

An Explicit Asymptotic Approach Applied To Neutrino-Electron Scattering In The Neutrino Transport Problem

A Thesis Presented for the
Masters of Science
Degree

The University of Tennessee, Knoxville

Aaron Michael Lackey-Stewart
December 2020

Copyright © by Aaron Michael Lackey-Stewart, 2020
All Rights Reserved.

dedication...

For all of my friends and family. New and old. Here and passed on. Most of all to my mother. Thank you all.

Acknowledgments

I would like to thank The University of Tennessee's physics department for allowing me the opportunity to learn at this program as well as the professors who I learned from. I would also like to thank Dr. Mike Guidry and Dr. Eirik Endeve who afforded me the opportunity to work on their project and guided me through this research and the completion of this thesis.

Abstract

This thesis presents results of explicit asymptotic calculations applied to neutrino-electron collisions in the neutrino transport problem; a problem that is generally solved using implicit methods when simulating core collapsed supernovae. It is shown that the explicit asymptotic method provides stable solutions to these stiff systems of equations while also yielding comparative accuracy and time stepping to standard implicit treatments such as Backward Euler, Fixed Point Iteration, and Anderson Accelerated Fixed Point. Because implicit methods are found to be less efficient for large systems of stiff, coupled equations, these results could help cut costs in solving this problem while also serving as a baseline for what the method can be used for in other scientific contexts; much the same as with the thermonuclear network calculations detailed by Guidry [5]. The particular algorithm detailed in this thesis is only applied to a simplified problem using the spatially homogenous transport equation where we consider relaxation problems. When using our time stepping algorithm that limits the error with each iteration, explicit asymptotic generates results that are competitive even for large time steps given a sufficient choice of parameters.

Table of Contents

1	Introduction	1
2	Mathematical Formulation	6
2.1	Mathematical Model	6
2.2	Discretization of the Problem	9
2.3	Matrix Formulation and Stiffness	11
3	Numerical Methods	16
3.1	Explicit Methods	17
3.1.1	Forward Euler Method	17
3.1.2	Explicit Asymptotic Approximation	19
3.2	Implicit Methods	33
3.2.1	Backward Euler Method	34
3.2.2	Fixed Point Iteration	38
3.2.3	Anderson Accelerated Fixed Point Iteration	41
4	Numerical Comparisons	45
4.1	Error And Time Stepping Comparisons	45
4.2	Speed Comparisons	54
5	Conclusions	61
	References	63
	Appendix	66

List of Tables

2.1	Table of models used in thesis. Further information can be found in [7] . . .	13
3.1	Cycle and Truecycle for $dt_{grow} = 1.003$	29
3.2	Cycle and Truecycle for $dt_{grow} = 1.03$	30
3.3	Cycle and Truecycle for $dt_{grow} = 1.20$	31

List of Figures

1.1	Example of an explicit calculation via Forward Euler.	3
1.2	Example of an implicit calculation via Backward Euler	3
2.1	Plot of average time taken to do matrix-vector multiplications versus matrix inversions for $n \times n$ matrices. Here it is evident that matrix inversions take considerably longer.	13
2.2	Plot of mean collision times for all energy bins in the low occupancy limit. The collision times for all densities differ by orders of magnitude across all energies, indicating stiffness that's induced by the scattering kernel.	15
2.3	Plot of mean free path for all energy bins in the low occupancy limit. For the higher energy bins, collisions are substantially more common than for the lower energy bins.	15
3.1	Time evolution of neutrino distribution function for all energies.	19
3.2	Plot showing the time evolution of number densities for five arbitrary energy bins. EA is solid and FE is dashed and they appear to overlap.	24
3.3	Relative difference between EA and FE number densities for the same energy groups in Fig.(3.2). The error never exceeds 10^{-2} for any energy bin. As such the EA method solves the equations while being reasonable accurate with FE even for a tight tolerance on particle number error.	24
3.4	Plot showing the time evolution of number densities for five arbitrary energy bins. EA is solid and FE is dashed and they appear to overlap.	25

