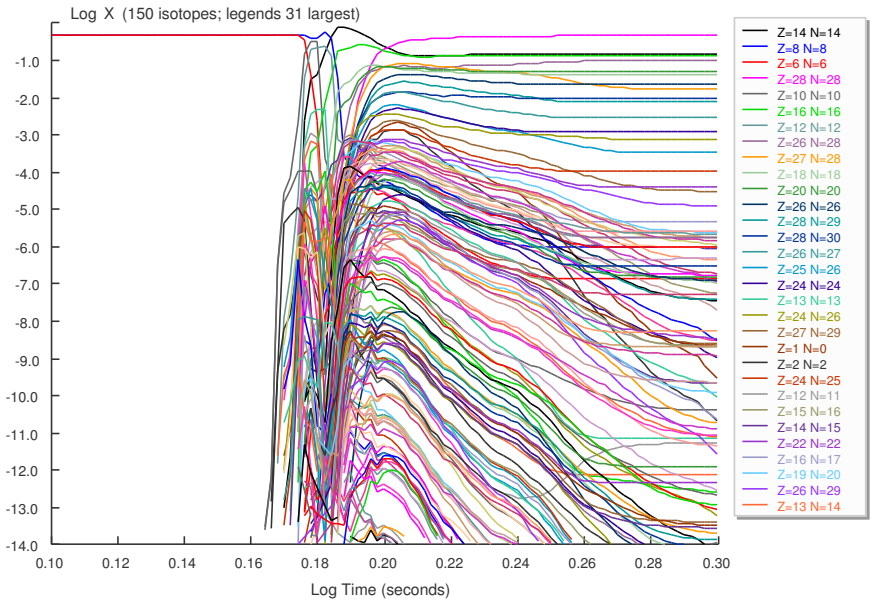


Figure 4.13: Time evolution of the mass fractions for a nuclear reaction network using a 150-isotope network under type Ia supernova conditions.

The timesteps for all three methods (asymptotic, backward Euler, and Bader-Deuflhard) are shown in Figure 4.17 for the region between $\log(t) = .1$ and $\log(t) = .3$. The timestep behavior is quite similar to that found in the 150-isotope network examined above, with the asymptotic algorithm taking hundreds of millions of timesteps in comparison to the implicit solvers taking thousands of timesteps. A factor of 10^5 difference in the number of timesteps taken means that even for this larger network the asymptotic algorithm is not competitive with the implicit techniques.

For these type Ia simulations, the asymptotic solver is unable to take competitive timesteps. We speculate that the failure of the asymptotic approximation for this problem is due to forward and inverse rates coming into partial equilibrium, which will be discussed in Section 4.10.

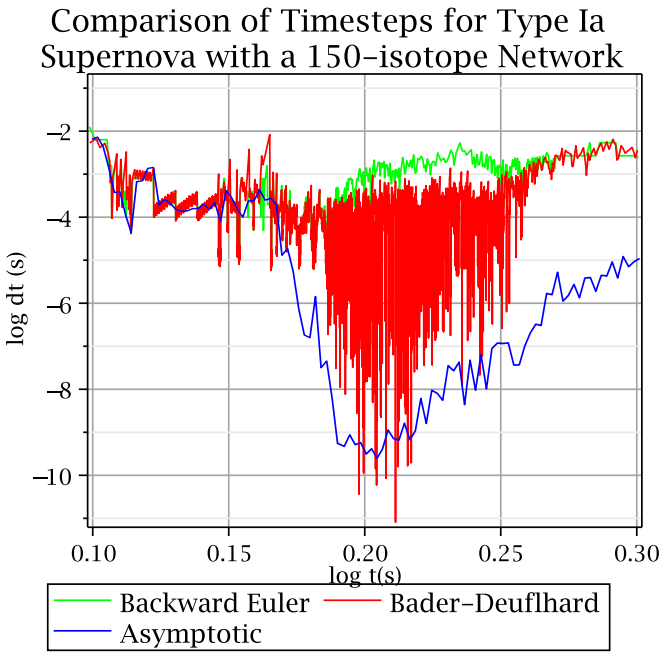


Figure 4.14: Comparison of the timesteps taken by the asymptotic solver, Bader-Deuflhard, and backward Euler for a 150-isotope network.

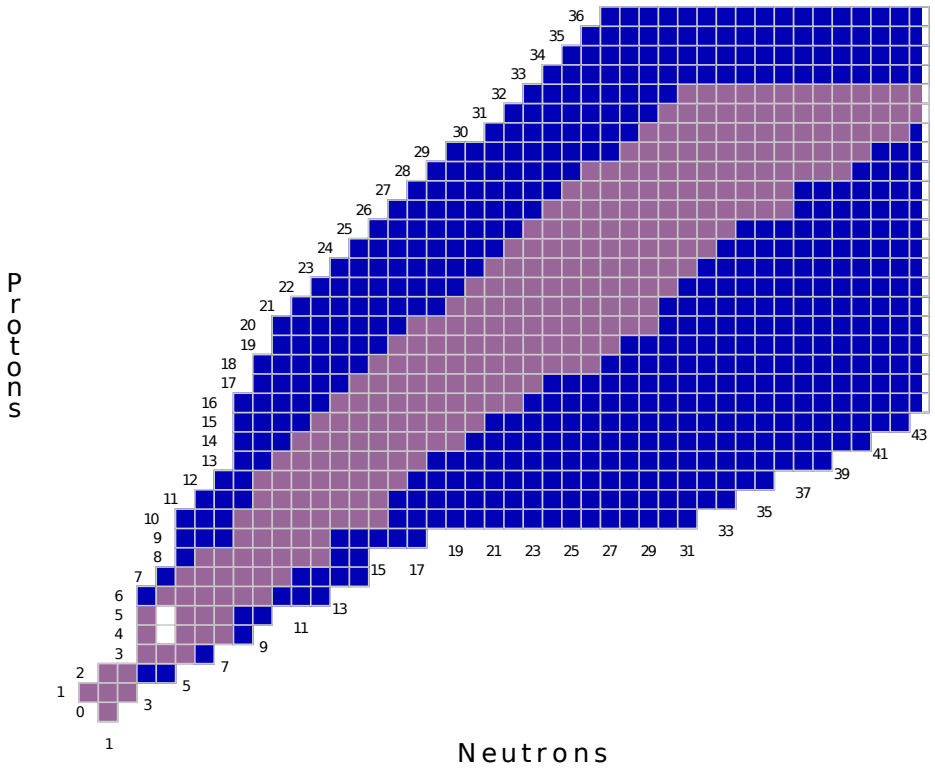


Figure 4.15: Overview of the 299 isotope nuclear network referred to as "Si02_fix" in Table 4.4 used for the tidally induced supernova and type Ia supernova test problems.

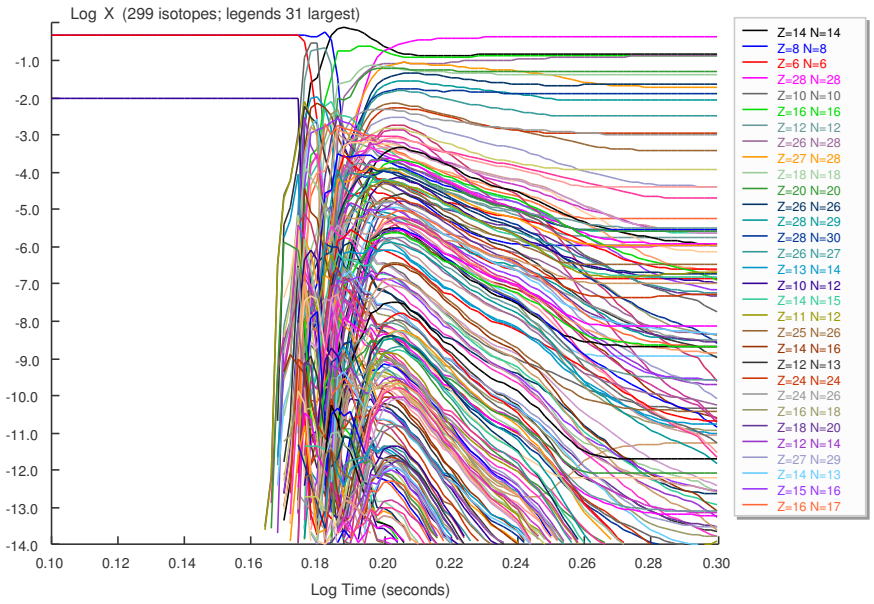


Figure 4.16: Time evolution of the mass fractions for a nuclear reaction network using a 299-isotope network under type Ia supernova conditions.

4.6 Test Problem: Tidally Induced Thermonuclear Supernova

It is suspected that some globular clusters have black holes of intermediate-mass ($10^2 - 10^5$ $M_\odot$) within them. In the presence of such a black hole, a star in the cluster may drift close enough to the black hole that the black hole exerts strong tidal effects upon the star. If the star is a white dwarf, the tidal effect will elongate the white dwarf in one direction and crush it in the other two. This strong deformation increases the temperature in the degenerate matter of the white dwarf and triggers explosive burning.

In Ref [50], Rosswog, Ramirez-Ruiz, and Hix considered the case of a 0.2 $M_\odot$ white dwarf composed initially of ^{4}He. They found in their simulation that detonation did occur and the star reached peak temperatures in excess of $T = 2 \times 10^9$ K. About 15% of the white dwarf's mass underwent nuclear burning into iron-group nuclei, and according to Ref [51] about 35% of the white dwarf's mass remains gravitationally

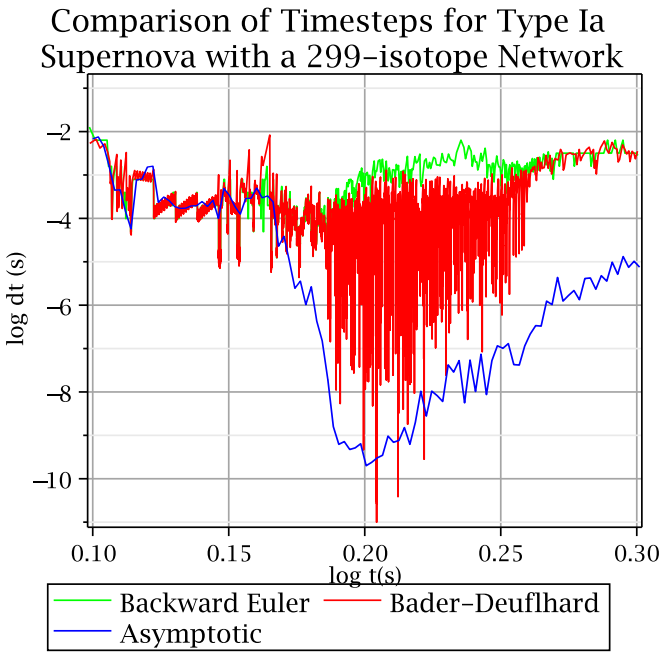


Figure 4.17: Comparison of the timesteps taken by the asymptotic solver, Bader-Deuflhard, and backward Euler for a 299-isotope network.

bound to the black hole (assumed to be 1000 $M_\odot$) and will be accreted onto it, the remaining matter is ejected.

In some ways these tidally induced thermonuclear supernovae are similar to type Ia supernovae; both are caused by a thermonuclear runaway. A key difference is that a tidally induced supernova is not limited to stellar masses close to the Chandrasekhar limit. The example concentrated on in Ref. [50] has a mass of only 0.2 $M_\odot$ and it is mentioned that they find detonations over a range of masses from 0.2 $M_\odot$ to 1.2 $M_\odot$, including at the peak of the observed distribution of white dwarfs at 0.6 $M_\odot$ [52].

The initial composition used for this test problem is composed entirely of ^{4}He. Figure 4.18 shows the thermodynamic profile for temperature and density followed by the simulation. The temperature starts at about 1.0×10^7 K, rises to a peak of 2.785×10^9 K, and then slowly cools down. Three different networks were used to simulate the conditions of a tidally induced supernova: a 14-isotope alpha network, a 150-isotope network, and a 299-element network.

The first test network for the tidally induced supernova is a 14-element alpha network consisting of ^{4}He, ^{12}C, ^{16}O, ^{20}Ne, ^{24}Mg, ^{28}Si, ^{32}S, ^{36}Ar, ^{40}Ca, ^{44}Ti, ^{48}Cr, ^{52}Fe, ^{56}Ni, and ^{60}Zn. Figure 4.19 shows the time evolution of the mass fractions for an alpha network under the tidally induced supernova conditions given in Figure 4.18. After the initial period of burning, which converts ^{4}He into heavier elements, the calculation enters a period where the populations are essentially frozen.

Table 4.12 shows the mass fractions at the end of the simulation for the asymptotic, backward Euler, and Bader-Deuflhard algorithms. The largest difference between the asymptotic and the implicit methods is in the populations of ^{36}Ar and ^{40}Ca with 5.4% difference when the asymptotic is compared to the backward Euler calculation. The difference in the population of ^{32}S between asymptotic and backward Euler is the next highest difference in the populations, differing by 4.7%. The asymptotic matches Bader-Deuflhard slightly better in general than it matches backward Euler, with percent differences of 3.8%, 4.4%, and 4.7% for ^{32}S, ^{36}Ar, and ^{40}Ca respectively. The difference between asymptotic and backward Euler for the remaining isotopes is

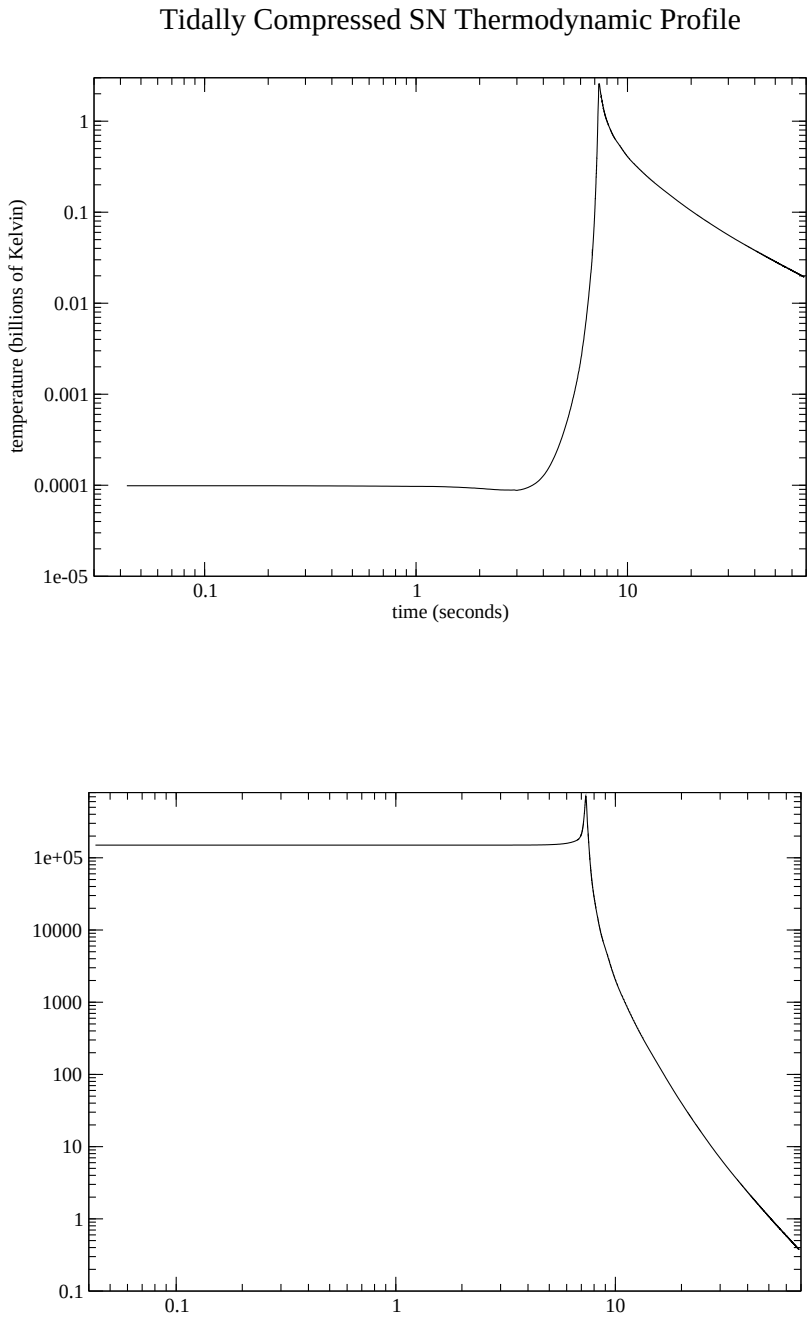


Figure 4.18: The temperature and density profiles for the Tidally Induced Thermonuclear Supernova problem.

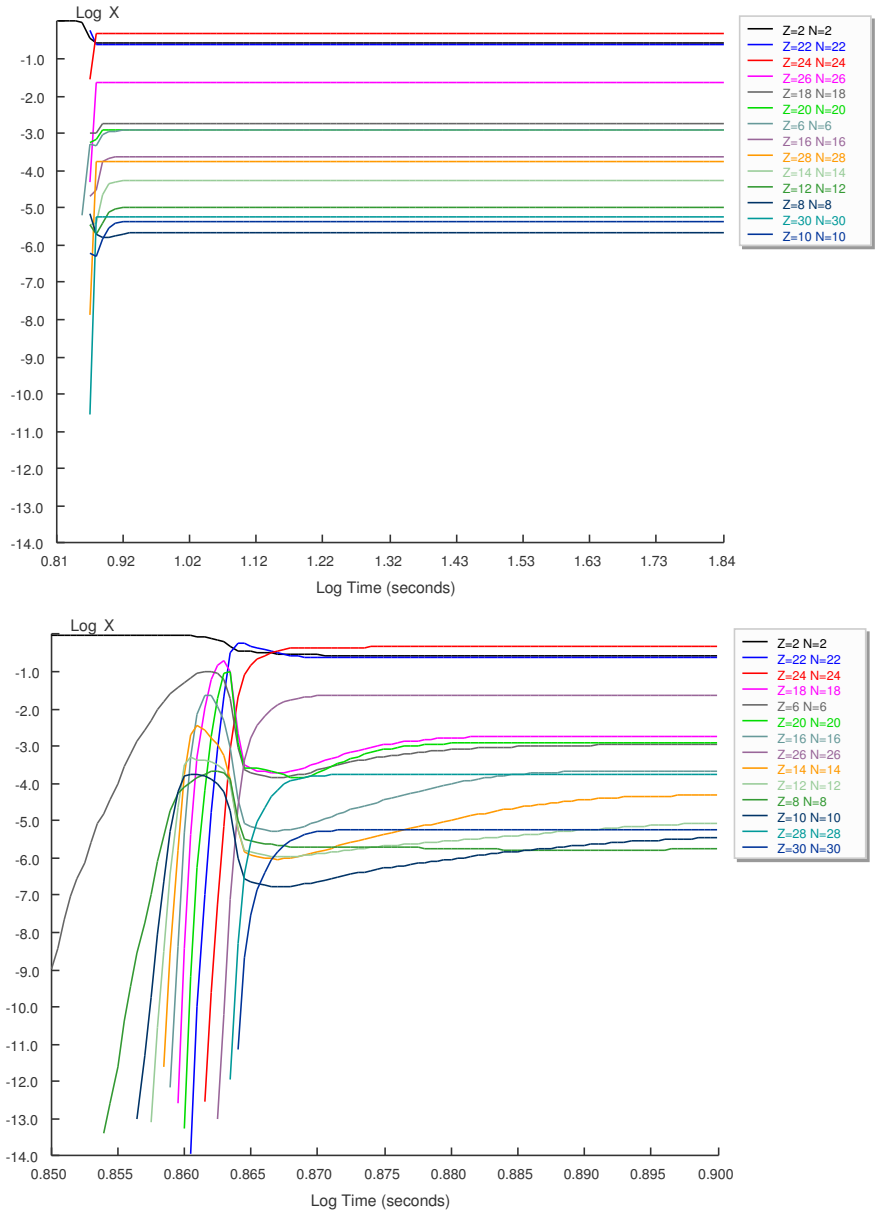


Figure 4.19: Time evolution of the mass fractions for a nuclear reaction network using a 14-isotope alpha network under tidally induced supernova conditions. The top shows the mass fractions over the whole range, while the bottom is focused on the area near the temperature spike.

Table 4.12: Final mass fractions for the tidally induced supernova with an alpha network

Isotope	Asymptotic	Backward Euler	Bader-Deuflhard
^{4}He	2.68×10^{-1}	2.72×10^{-1}	2.70×10^{-1}
^{12}C	1.15×10^{-3}	1.18×10^{-3}	1.17×10^{-3}
^{16}O	2.03×10^{-6}	2.09×10^{-6}	2.07×10^{-6}
^{20}Ne	4.23×10^{-6}	4.33×10^{-6}	4.30×10^{-6}
^{24}Mg	9.96×10^{-6}	1.02×10^{-5}	1.01×10^{-5}
^{28}Si	4.94×10^{-5}	5.06×10^{-5}	5.04×10^{-5}
^{32}S	2.13×10^{-4}	2.24×10^{-4}	2.22×10^{-4}
^{36}Ar	1.79×10^{-3}	1.90×10^{-3}	1.88×10^{-3}
^{40}Ca	1.20×10^{-3}	1.27×10^{-3}	1.26×10^{-3}
^{44}Ti	2.46×10^{-1}	2.48×10^{-1}	2.44×10^{-1}
^{48}Cr	4.60×10^{-1}	4.53×10^{-1}	4.57×10^{-1}
^{52}Fe	2.32×10^{-2}	2.29×10^{-2}	2.31×10^{-2}
^{56}Ni	1.69×10^{-4}	1.71×10^{-4}	1.69×10^{-4}
^{60}Zn	5.55×10^{-6}	5.74×10^{-6}	5.60×10^{-6}

below 3% in all cases and the difference between asymptotic and Bader-Deuflhard is below 2%. The difference between the backward Euler and Bader-Deuflhard answers is on the order of 1% to 2%. With the exception of the three named isotopes above, the asymptotic approximation gives results within the same level of difference as exists between the standard backward Euler and Bader-Deuflhard methods.

The asymptotic, backward Euler, and Bader-Deuflhard calculations all take comparable timesteps illustrated in Figure 4.20. All three methods decrease their timestep when the temperature rises, starting near $t = 7.16$ seconds ($\log(t) = 0.855$). The temperature peaks at $t = 7.33$ seconds ($\log(t) = 0.865$) and all three methods begin to increase their timestep as the temperature cools. The asymptotic algorithm lags behind initially, but by $t = 7.59$ seconds ($\log(t) = 0.88$) it has caught back up to backward Euler. The timestep taken by asymptotic is within a factor of 2 or 3 of the implicit methods.

Over the course of the entire calculation the asymptotic solver takes 3,405 timesteps, backward Euler took 2,186 timesteps, and Bader-Deuflhard took 2,637 timesteps. Backward Euler spends 63.2% of its time performing linear algebra

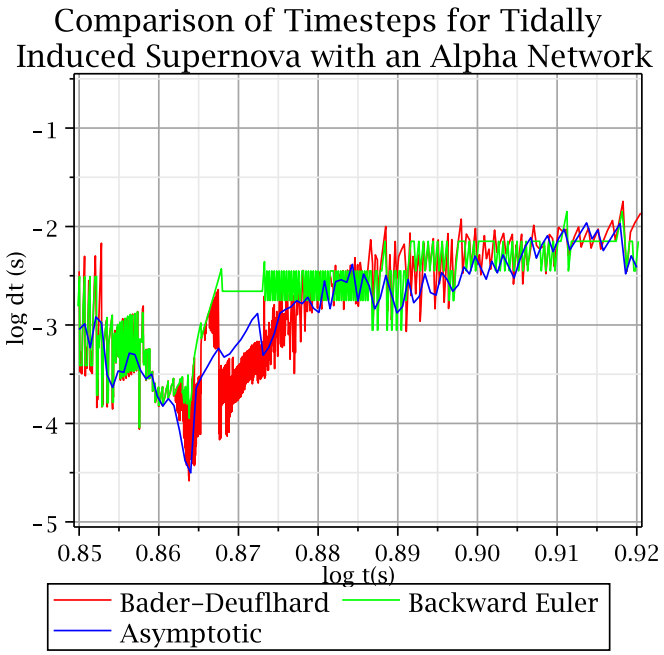


Figure 4.20: Comparison of the timesteps taken by the asymptotic, Bader-Deuflhard, and backward Euler solvers for the 14-isotope alpha network focused around the interaction region.

Table 4.13: Timesteps in each region for tidally induced supernova with an alpha network

Region	Time Interval (s)	Asymptotic	Backward Euler	Bader-Deuflhard
Build Up	6.8-7.16	679	629	539
Outburst	7.16-7.33	1399	585	776
Early Expansion	7.33-7.59	341	119	484
Late Expansion	7.59-68.6	986	853	818

inversions on a network of this size. To be competitive with the implicit solvers, the asymptotic approximation can at most take twice as many timesteps as backward Euler takes. The 3,405 timesteps taken overall is smaller than the 5,940 timestep maximum for competitiveness.

Table 4.13 breaks down the timesteps taken into four regions. In the earliest "build up" region where the temperature steadily rises, the asymptotic solver takes 7.9% more timesteps than backward Euler and 26% more timesteps than Bader-Deuflhard. The outburst region where most of the isotopic evolution occurs is also where the asymptotic solver takes the largest number of timesteps, taking 139% more timesteps than backward Euler. In the early expansion region the asymptotic solver takes three times as many timesteps as backward Euler, but fewer than Bader-Deuflhard. In the final late expansion region, which takes place over the longest timescale, the asymptotic solver takes 16% more timesteps than backward Euler and 21% more timesteps than Bader-Deuflhard.