3.5	Relative difference between EA and FE number densities for the same energy groups in Fig.1. The error maxes out at $\sim 5\%$. As such the EA method solves the equations while being reasonable accurate with FE even for a tight tolerance on particle number error.	26
3.6	Time stepping and particle conservation sample plots. The goal is to achieve competitive time stepping with implicit methods while also minimizing the change in particle number.	28
3.7	Time stepping evolution for EA in high density case. Small dt_{grow} . Evolution for all tolerances is linear due the algorithm never having to adjust the time step.	29
3.8	Time stepping evolution for EA in high density case. Intermediate dt_{grow} . We see more pronounced diverging from linearity due to time steps being too large and violating the tolerance conditions and thus needing adjustment. . .	30
3.9	Time stepping evolution for EA in high density case. Large dt_{grow} . This divergence from a linear evolution exacerbates with more aggressive time stepping.	31
3.10	EA particle conservation for small time stepping. Small dt_{grow} yields better particle conservation due to the tolerance condition never being violated. . .	32
3.11	EA particle conservation for intermediate time stepping. Here the change doesn't exceed 1% for a given tolerance however tolerances greater than 10^{-3} don't yield ideal time stepping as shown in Fig.4(b).	32
3.12	EA particle conservation for large time stepping. Aggressive time stepping can lead to the algorithm losing efficiency.	33
3.13	Relative Error for BE using intermediate time stepping.	36
3.14	Relative Error for BE using large time stepping	36
3.15	Change in particle number for intermediate stepping; BE.	37
3.16	Change in particle number for large time stepping; BE.	37
3.17	Relative Error for FP using intermediate time stepping.	39
3.18	Relative Error for FP using aggressive time stepping.	40
3.19	Change in particle number for intermediate time stepping; FP.	40

3.20	Change in particle number for large time stepping; FP	41
3.21	Number of iterations AFP takes for a given m	42
3.22	Relative Error for AFP using intermediate time stepping.	43
3.23	Relative Error for AFP using aggressive time stepping.	43
3.24	Change in particle number for intermediate time stepping; AFP	44
3.25	Change in particle number for large time stepping; AFP.	44
4.1	Max error from FE versus time compared to the error for our standard EA calculation for small dt_{grow}	46
4.2	Max error from FE for a larger dt_{grow} . Here the EA, FP, and AFP are more competitive with BE.	46
4.3	Max error from FE for an aggressive dt_{grow}	46
4.4	Time stepping and particle conservation for small, intermediate, and aggres- sive time stepping in the high density case. The time stepping for all methods are virtually identical and the methods all conserve particle number to an acceptable level.	48
4.5	Max error from FE for a small dt_{grow} in the intermediate density case.	49
4.6	Max error from FE for a larger dt_{grow} in the intermediate density case.	49
4.7	Max error from FE for an aggressive dt_{grow} in the intermediate density case.	49
4.8	Time stepping and particle conservation for small, intermediate, and aggres- sive time stepping in the intermediate density case. The time stepping for all methods are virtually identical and the methods all conserve particle number to an acceptable level.	50
4.9	Max error from FE versus for the smallest dt_{grow} in the low density case.	52
4.10	Max error from FE for a larger dt_{grow} in the low density case.	52
4.11	Max error from FE for the most aggressive dt_{grow} in the low density case.	52
4.12	Time stepping and particle conservation for small, intermediate, and aggres- sive time stepping in the high density case. The time stepping for all methods are virtually identical and the methods all conserve particle number to an acceptable level.	53

4.13	Number of right hand evaluations for the least aggressive time stepping in the high density scenario.	56
4.14	Number of right hand evaluations for a growth factor of 1.03 in the high density case.	56
4.15	Number of right hand evaluations for the most aggressive time stepping. Here the EA method is not competitive with the implicit methods.	56
4.16	Number of right hand evaluations for the least aggressive time stepping in the intermediate density case.	57
4.17	Number of right hand evaluations for a larger time step in the intermediate density case.	57
4.18	Number of right hand evaluations for the most aggressive time stepping in the intermediate density case.	57
4.19	Number of right hand evaluations for the least aggressive time stepping for the low density scenario.	58
4.20	Number of right hand evaluations for larger time stepping in the low density scenario.	58
4.21	Number of right hand evaluations for the most aggressive time stepping in the low density scenario.	58
4.22	Time stepping in high density case using stricter tolerances for implicit methods.	60
4.23	Number of right hand evaluations in the high density case using strict tolerances for implicit methods.	60
4.24	Particle conservation in the high density case using stricter tolerances for implicit methods.	60