Even in the early expansion region, where the asymptotic solver behaves the worst compared to backward Euler, the asymptotic solver is only taking 2.86 timesteps for each implicit Euler timestep. To be competitive here the asymptotic solver can take at most $\frac{1}{1-.632} = 2.72$ timesteps for each backward Euler timestep. The asymptotic solver slightly underperforms in this region compared to backward Euler, but it compares well to the number of timesteps taken by Bader-Deuflhard.

The outburst region is where asymptotic takes the largest portion of its timesteps. Here it takes 2.39 timesteps for each backward Euler timestep, below the 2.72

Table 4.14: Timesteps in each region for tidally induced supernova with a 150-isotope network

Region	Time Interval (s)	Asymptotic	Backward Euler	Bader-Deuflhard
Build Up	6.8-7.16	680	629	563
Outburst	7.16-7.33	6935	758	761
Early Expansion	7.33-7.59	11297	169	181
Late Expansion	7.59-68.6	1138	872	862

timesteps needed to be competitive. In the two remaining regions the asymptotic solver takes few enough timesteps that it should easily be competitive. Overall, the asymptotic is competitive with the implicit methods for this case.

The second test network is a 150-isotope network which includes elements up to Zinc. The heaviest isotope included is ^{66}Zn. The biggest change between the 150-isotope network and an alpha network is the inclusion of neutrons and protons with the consequent fast neutron and proton reactions. Figure 4.12 gives an overview of the 150-isotope network.

Figure 4.21 shows the mass fractions as they evolve over the integration time. The most noticeable difference between evolution of mass fractions in Figure 4.21 and the results from the 14-isotope network shown in Figure 4.19, aside from the additional populations tracked, is the inclusion of weak reactions such as β-decays which continue to operate over longer timescales even after nuclear population changes had "frozen out" in the 14-isotope alpha network. The influence of these weak rates can be see in Figure 4.21 where several populations continue to evolve over the entire length of the simulation.

Backward Euler spends about 86.8% of its time building and solving the Jacobian matrix for this network. For the asymptotic solver to be competitive it should take no more than $\frac{1}{1-.868} = 7.58$ timesteps for each backward Euler timestep. Figure 4.22 shows dt vs t for each of the three solvers and Table 4.14 contains the number of timesteps taken for each of the methods broken down into several regions.

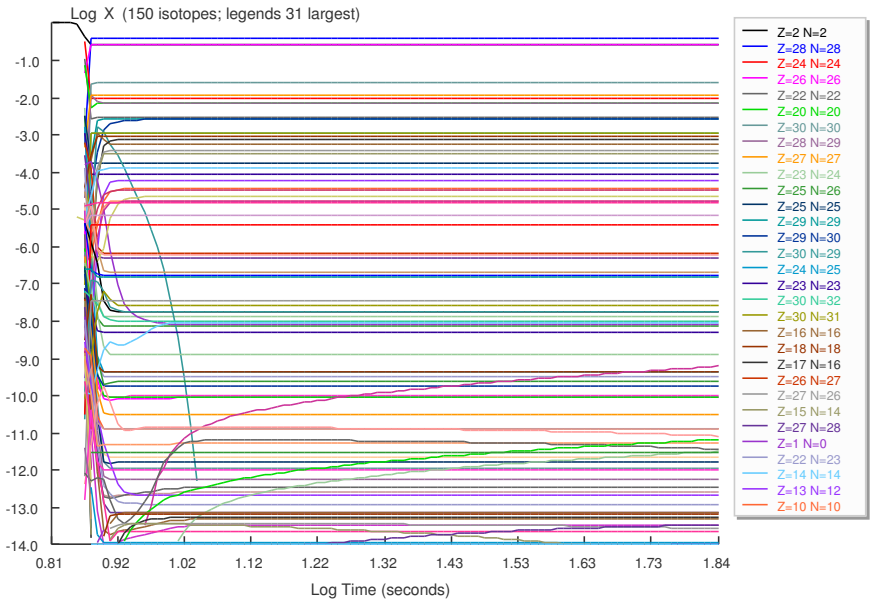


Figure 4.21: Time evolution of the mass fractions for the backward Euler (top) and asymptotic (bottom) calculations of the nuclear reaction network using a 150-isotope network under tidally induced supernova conditions.

The asymptotic solver takes 8.1% and 30% more timesteps than backward Euler for the build up and late expansion regions respectively. In the outburst region, the asymptotic solver takes 9.15 timesteps for each backward Euler timestep, which underperforms by 21% compared to the limit of 7.58 timesteps for each implicit timestep to be competitive. In the early expansion phase, the asymptotic solver takes 66.8 timesteps for each backward Euler timestep. Taken as an overall picture, the asymptotic solver underperforms only slightly, taking 20,050 timesteps total compared to the 18,404 limit for competitiveness because its good behavior in the build up and late expansion regions compensates somewhat for all of the timesteps spent in the early expansion region.

The third network investigated is a 299-isotope network. Figure 4.15 illustrates the 299-isotope network for which the populations, represented as mass fractions, are computed and displayed in Figure 4.23. This network is similar to the 150-isotope network previously examined, the main difference is in its expanded size. It can

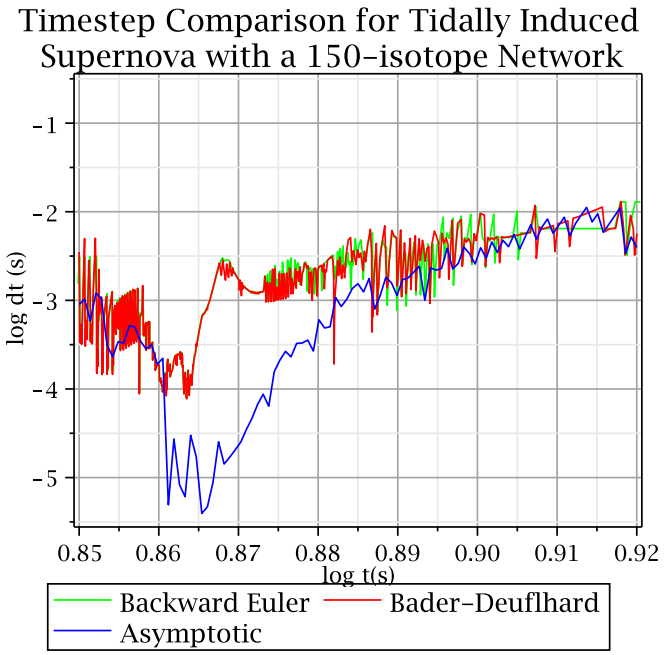


Figure 4.22: Comparison of the timesteps taken by the asymptotic solver, Bader-Deuflhard, and backward Euler for a 150-isotope network under tidally induced supernova conditions focused around the interaction region.

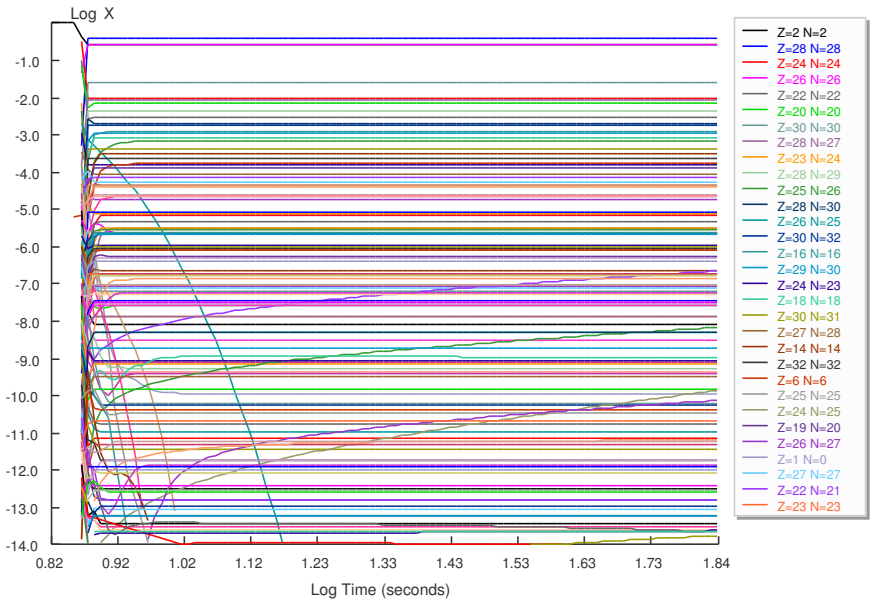


Figure 4.23: Time evolution of the mass fractions for a nuclear reaction network using a 299-isotope network under tidally induced supernova conditions.

Table 4.15: Timesteps in each region for tidally induced supernova with a 299-isotope network

Region	Time Interval (s)	Asymptotic	Backward Euler	Bader-Deuflhard
Build Up	6.8-7.16	680	629	563
Outburst	7.16-7.33	10262	805	808
Early Expansion	7.33-7.59	10882	134	159
Late Expansion	7.59-68.6	1143	866	827

include isotopes which are not populated as heavily as those selected for the 150-isotope network.

The 299-isotope network behaves very similarly to the 150-isotope network, taking similar timesteps in most regions. Table 4.15 contains the number of timesteps taken during the simulation for each of the three solver methods. Figure 4.24 shows the size of the timesteps taken, dt.

For the 299-isotope network, the backward Euler solver spent 93.9% of its time building and inverting matrices. The asymptotic solver needs to take fewer than 16.4 timesteps for each backward Euler timestep to be competitive with the standard

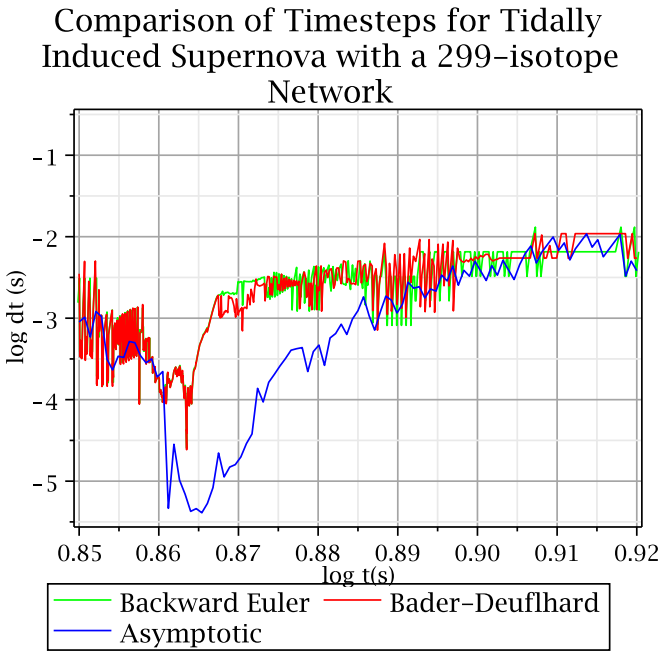


Figure 4.24: Comparison of the timesteps taken by the asymptotic solver, Bader-Deuflhard, and backward Euler for the 299-isotope network under tidally induced supernova conditions focused around the interaction region.

implicit techniques. The overall picture, with asymptotic taking 22,977 timesteps compared to the 2,434 taken by backward Euler and 2,357 timesteps taken by Bader-Deuflhard suggests that the asymptotic algorithm is competitive with implicit techniques for this size network under these conditions, even though asymptotic underperformed for the smaller 150-isotope network examined previously.

In the build up region the asymptotic approximation took 8.1% more timesteps than backward Euler and 20% more timesteps than Bader-Deuflhard, making it clearly competitive in this region. The outburst region is also competitive; here asymptotic takes 12.7 times more steps than backward Euler, smaller than the 16.4 timestep limit. For the early expansion region, the asymptotic approximation is not competitive at all, taking 81.2 timesteps for every backward Euler timestep. In the late expansion region, the asymptotic approximation takes 31.9% more timesteps than backward Euler and 38.2% more timesteps than Bader-Deuflhard. For three of the four regions, the asymptotic solver is competitive with the implicit solvers.

4.7 Test Problem: Type II Supernova

Type II supernovae are caused by the collapse of the central core of a massive star (masses greater than $\sim 8M_{\odot}$). As the star evolves, the burning leaves behind a progression of heavier elements in its core. Eventually the core becomes composed of elements at or near iron in the periodic table. Iron is at the peak of the nuclear binding energy curve, thus fusing iron to make heavier elements requires energy input into the system [53]. Thus, unable to release more energy via fusion processes and resist gravitational collapse, the core starts to shrink. The eventual outcome of the collapse is a neutron star or black hole [54][55][56].

As the core collapses, the outer layers of the star fall inwards, heating up. New burning may ignite in these outer layers during that time. Eventually the outer layers collide with the collapsed core, rebounding and producing a shockwave. This shockwave causes the explosion associated with a supernova. There are many processes active during this time, e.g. electron capture, neutrino trapping in the superdense core, neutrino oscillation [57], which make for a very dynamic environment [58].

The high temperatures in the core ejecta and the expansion of the material lifts the electron degeneracy and creates a proton-rich matter composition. The neutrino-rich environment then drives the νp-process, where antineutrinos capture onto protons and create neutrons. The neutrons are then readily absorbed by nuclei with $N \sim Z$, such as alpha-nuclei like ^{64}Ge [59].

Figure 4.25 illustrates the size of the 1072 isotope nuclear network used in this simulation. Figure 4.26 shows the thermodynamic temperature and density profile used. The REACLIB library is not valid at temperates above $T = 1 \times 10^{10}$ K. For that reason, the simulation takes as its start time $t = 0.342$ seconds, which is after the highest temperatures in the profile.

The simulation has an initial abundance of protons and neutrons, with initial mass fractions $X(\text{protons}) = 0.35$ and $X(\text{neutrons}) = 0.65$. Several attempts were made

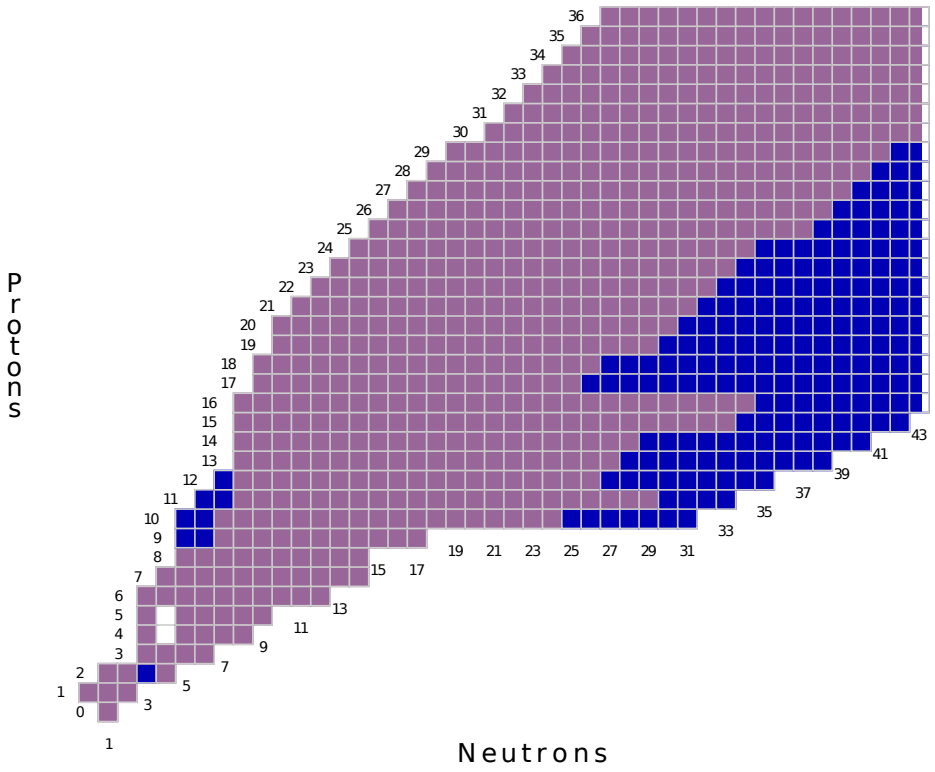


Figure 4.25: Illustration of the 1072 isotope network used in the Type II Supernova νp-process calculation. The network extends considerably above the region shown, up to ^{182}Xe as its heaviest isotope.

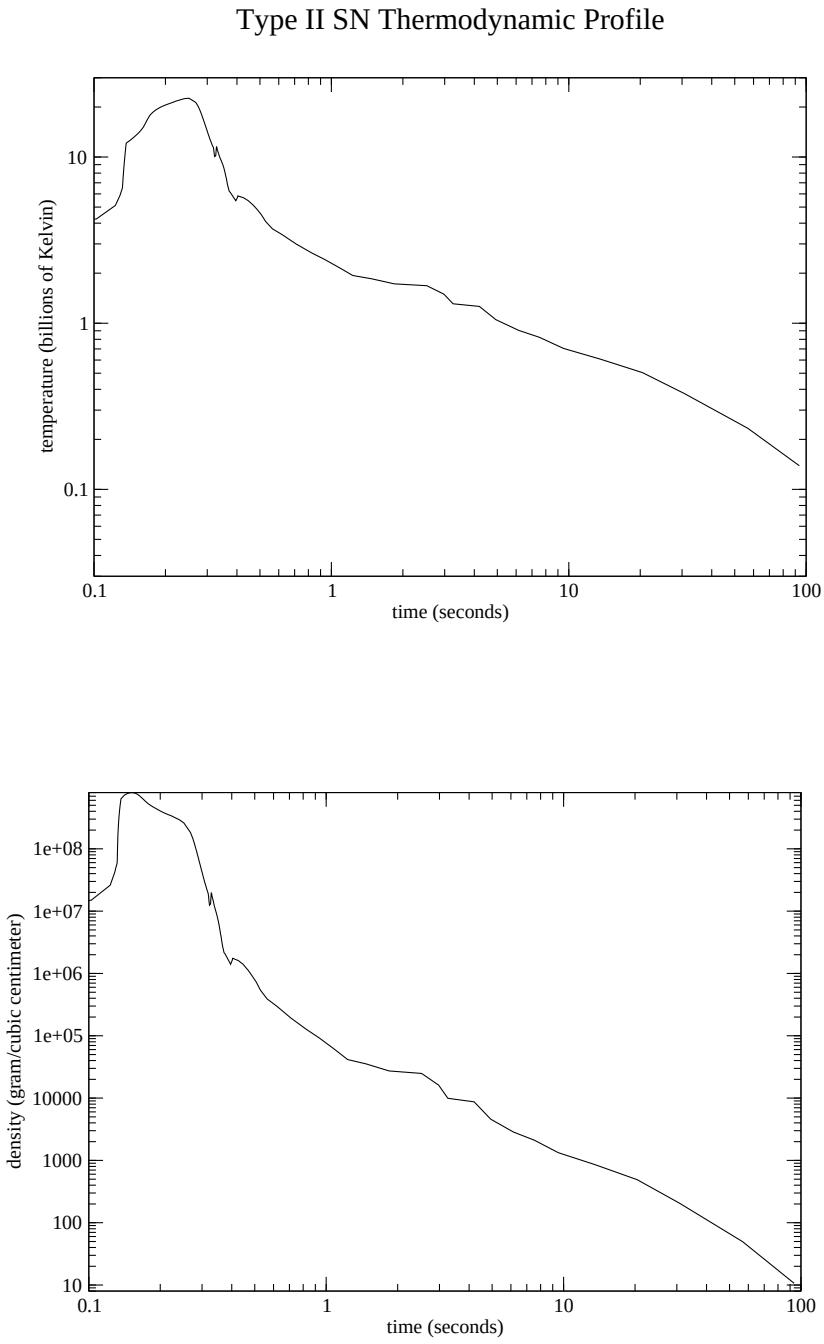


Figure 4.26: The temperature and density profiles for the Type II Supernova problem.

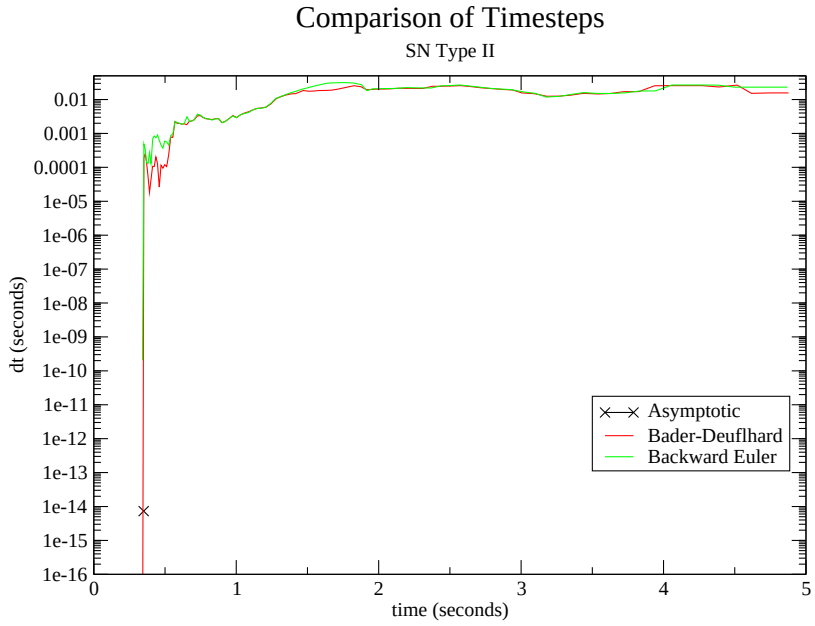


Figure 4.27: Comparison of timesteps for the Type II Supernova simulation. The asymptotic timestep is denoted with an X.

to start the asymptotic integration in the time region near $t = 0.342$ seconds, but each attempt yielded a very small timestep ($dt \sim 10^{-15}$). Figure 4.27 compares the backward Euler and Bader-Deuflhard timesteps to the asymptotic timestep for one of these attempts. As the temperature and density are relatively constant over the time region from $t = 0.35$ to $t = 0.36$ it's a safe approximation to assume that the timestep taken by the network will remain constant over that interval. The asymptotic calculation in Figure 4.27, on the processor it was executing, would have needed to take an estimated 2×10^{12} timesteps to advance to a time of $t = 0.36$ seconds. At a rate of 2×10^5 timesteps in 18 minutes, to advance to $t = 0.36$ seconds would take an estimated 342 years. As this is just a small range of the entire calculation, it is safe to say that integrating the network over the entire simulation time would take even longer. The asymptotic approximation is unsuited to tackle these initial conditions.

4.8 Test Problem: Type I X-Ray Burst

Neutron stars are very compact stellar objects which are the end result of stellar evolution for stars with masses greater than 8-10 Solar masses. Due to having a radius of about 10 km while still having a mass of around 1.4 Solar masses, the surface gravity of a Neutron star is many billions of times as strong as that of the gravity at the surface of the Earth. Neutron stars have been observed due to their radiation emissions; many are radio pulsars while others emit x-rays [60].

Two groups discovered x-ray bursts late in 1975, one was Grindlay and Heise [61] and the other was Belian, Conner, and Evans [62]. There was a flurry of speculation as to the nature of the radiation bursts as well as several discoveries of additional objects [63]. The next year Woosley and Taam identified the source of cosmic x-ray bursts as thermonuclear detonations on neutron stars [64]. These thermonuclear explosions release short, 10-100 seconds long, but very intense bursts of x-rays [65][66]. The fuel for the explosion comes from the neutron star's companion which loses matter to the gravity of the neutron star. This matter slowly accretes onto the neutron star. Once sufficient amounts of matter have been accumulated, fusion burning begins, grows unstable, and causes a thermonuclear explosion on the surface of the neutron star which releases a burst of x-rays with an energy release of $10^{39} - 10^{40}$ ergs.

Early studies such as Hansen and van Horn's focused on hydrogen burning and lower mass isotopes [67]. Newer studies such as that by Fisker, Schatz, and Thielemann [68] or Thielemann, et. al. [69] have included larger and more robust nuclear reaction networks, providing for more realistic calculations. Ref. [67] also neglected hydrodynamic effects, while Ref. [68] and Rembges, et. al. [70], include hydrodynamic models of varying complexity. Rembges found that there can be a second burst after the initial, and speculated that this could be from partially-burned matter that was ejected and then fell back onto the neutron star and reignited. The detected signals from these bursts have been sometimes found to oscillate, suggesting

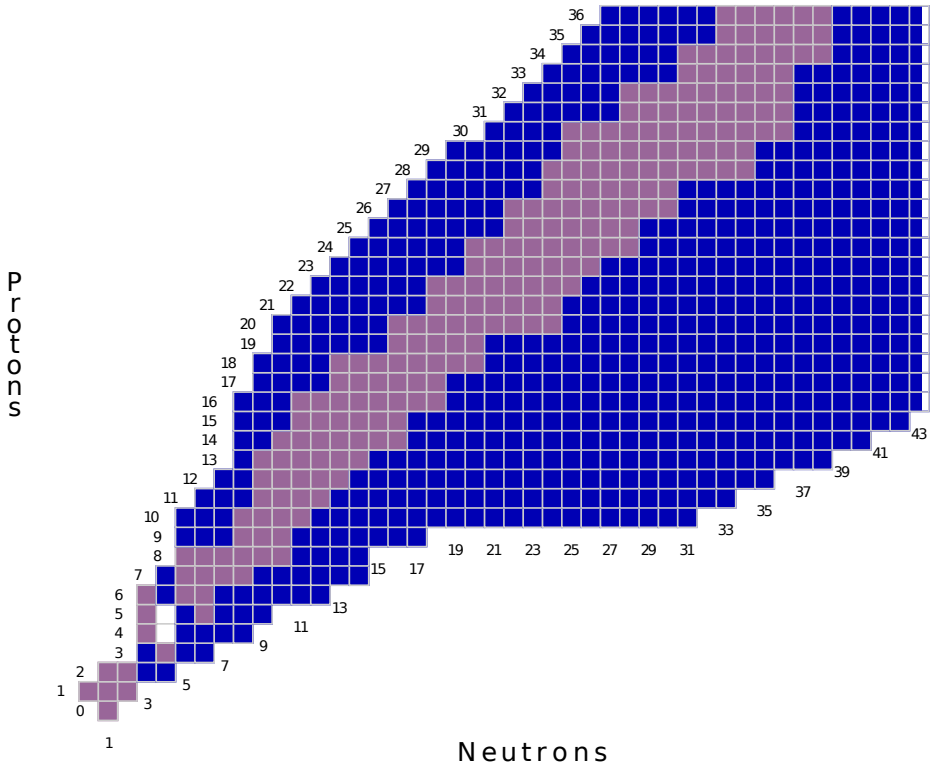


Figure 4.28: Illustration of the nuclear network up to isotopes of $Z = 36$ for the x-ray burst simulation.

that the x-ray bursters have some rotational angular momentum [71]. Psaltis [72] and Lewin [63] give observational details of some x-ray bursts.