Chapter 1

Introduction

Constructing computer simulations of core collapse supernova explosions requires solving systems of partial differential equations that govern different processes that occur at different timescales. These timescales include those related to particle collisions, hydrodynamics, and thermonuclear processes. Because rate parameters related to these processes can differ by orders of magnitude, "stiff" systems as they are called, we must be careful in choosing the methods used to solve these systems [9, 4, 6, 13]. Conventionally, implicit methods are used to solve stiff systems of equations due to there being no need to worry about stability constraints. Thus we can take larger time steps and solve the problem much faster. However, given that calculating the $n + 1$ iteration using an implicit method requires us to know information about the system at that iteration (which we don't), there are efficiency concerns for large systems of differential equations because of the extra computations that must be done per time step to solve for the unknown variable at the $n + 1$ step. For implicit treatments, these extra computations are algebraic methods like Newton-Raphson. Standard explicit methods are generally easier to implement because we only need to know the current state of the system to compute a future state and so on and so forth. However because there are stability constraints with explicit methods, this limits how large a time step you can take in your iterative process which is impractical for solving the stiff systems required to simulate supernova explosions. To illustrate the differences in explicit and implicit

calculations, consider as an example a simple exponential decay problem of the form:

$$\frac{dy}{dt} - ky = 0 \quad (1.1)$$

Where $k > 0$ is a decay rate and we can consider the initial condition $y(0) = 2$. It's straight forward to show that this equation has a solution:

$$y(t) = 2e^{-kt} \quad (1.2)$$

Here it is clear that the inverse of the decay rate $1/k$ represents a timescale over which Eq.(1.2) asymptotically approaches zero. If we were to apply the Forward Euler method to Eq.(1.1) it would take the form:

$$y_{n+1} = y_n + \Delta t y'_n = y_n(1 - k\Delta t) \quad (1.3)$$

Where Δt is a time step, y' is the derivative of y with respect to time t , and the indices in n represent iterative steps. As can clearly be seen, calculating the quantity y at some future time represented by the $n + 1$ step only requires knowing information at the present n step. Hence this is an explicit method. As long as $\Delta t \leq 2/k$, $|y_{n+1}| \leq |y_n|$ and this method produces stable solutions. We can contrast with Backward Euler, an implicit method, which takes the form:

$$y_{n+1} = y_n + \Delta t y'_{n+1} \quad (1.4)$$

Where we see there are $n + 1$ terms on both sides. In order to solve the equation using Backward Euler, we must algebraically solve for y_{n+1} to which we get:

$$y_{n+1} = \frac{y_n}{1 + k\Delta t} \quad (1.5)$$

Which we get from substituting Eq.(1.1) for y' . As can be seen, Backward Euler is unconditionally stable as $y_{n+1} < y_n$ for nonzero k . In general, using implicit methods like Backward Euler isn't this trivial and may require something like Newton-Raphson method, but we can see the differences between the two methods looking at Fig.(1.1) and Fig(1.2).

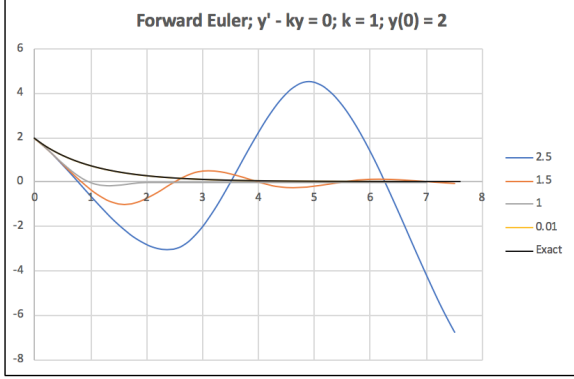


Figure 1.1: Example of an explicit calculation via Forward Euler.