Given that accurate simulations of x-ray bursts depend on both good hydrodynamic models and good nuclear reaction models, an improvement in one of those domains will produce a calculation with a more realistic final outcome. This test utilized a 304-isotope network (see Figure 4.28) with an XRB profile based on Fisker's work (see Figure 4.29). The temperature starts at 10^7 Kelvin and increases to a peak just under 10^9 Kelvin before cooling back down again. The density begins at 10^3 g/cm^3 and increases to 1.13x10^6 g/cm^3, with a sudden but short term drop in density occurring in correlation to the rise in temperature. The 304-isotope network begins with hydrogen and neutrons and includes isotopes up to antimony and tellurium (^{106}Sb and ^{107}Te being the heaviest two isotopes).

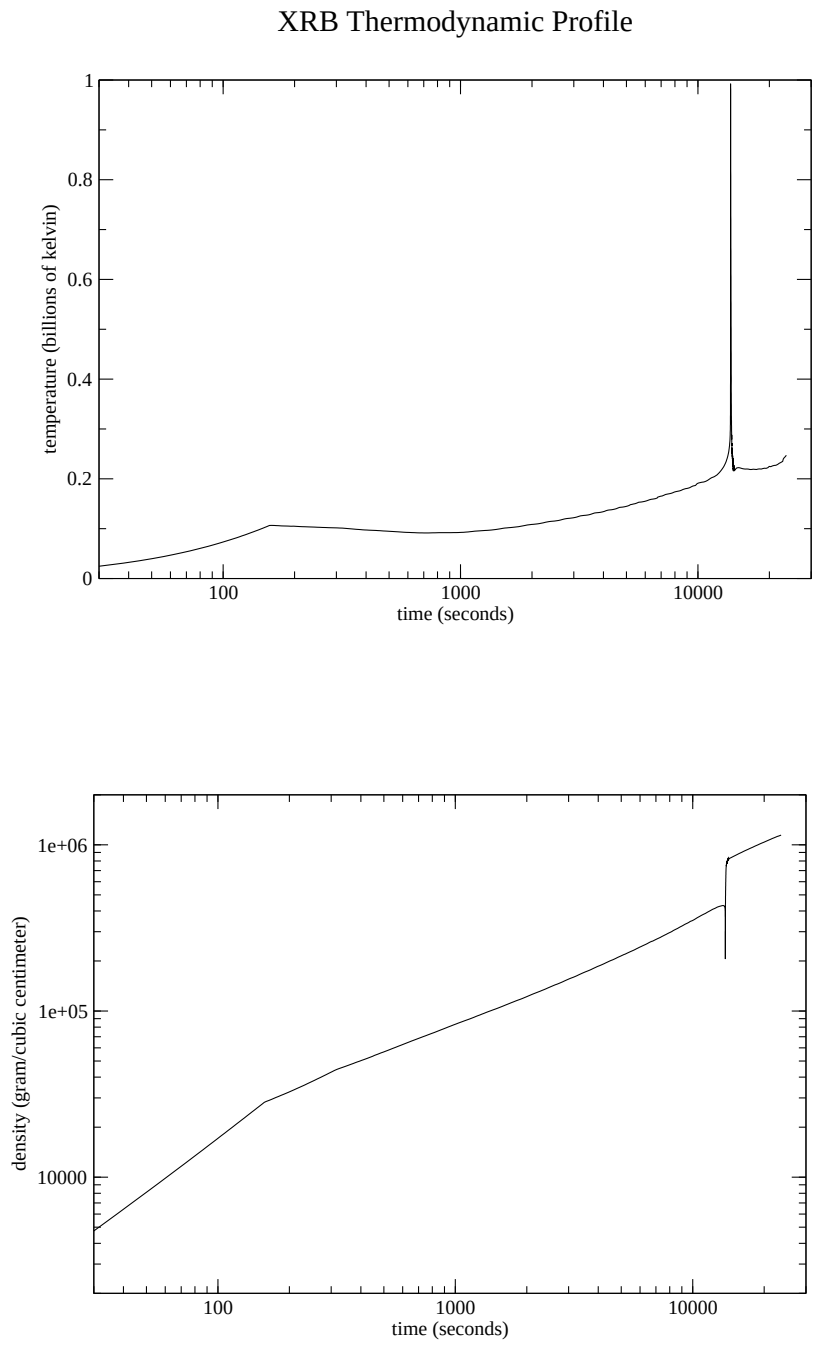


Figure 4.29: The temperature and density profiles for the XRB problem.

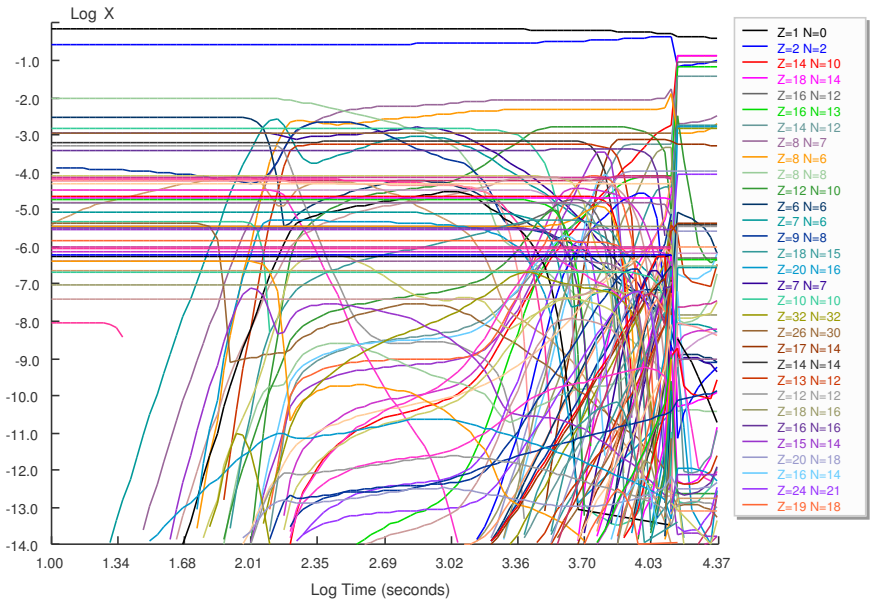


Figure 4.30: Time evolution of the mass fractions for the x-ray burst simulation.

Table 4.16: Selected final mass fractions for the x-ray burst problem

Isotope	Asymptotic	Backward Euler	Bader-Deuflhard
^{1}H	4.04×10^{-1}	3.95×10^{-1}	3.69×10^{-1}
^{4}He	9.92×10^{-2}	1.03×10^{-1}	1.02×10^{-1}

The asymptotic calculation shown in Figure 4.30 matches the implicit calculations for the larger populations to within 5%. As an example consider the mass fraction of two of the larger populations, ^{4}He and ^{1}H (shown in Table 4.16. The asymptotic calculation has a percent difference from backward Euler of 2.38% for ^{1}H and 3.69% for ^{4}He. The asymptotic calculation differs from Bader-Deuflhard by of 2.12% and 2.75% for ^{1}H and ^{4}He respectively.

Despite this level of accuracy, the asymptotic algorithm is uncompetitive with its timesteps. From $\log(t) = 2.2$ the asymptotic calculation takes smaller timesteps than the implicit methods. The majority of timesteps taken by the asymptotic lie in the region depicted in the bottom of Figure 4.31 where the asymptotic algorithm takes timesteps which are a factor of 10,000 smaller than the implicit timestep. While the backward Euler calculation took 2117 timesteps and the Bader-Deuflhard calculation

took 2039 timesteps over the entire time range, the asymptotic algorithm spent over 50,000,000 timesteps to compute the entire time domain. The asymptotic calculation is uncompetitive for the x-ray burst problem.

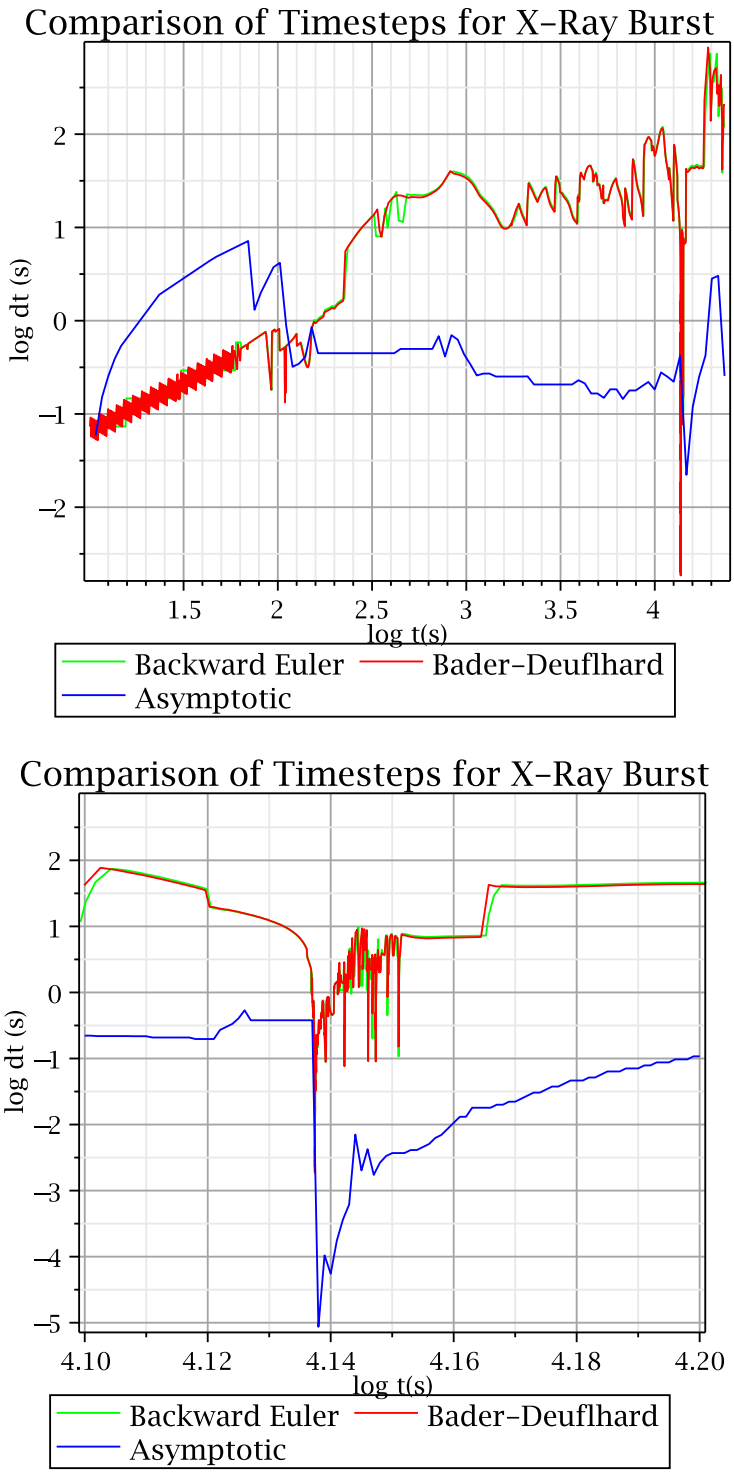


Figure 4.31: Timestep comparison for the x-ray burst simulation over the entire range of integration (top) and focused around the interaction region (bottom).

4.9 Test Problem: r-process

The r-process takes its name from the fact that it involves neutron captures on much shorter (10^{-5} seconds) timescales than the typically slower β-decays (such as the decay of free neutrons into protons which occurs in about 600 s), hence the process is called rapid neutron capture processes, or r-processes for short [53]. The r-process is a chain of (n, γ) and (γ, n) reactions, linked by β-decay; which generate many of the nuclides with masses $A \geq 80$[73]. It is hypothesized that the main site for the r-process to occur, especially for the production of isotopes with $A \approx 195$, is in the neutron-rich and neutrino-driven wind of core-collapse supernova [74], but other sites such as the merger of two neutron stars remain possibilities.

To accommodate the creation of so many heavy and neutron rich isotopes, the network has 2184 isotopes in it with 25,902 reactions from the REACLIB library coupling them together. The calculation starts at a time of 6.88990×10^{-2} seconds when the temperature is 9.98×10^9 K and the density is 3.8106×10^6 g cm^{-3}. Neuclei at this point are in nuclear statistical equilibrium. The temperature and density steadily decrease over the length of the calculation to a minimum temperature of 2.5×10^9 K and density of 5.52×10^4 g cm^{-3}. This represents the point where charged particle capture freezes out, leaving only neutrons captures and β-decays. The temperature and density profile used in this calculation can be seen in Figure 4.32.

Figure 4.33 shows the timesteps taken by Bader-Deuflhard and backward Euler in XNET and compares them to the asymptotic solver. The asymptotic solver started off taking timesteps of $dt = 10^{-11}$ seconds, but soon the size of the timestep oscillated between of order 10^{-11} seconds and 10^{-15} seconds.

The asymptotic solver was allowed to run for a month to solve this test problem. After the first twenty hours of running, the asymptotic solver had progressed 4.03×10^6 timesteps and advanced from $t = 7 \times 10^{-2}$ seconds to $t = 7.007 \times 10^{-2}$ seconds. Over the course of the remaining time, the asymptotic solver was taking such small timesteps that it had not even reached $t = 7.008 \times 10^{-2}$ seconds. Assuming the

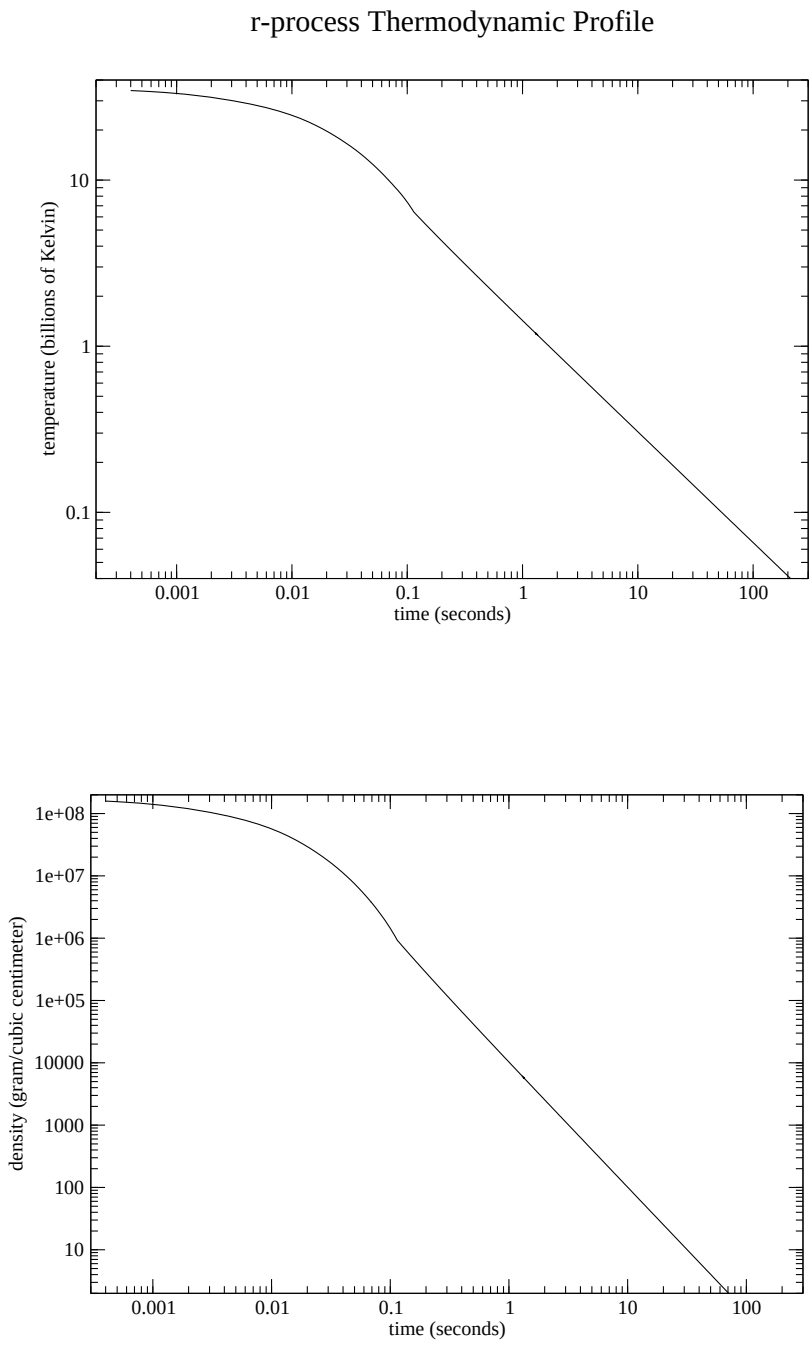


Figure 4.32: The temperature and density profiles for the r-process problem.

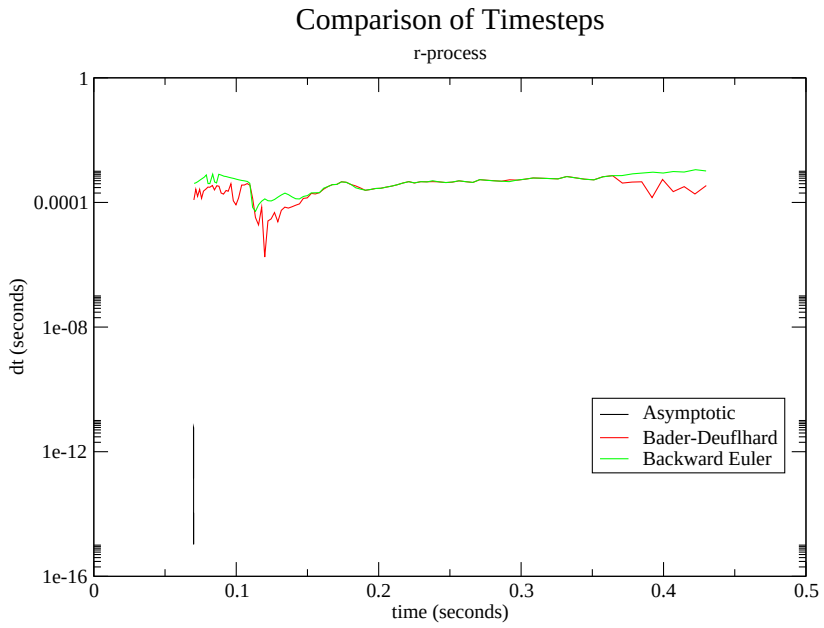


Figure 4.33: A comparison of timestep size for the asymptotic, Bader-Deuflhard, and backward Euler solvers for a simulation of the r-process.

timestep would have remained relatively constant at around $dt = 10^{-11}$ seconds, the asymptotic solver would need an estimated 14 years to complete the simulation. The asymptotic approximation is not suitable for treating this r-process simulation.

4.10 Summary

The asymptotic method appears competitive in only a handful of the different network and thermodynamic profile combinations studied. The asymptotic method does well for the classical nova simulation in Section 4.4, where the timestep taken by the asymptotic method is at worst a factor of 3 smaller than the implicit timestep, and the tidally induced supernova cases where for the alpha network the asymptotic solver was competitive for a 14-element alpha network and underperformed slightly for the larger networks in certain regions. For other test problems the asymptotic algorithm proved it was capable of handling a variety of different computational conditions (the Type Ia supernova cases and the x-ray burst simulation) even if the method wasn't computationally efficient, but in a small number of cases (type II supernova and r-process) the problem was too costly, taking estimated years of time to finish, with the asymptotic algorithm.

There are a few things which make the nova simulation unique. It extends over a very large timescale compared to the other test problems (see Table 4.1). It has the lowest minimum and peak temperature among the various profiles (see Table 4.2). It also has the lowest minimum and peak densities among the profiles (see Table 4.3). Another feature is that as the calculation continues, the reactions in it generally tend not to go into partial equilibrium (meeting the conditions outlined in Section 3.3).

Figure 4.34 shows the fraction of isotopes which are asymptotic and the fraction of pairs of reactions which satisfy the partial equilibrium conditions during the Nova problem. The asymptotic algorithm is quite successful at integrating many systems quickly when there is little to no partial equilibrium in the system. Less than 10% in equilibrium as in the nova calculation doesn't pose a hurdle at all. A system with more partial equilibrium, such as occurs under the type Ia supernova conditions, causes the asymptotic to choke and rapidly decrease its timestep to maintain stability. Figure 4.35 shows the fraction of isotopes being treated asymptotically and the fraction of

reaction pairs which are in partial equilibrium for the 150-isotope network under type Ia supernova conditions.

The asymptotic method in the nova calculation could stay competitive and only drop its timestep down to a factor of three smaller than the implicit methods, then quickly recover and start taking timesteps comparable in size again. The asymptotic timestep in the type Ia calculation drops to six orders of magnitude smaller than the implicit methods. Both timestep size decreases happen near a temperature spike, qualitatively the features in the temperature profiles are similar. The temperature in the nova case stays relatively cool, and nothing in the system is driven into the partial equilibrium conditions; indeed, some things are driven out of partial equilibrium with the partial equilibrium fraction dropping from close to 10% to a number more like 6% or 7%. In the case of the type Ia profile, the temperature rises high enough that nearly 80% of all reaction pairs in the system are driven into partial equilibrium. The asymptotic method can't cope with instability introduced by reactions going into equilibrium.

The tidally induced supernova problem, especially with a 14-element alpha network, is treatable with the asymptotic algorithm. The alpha network case with the asymptotic method took only 56% more timesteps than the backward Euler solution. The other 14-isotope alpha network test, for the type Ia supernova case took over 4,000,000 timesteps which is over 1000 times the number of timesteps taken by the implicit solvers in that case. The difference here is again the amount of partial equilibrium in the system. The tidally induced supernova generates almost no reactions in partial equilibrium (Figure 4.36) while even for an alpha network the type Ia supernova reaches more than 40% of reaction pairs in partial equilibrium (Figure 4.37).

One of the current weaknesses of the implementation of the asymptotic algorithm lies in its current lack of a sophisticated and reliable timestepping algorithm that can choose timesteps appropriate for the problem at hand and optimize the step size somewhat. The current algorithm is described in Section 3.2.2. It tries to conserve

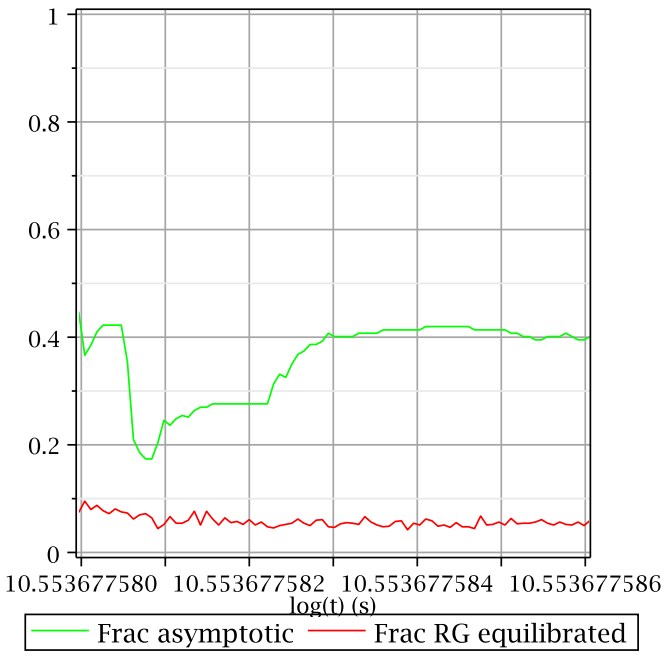


Figure 4.34: The fraction of isotopes that are being treated asymptotically and the fraction of reaction groups which meet the partial equilibrium condition described in Section 3.3 for the nova simulation.

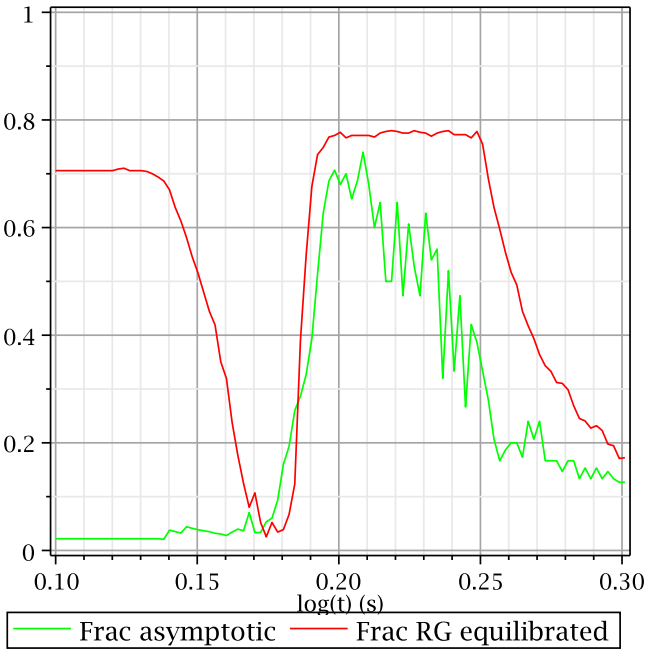


Figure 4.35: The fraction of isotopes that are being treated asymptotically and the fraction of reaction groups which meet the partial equilibrium condition described in Section 3.3 for a 150-isotope network under Type Ia Supernova conditions.