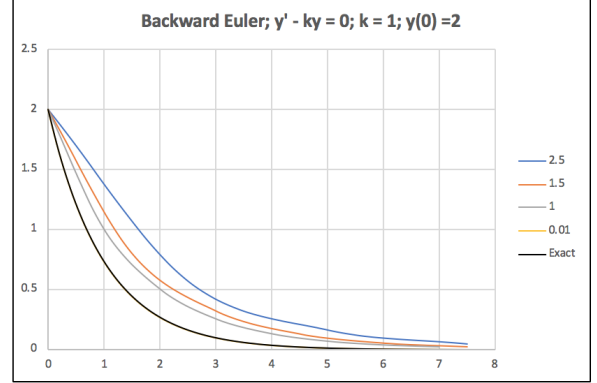


Figure 1.2: Example of an implicit calculation via Backward Euler

Here $k = 1$, and the time step cases are 0.01, 1, 1.5, and 2.5 seconds. We also compare these cases to the exact solution given by Eq.(1.2). We see with each method that the small time stepping yields the most accurate results. When the time stepping is made larger relative to k , in Forward Euler's case we get solutions that are oscillatory. In the 1.5 time stepping case the solution oscillates with a decaying amplitude that asymptotically approaches zero as time increases. In the 2.5 time stepping case the amplitude increases with increasing time, demonstrating that this solution is not stable. In contrast, we get stable solutions for Backward Euler for all time stepping cases (though the larger time stepping cases are not accurate solutions). Thus we see the issues with solving even more complicated systems with standard explicit methods. It is not practical when there are constraints that limit the size of the time step. Which leads in to the actual problem that will be discussed.

A specific problem that this thesis will address is in solving the neutrino transport problem in core collapse supernovae. To contextualize further, the energy from neutrinos released during the core collapse of a star are thought to assist the resulting shock wave in permeating throughout the star. Thus being able to model and solve this phenomena is important. However, we will not focus on all neutrino-matter interactions in supernova explosions; such as emission and absorption on nucleons and nuclei, scattering on nucleons and nuclei, and electron-positron pair creation and annihilation. Namely, we focus on neutrino-electron scattering (NES). Though this study only pertains to neutrino-electron interactions, this

problem is broad enough in scope that the methods used to solve it can serve as a guide to solving the neutrino transport problem more generally. Neutrino transport is particularly difficult to simulate because neutrinos weakly interact with the surrounding star material, and neutrino transport in the region where the neutrino mean free paths are approximately the same size as the proto-neutron star (PNS) cannot be described by models which only depend on parameters in position and time. Instead, you must utilize kinetic models which use the Boltzmann equation for the distribution function and will depend on position and momentum coordinates in phase space, as well as time and neutrino energy in momentum space. These interactions in phase space must be accounted for in every spatial coordinate, and interactions being more diffuse in some region of the star than others is why this problem is so costly, and why implicit methods are preferred. For a more comprehensive review of the neutrino transport problem, and the mechanisms behind core collapse supernovae more generally, see Mezzacappa [8]. In this thesis we will be exploring the uses of an explicit method, named the Explicit Asymptotic Approximation (EA), detailed in Guidry [5] which can potentially serve as an alternative method to standard explicit methods while potentially solving the problem faster than standard implicit methods.

As of the writing of this thesis, this method has not before been applied to this specific problem, but has been applied by Guidry in [5] in the context of thermonuclear processes. This analysis will be a test of this method in which we solve a system of equations related to NES and compare and contrast the results with standard implicit methods. Namely, Backward Euler (BE), Fixed Point Iteration, (FP), and an Accelerated Fixed Point (AFP) method as described in Toth and Kelley [12] with the idea that EA (for some choice of parameters) will be able to solve the equations with competitive speed or even faster before the next generation of super computers. We also introduce a matrix formalism to handle this system of equations and show that EA will cost less per time step than implicit methods (all other comparisons fair) because of the cost associated with doing simple matrix-vector multiplications versus matrix inversions in the case of implicit treatments. As implied by the name, EA is an approximation. Even though this is the case, for practical concerns it is only necessary that the errors associated with using this method are smaller than the

errors associated with solving other processes in supernovae (such as the hydrodynamics or thermonuclear interactions).

Chapter 2

Mathematical Formulation

In this chapter we will be discussing the mathematical description of the neutrino transport problem leading to the equations that we must solve. We then introduce a formulation to handle a system of equations. Also featured is a discussion of the stiffness in the equations as well as its origin.