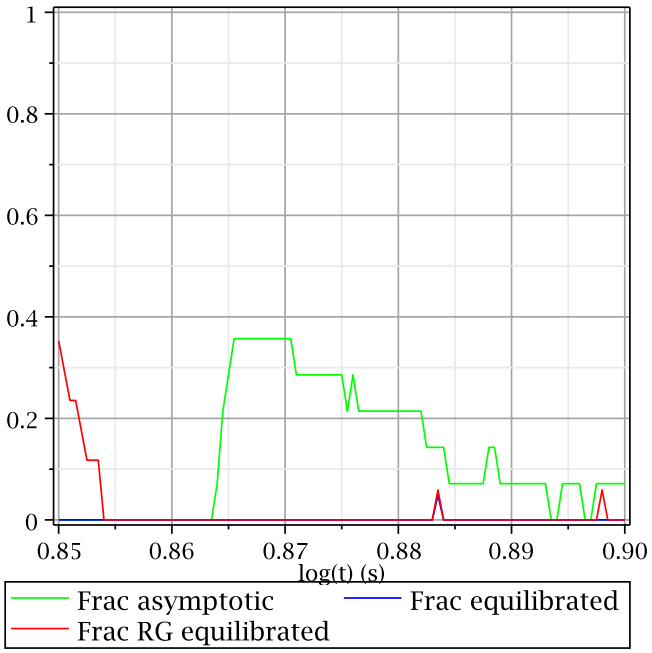


Figure 4.36: The fraction of isotopes that are being treated asymptotically and the fraction of reaction groups which meet the partial equilibrium condition described in Section 3.3 for the tidally induced supernova case.

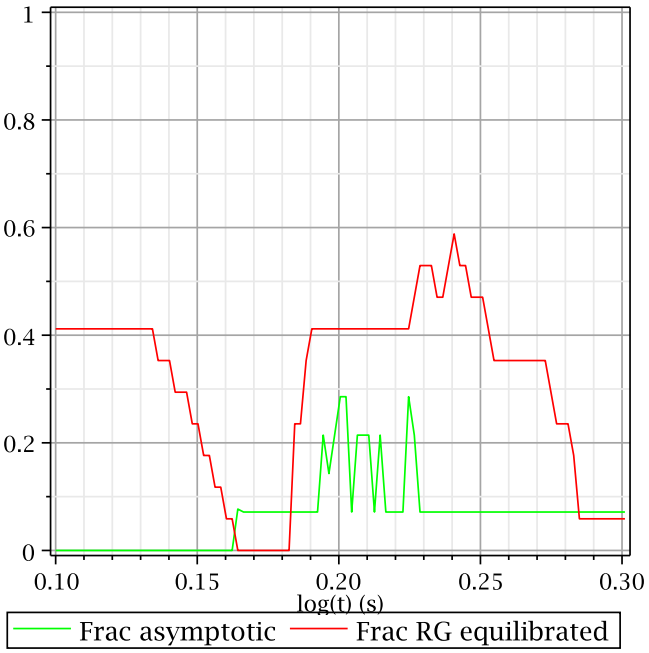


Figure 4.37: The fraction of isotopes that are being treated asymptotically and the fraction of reaction groups which meet the partial equilibrium condition described in Section 3.3 for the type Ia supernova test problem.

the particle number in the system, making the timesteps smaller if conservation is violated by a large enough amount and increasing the timestep size if conservation is within some tolerance. A better timestepping algorithm that can adjust more quickly to changing conditions in the simulation, whether by changing its conservation tolerance or some other means, could take more optimal timesteps.

The lack of a good timestepping algorithm combined with the weakness of the asymptotic algorithm to partial equilibrium conditions means that it is only suitable for problems where the amount of partial equilibrium is limited. The classical nova test problem suggests that the asymptotic approximation's ability to cope with partial equilibrium is limited to about a 10% fraction of reaction groups in partial equilibrium.

The partial equilibrium method coupled with the asymptotic approximation is only implemented for alpha networks at present, so the only example presented of its use for these test problems is for a 14-element alpha network. The type Ia supernova becomes heavily equilibrated, peaking with almost 60% of reaction pairs in partial equilibrium (see Figure 4.37). Using the partial equilibrium method developed in Section 3.3 combined with the asymptotic algorithm to treat the type Ia supernova with an alpha network results in a much faster integration than the asymptotic solver on its own. The asymptotic solver takes 4,810,018 timesteps to finish its simulation, the combined partial equilibrium and asymptotic method takes 87,465 timesteps. This is still a far cry from being directly competitive with the implicit methods (backward Euler takes 2,441 timesteps), however it is a great improvement over the asymptotic method on its own.

There does not yet exist a good timestepping algorithm for the combined partial equilibrium and asymptotic method. We speculate that the timesteps being taken by combined partial equilibrium and asymptotic algorithm (the curve labeled ASY+PE in Figure 4.38) are not as aggressive as is possible and that a better timestepping algorithm can take steps at least factors of several and possibly factors of ten larger than is currently being taken in the interaction region from $\log(t) = 0.18$ to $\log(t) = 0.26$. An increase of a factor of ten in the timestep size for ASY+PE would put it

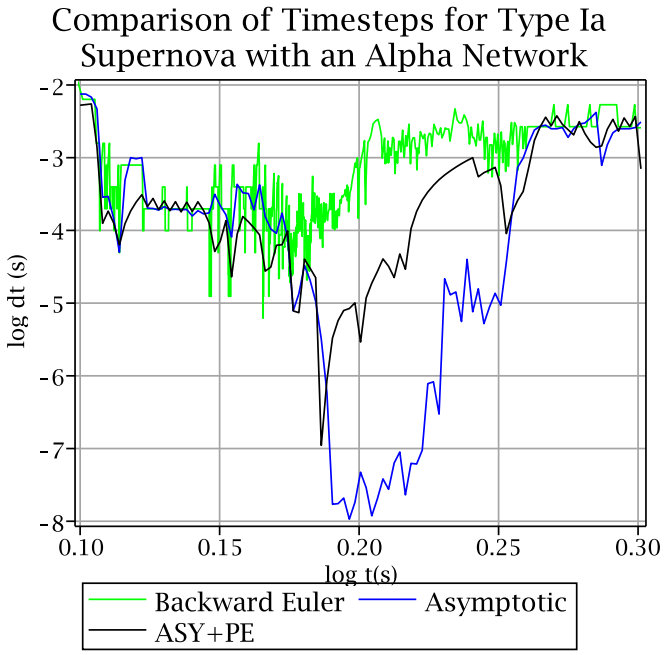


Figure 4.38: Comparison of the backward Euler, asymptotic, and combined partial equilibrium with asymptotic algorithm (ASY+PE) timesteps for the type Ia supernova test problem from Section 4.5 with an alpha network.

close to being competitive for this problem, as it is the combined partial equilibrium and asymptotic method underperforms compared to the implicit methods but in this test case it demonstrates how utilizing partial equilibrium techniques helps the solver to takes fewer, larger timesteps under equilibrated conditions than the asymptotic solver on its own is capable of.

Chapter 5

Conclusion

It is important in any doctoral dissertation to clearly state what has been done, why it is important, and what questions remain to be answered or what new directions for research are suggested. New explicit techniques, if they are faster than standard implicit techniques, could allow for larger and more realistic thermonuclear reaction networks to be coupled to hydrodynamics in large simulations of core-collapse supernova, type Ia supernova, or other astrophysical problems of interest. A better thermonuclear network would yield more correct results for the composition of ejecta and of the energy released by thermonuclear burning, both of which would enable more accurate predictions to be made and further the understanding of these complex systems.

5.1 What Has Been Accomplished

In Chapter 3 we developed and adapted the asymptotic approximation and partial equilibrium techniques into a form usable for the solution of astrophysical nuclear reaction networks and presented the results of example calculations. In Chapter 4 we investigated the feasibility of using the asymptotic approximation in post-processing schemes using thermodynamic profiles from realistic hydrodynamic simulations of various astrophysical objects. There we demonstrated that the asymptotic algorithm

was not a suitable choice to simulate the nuclear reaction networks for several thermodynamic temperature and density profiles with a variety of initial abundances. It was also demonstrated that the asymptotic approximation can be successfully used when the reactions in a network do not satisfy the partial equilibrium condition.

- The asymptotic algorithm proved unable to realistically treat the r-process or type II supernova test problems.
- For the type I x-ray burst and type Ia supernova test problems the asymptotic algorithm was unable to treat the problems competitively. As shown in Section 4.10, it is believed that this failure arises from partial equilibrium conditions and there is hope the partial equilibrium techniques will help to make explicit techniques competitive for these cases.
- In the case of the tidally induced supernova test problem, the asymptotic method could take competitive timesteps for the 14-isotope alpha network. For the 150-isotope and 299-isotope networks the asymptotic algorithm underperformed in certain regions but was either competitive or close to competitive overall.
- The asymptotic algorithm was very competetitive with the implicit solvers for the nova test, taking competitive timesteps in every thermodynamic region examined.

5.2 Work to Be Done

Currently, effort is being concentrated on three fronts. The first is integrating the asymptotic approach into calculations which couple the network to a hydrodynamics code. The second is the application of partial equilibrium to larger networks, which has yet to be demonstrated. The third front is improvements upon the asymptotic algorithm itself.

Viktor Chupryna in his doctoral dissertation [75] and Chris Smith in his master's thesis [76] have shown that the asymptotic approximation behaves well when coupled to hydrodynamic calculations done in the astrophysical code called FLASH [77]. FLASH uses an adaptive mesh grid to simulate fluid flow and functions in an operator split formalism. In such an environment the maximum timestep taken by the nuclear reaction network is limited by the Courant condition, which means that implicit methods may not be able to fully exploit their inherent stability and take as large a timestep as possible, making it easier for the asymptotic method to be competitive.

Much effort (see Chris Smith's thesis for the beginning of this effort [76]) has been expended to reproduce and improve upon the calculation of Maier and Neimeyer [78] for the interaction between a thermonuclear detonation and bubbles of previously burned matter for type Ia supernovae. The "Maier" problem sends a nuclear detonation down a rectangular region filled with fuel which is obstructed by an obstacle of previously burned material (or ash). Figure 5.1 shows a snapshot of the temperature from an ongoing calculation of a thermonuclear flame traveling through a region of fuel. At this level of resolution (6 levels of refinement in FLASH, about 160,000 computational zones) the 70-isotope asymptotic network has developed a Landau-Darrieus instability [79] [80]. The same calculation performed with an alpha network at 6 levels of refinement in FLASH doesn't develop the detailed structure of the Landau-Darrieus instability; the alpha network calculation requires 10 levels of refinement (about 2,500,000 computational zones) to resolve the flame front and develop a Landau-Darrieus instability.

Being able to resolve detailed hydrodynamic effects from the inclusion of a larger nuclear reaction network as opposed to increasing the resolution of the hydrodynamic grid could lead to computational time savings if the network can be computed quickly enough. As reported in [76], the asymptotic algorithm code has not been optimized and there remains much work to do in that area. There are also lingering problems with the equation of state routines in FLASH when coupled to a large nuclear reaction network.

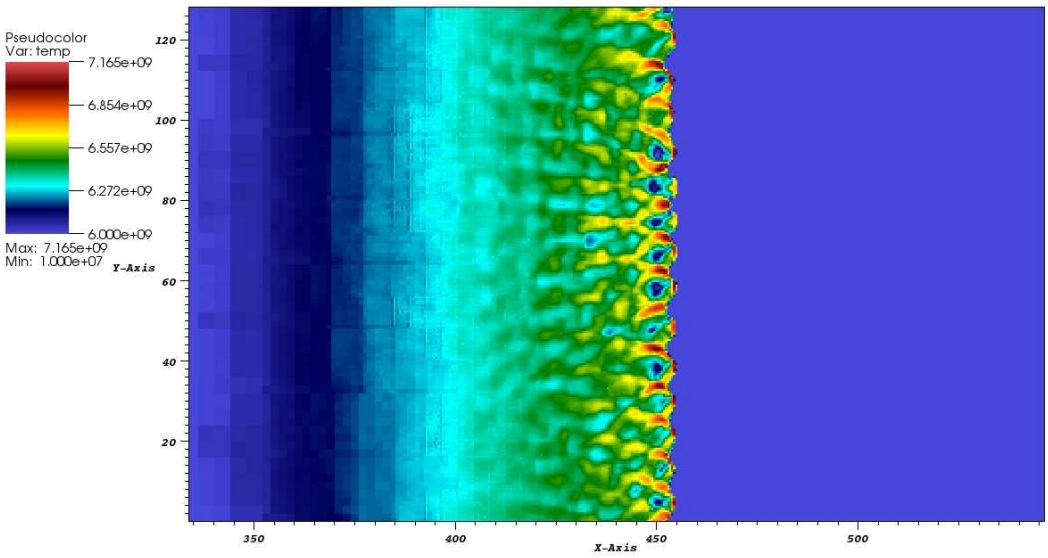


Figure 5.1: Plot of temperature for a thermonuclear detonation traveling through a region of fuel.

In Section 3.3.7 it was shown that the partial equilibrium technique was able to take timesteps comparable to the semi-implicit YASS calculation performed in Mott's thesis [19]. If partial equilibrium is able to take such large timesteps when applied to larger networks, then because its cost per timestep scales linearly it would be able to outperform implicit solvers. However, an extension of partial equilibrium to larger and more realistic nuclear networks has yet to be demonstrated. There are other explicit techniques that have been developed, though not discussed here, such as quasi-steady state methods [19][81] used in CHEMEQ2 [82], which need to be properly explored as well.

There are several different directions that the asymptotic approximation can be improved upon. A student named Sophia Minxin He completed an extension of the range of validity for the asymptotic approximation. This has yet to be fully tested. Dr. Michael Guidry is working on a way to compensate for the loss of mass conservation inherent in the asymptotic approximation by renormalizing the isotopes

being treated asymptotically. Finally, the most important piece is the design and implementation of a smarter adaptive timestep that can exploit the full power of the asymptotic approximation.

Bibliography

Bibliography

[1] W.H. Press, et al., Numerical Recipes in C, 2nd Edition, 1992, Cambridge University Press 2, 3, 15

[2] E. Hairer and G. Wanner, Solving Ordinary Differential Equations II, 1991, Springer-Verlag 2

[3] C.F. Curtiss and J.O. Hirschfelder, PNAS, **38**: (3) 235-243, 1952 xii, 2, 3, 5

[4] W.H. Enright, SIAM Journal on Numerical Analysis, **11**: (2) 321-331, 1974 2

[5] A. Prothero and A. Robinson, Mathematics of Computation, **28**: (125) 145-162, 1974 2

[6] D. Barton, ACM Transactions on Mathematical Software, **6**: (3) 280-294, 1980 2

[7] W.E. Boyce and R.C. DiPrima, Elementary Differential Equations and Boundary Value Problems, 7th Edition, 2001, Wiley & Sons 3, 4

[8] R.K. Nagle, E.B. Saff and A.D. Snider, Fundamentals of Differential Equations, 5th Edition, 2000, Addison-Wesley 3

[9] L.F. Shampine and C.W. Gear, SIAM Review, **21**: (1) 1-17, 1979 4

[10] V.K. Barwell, BIT, **15**: (2) 130-135, 1975 12

[11] G. Bader and P. Deuflhard, Numerische Mathematik, **41**: (3) 373-398, 1983 13, 15

[12] F.X. Timmes, The Astrophysical Journal Supplemental Series, **124**: 241-263, 1999 13, 15

[13] P. Deuflhard, Numerische Mathematik, **41**: (3) 399-422, 1983 15

[14] T. Rauscher and F. Thielemann, Atomic Data and Nuclear Data Tables, **79**: (1) 47-64, 2001 19, 94

[15] M.W. Guidry, R. Budiardja, E. Feger, W.R. Hix, O.E.B. Messer, J.J. Billings, K.J. Roche and Erin McMahon, Explicit Integration of Very Stiff Reaction Networks: (I) Non-Equilibrium Conditions, 2010, unpublished manuscript, Department of Physics and Astronomy, University of Tennessee-Knoxville xiii, 21

[16] Mike Guidry, Stellar Structure and Early Stellar Evolution, 2008, unpublished manuscript, Department of Physics and Astronomy, University of Tennessee-Knoxville xvii, 22, 101, 102, 111

[17] T.R. Young and J.P. Boris, Journal of Physical Chemistry, **81**: (25) 2424-2427, 1977 26

[18] Mike Guidry, Explicit Methods for Solutions of Large Thermonuclear Networks Coupled to Multidimensional Hydrodynamics Simulations, 2010, unpublished manuscript, Department of Physics and Astronomy, University of Tennessee-Knoxville xiii, xiv, xv, xvi, xvii, 29, 30, 32, 33, 34, 42, 53, 54, 75, 76, 77, 80, 81, 82, 85, 88

[19] David Ray Mott, 'New Quasi-Steady-State and Partial-Equilibrium Methods for Integrating Chemically Reacting Systems' (PhD Dissertation, University of Michigan 1999) xvi, 41, 50, 74, 85, 158

[20] David Bodansky, Donald D. Clayton and William A. Fowler, The Astrophysical Journal Supplement Series, **16**: 299-371, 1968 41, 60

[21] John D. Ramshaw and Lawrence D. Cloutman, Journal of Computational Physics, **39**: 405-417, 1981 41

[22] CRC Handbook of Chemistry and Physics, 88th Edition, ed. David R. Lide, 2008, CRC Press 49

[23] Howard Anton, Elementary Linear Algebra, 8th Edition, 2000, Wiley & Sons 56

[24] A.M. Khokhlov, The Astrophysical Journal, **419**: 200-206, 1993 79

[25] W.R. Hix and B.S. Meyer, Nuclear Physics A, **777**: 188-207, 2006 93

[26] G.R. Caughlan and W.A. Fowler, Atomic Data and Nuclear Data Tables, **40**: (2) 283-334, 1988 94

[27] M. Smith, Nuclear Data for Nuclear Astrophysics, http://www.phy.ornl.gov/astrophysics/data/data.html 94

[28] R. Firestone and S. Wu, Nuclear Astrophysics Data, http://ie.lbl.gov/astro.html 94

[29] W. Raphael Hix and Friedrich-Karl Thielemann, The Astrophysical Journal, **511**: 862-875, 1999 94

[30] Ed Anderson, LAPACK Users' Guide, 3rd Edition, 1999, SIAM 94

[31] Iain S. Duff, A.M. Erisman and J.K. Reid, Direct Methods for Sparse Matrices, 1986, Oxford University Press 94

[32] Parallel Sparse Direct Linear Solver (PARDISO) User Guide, Version 3.2, Computer Science Department, (undated) University of Basel, Switzerland 94

[33] O. Schenk, K. Gartner and W. Fichtner, BIT, **40**: (1) 158-176, 2000 94

[34] O. Schenk and K. Gartner, Journal of Future Generation Computer Systems, **20**: (3) 475-487, 2004 94

[35] O. Schenk and K. Gartner, Electronic Transactions on Numerical Analysis, **23**: 158-179, 2006 94

[36] Scientific Computing Software Library (SCSL) User's Guide, 2003, Silicon Graphics, Inc., California 100

[37] A.E. Champagne and M. Wiescher, Annual Review of Nuclear and Particle Science, **42**:39-76, 1992 101

[38] B. Lazareff, et al., The Astrophysical Journal, **228**: 875-880, 1979 102

[39] Tod Strohmayer and Lars Bildsten, 2006, Compact Stellar X-ray Sources, eds. Walter H.G. Lewin and Michiel van der Klis (Cambridge University Press) 113 102

[40] R.K. Wallace and S.E. Woosley, The Astrophysical Journal Supplement Series, **45**: 389-420, 1981 102

[41] S. Starrfield, et al., The Astrophysical Journal, **692**: 1532-1542, 2009 102, 103

[42] Massimo Turatto, Lecture Notes in Physics, **598**: 21-36 2003 111

[43] D. Branch, et al., Publications of the Astronomical Society of the Pacific, **107**: 1019-1029, 1995 111

[44] T. Plewa, A.C. Calder and D.Q. Lamb, The Astrophysical Journal Letters, **612**: (1) L37-L40, 2004 111

[45] Rüdiger Pakmor, et. al., Nature, **463**: 61-64, 2010 111

[46] D. Branch and G. A. Tammann, Annual Review of Astronomy and Astrophysics, **30**: 359-389, 1992 111

[47] W. Hildebrandt and J.C. Niemeyer, Annual Review of Astronomy and Astrophysics, **38**: 191-230, 2000 111

[48] E.F. Brown, et al., Nuclear Physics A, **758**: 451c-454c, 2005 111

[49] R.H. Cyburt, et. al., The Astrophysical Journal Supplement Series, **189**: 240-252, 2010 113

[50] Stephan Rosswog, Enrico Ramirez-Ruiz and William R. Hix, The Astrophysical Journal, **679**: 1385-1389, 2008 120, 122

[51] Stephan Rosswog, et al., Computer Physics Communications, **179**: 184-189, 2008 120

[52] Stephan Rosswog, Enrico Ramirez-Ruiz and W. Raphael Hix, Journal of Physics: Conference Series, **172**: (012036) 1-8, 2009 122

[53] Samuel S.M. Wong, Introductory Nuclear Physics, 2nd Edition, 2004, Wiley-VCH 134, 144

[54] E.H. Lieb and H.T. Yau, The Astrophysical Journal, **323**: (1) 140-144, 1987 134

[55] H.A. Bethe, Physics Today, **43**: (9) 24-27, 1990 134

[56] K. Takahashi, K. Sato, A. Burrows, and T.A. Thompson, Physical Review D, **68**: (113009) 1-8, 2003 134

[57] K. Takahasi, K. Sato, H.E. Dalhed and J.R. Wilson, Astroparticle Physics, **20**: 189-193, 2003 134

[58] H.A. Bethe, Reviews of Modern Physics, **62**: (4) 801-866, 1990 134

[59] C. Frohlich, et al., Physical Review Letters, **96**: (142502) 1-4, 2006 134

[60] Max Camenzind, Compact Objects in Astrophysics, 2007, Springer 138

[61] J. Grindlay and J. Heise, 1975, International Astronomical Union Circular, No. 2879 138

[62] R.D. Belian, J.P. Conner and W.D. Evans, Bulletin of the American Astronomical Society, **8**: 396, 1976 138

[63] Walter H.G. Lewin and Paul C. Joss, 1983, Accretion-driven stellar X-ray sources, eds. Walter H.G. Lewin and Edward P.J. van den Heuvel (Cambridge University Press) 41 138, 139

[64] S.E. Woosley and Ronald E. Taam, Nature, **263**: 101-103, 1976 138

[65] A. Cumming, Nuclear Physics B - Proceedings Supplements, **132**: 435-445, 2004 138

[66] E. Kuulkers, Nuclear Physics B - Proceedings Supplements, **132**: 466-475, 2004 138

[67] C.J. Hansen and H.M. Van Horn, The Astrophysical Journal, **195**: 735-741, 1975 138

[68] Jacob Lund Fisker, Hendrik Schatz and Friedrich-Karl Thielemann, The Astrophysical Journal Supplement Series, **174**: 261-276, 2008 138

[69] F.-K. Thielemann, et al., 1998, Stellar Evolution, Stellar Explosions and Galactic Chemical Evolution, ed. Anthony Mezzacappa (Institute of Physics Publishing) 483 138

[70] J.-F. Rembges, et al., 1998, Stellar Evolution, Stellar Explosions and Galactic Chemical Evolution, ed. Anthony Mezzacappa (Institute of Physics Publishing) 495 138

[71] Tod Strohmayer, 2001, X-ray Astronomy, eds. Nicholas E. White, Gluseppe Malaguti and Glorgio G.C. Palumbo (American Institute of Physics) 377 139

[72] Dimitrios Psaltis, 2006, Compact Stellar X-ray Sources, eds. Walter H.G. Lewin and Michiel van der Klis (Cambridge University Press) 1 139

[73] S.E. Woosley and R.D. Hoffman, The Astrophysical Journal, **395**: (1) 202-239, 1992 144

[74] Beun, et al., Physical Review C, **77**: (035804) 1-10, 2008 144

[75] Viktor Chupryna, 'Explicit Methods in a Nuclear Burning Problem for Supernova Ia Models' (PhD Dissertation, University of Tennessee-Knoxville 2008) 157

[76] Christopher Ryan Smith, 'Application of the Explicit Asymptotic Method to Nuclear Burning in Type Ia Supernova' (Masters Thesis, University of Tennessee-Knoxville 2009) 157

[77] G.C. Jordon, et al., The Astrophysical Journal, **681**: 1448-1457, 2008 157

[78] A. Maier and J.C. Niemeyer, Astronomy & Astrophysics, **451**: 207-212, 2006 157

[79] J.B. Bell, et al., The Astrophysical Journal, **606**: 1029-1038, 2004 157

[80] Sergi Iv. Blinnikov and Pavel V. Sasorov, Physical Review E, **53**: (5) 4827-4841, 1996 157

[81] David R. Mott, Elaine S. Oran, and Bram van Leer, Journal of Computational Physics, **164**: 407-428, 2000 158

[82] David R. Mott and Elaine S. Oran, CHEMEQ2: A Solver for the Stiff Ordinary Differential Equations of Chemical Kinetics, NRL/ML/6400–01-8553, 2001, Naval Research Laboratory 158

Vita

Elisha Feger was born in Louisville, Kentucky on July 21, 1981. He attended Berea College in Berea, Kentucky where he received a Bachelor of Arts in Mathematics and Physics in 2003. In the Fall of 2003 he joined the Physics and Astronomy Department of the University of Tennessee, Knoxville as a graduate student. He worked both as a Graduate Teaching Assistant and a Graduate Research Assistant. As a Graduate Teaching Assistant he taught undergraduate Physics labs and recitation. His research was in the field of Computational Astrophysics, specifically the computation of nuclear reaction networks. He completed his Doctoral Degree in August 2011.