2.1 Mathematical Model

First and foremost, we must construct a mathematical model of the neutrino transport problem. We view this problem as neutrinos scattering in and out of finite energy bins each having a volume V in phase space. As described earlier, how the distribution function of neutrinos evolves in time is described by the Boltzmann equation. For this particular model we use a Boltzmann equation that is spatially homogenous with inelastic neutrino-electron scattering with a fixed matter background (we assume the energy of the matter background remains constant during scattering). This expression is given by Bruenn [1]:

$$\begin{aligned} \frac{d\mathcal{F}}{dt} = (1 - \theta\mathcal{F}) \int_{V_P} \mathcal{R}^{in}(\epsilon, \epsilon', \mathbf{n} \cdot \mathbf{n}'; \mathbf{u}) \mathcal{F}' dV_{P'} \\ - \mathcal{F} \int_{V_P} \mathcal{R}^{out}(\epsilon, \epsilon', \mathbf{n} \cdot \mathbf{n}'; \mathbf{u}) (1 - \theta\mathcal{F}') dV_{P'} \quad (2.1) \end{aligned}$$

Where $\mathcal{F} = (\epsilon, \omega, t)$ is the phase space density as function of neutrino energy ϵ , neutrino propagation direction ω , and time t , normalized to be between 0 and 1. The corresponding primed parameters represent the phase space density, neutrino energy and propagation direction before collision. Here $\omega = (\vartheta, \phi)$ is a point on a sphere where $\vartheta \in [0, \pi]$ and $\phi \in [0, 2\pi]$. The kernels $\mathcal{K}^{in/out}$ represent transition rates either in or out of the bin in momentum space which are functions of neutrino energy before and after collision, the cosine of the angle between the unit three-vectors $\mathbf{n}$ and $\mathbf{n}'$, which we will denote as α , run parallel to ω and ω' respectively¹, and the thermodynamic state of the surrounding matter $\mathbf{u} = (\rho, T, Y_e)^T$ where ρ is the mass density, T is the temperature, and Y_e is the electron fraction. The momentum space volume element $dV_P = dV_\epsilon d\omega$ where $dV_\epsilon = \epsilon^2 d\epsilon$ is the volume element of an energy shell. Lastly, θ is a parameter that we set equal to 1 in order to account for Pauli blocking. Doing so ensures that the blocking factors $(1 - \mathcal{F})$ and $(1 - \mathcal{F}')$ go to zero when the respective phase space density is full, thus there are no neutrinos transitioning in or out of that phase space. That way our system adheres to the Pauli exclusion principle. For our purposes, we want to express Eq.(2.1) in terms of the number density of neutrinos $\mathcal{N}$. To simplify the problem, we assume that scattering is isotropic during the neutrino propagation. We do this by approximating the scattering kernel as a Legendre expansion provided by Smit and Cernohorsky [10]:

$$\mathcal{K}^{in/out}(\epsilon, \epsilon', \cos \alpha, \mathbf{u}) \approx \sum_{\ell=0}^L \Phi_\ell^{in/out}(\epsilon, \epsilon', \mathbf{u}) P_\ell(\cos \alpha) \quad (2.2)$$

Where $P_\ell(\cos \alpha)$ is the ℓ^{th} Legendre polynomial. The orthogonality relationship of the Legendre polynomials takes the form:

$$\Phi_\ell^{in/out}(\epsilon, \epsilon', \mathbf{u}) = \frac{2\ell + 1}{2} \int_{-1}^1 \mathcal{K}^{in/out}(\epsilon, \epsilon', \cos \alpha, \mathbf{u}) P_\ell(\cos \alpha) d \cos \alpha \quad (2.3)$$

For approximating the scattering as isotropic, we only consider the first term in the Legendre expansion given above. This removes the angular dependence from the scattering kernels. Lastly, we reduce the number of dimensions we have to integrate over to lower the computational cost by integrating away the ω contribution using the relation found in

¹Here $\mathbf{n} = (\cos \vartheta, \sin \vartheta \cos \phi, \sin \vartheta \sin \phi)^T$

Thorne [11]:

$$\mathcal{N}(\epsilon, \mathbf{x}, t) = \frac{1}{4\pi} \int_0^{2\pi} \int_0^\pi \mathcal{F} \sin \vartheta d\vartheta d\phi \quad (2.4)$$

This zeroth moment formalism is currently used in state of the art simulations for isotropic scattering. Using Eq.(2.4) and substituting the first term of the Legendre expansion, we set $\mathcal{R}^{in/out} = \Phi_0^{in/out}$ and the number density representation of Eq.(2.1) is given by:

$$\frac{d\mathcal{N}}{dt} = (1 - \mathcal{N}) \int_{\mathbb{R}^+} \mathcal{R}^{in}(\epsilon, \epsilon') \mathcal{N}(\epsilon') dV_{\epsilon'} - \mathcal{N} \int_{\mathbb{R}^+} \mathcal{R}^{out}(\epsilon, \epsilon') (1 - \mathcal{N}(\epsilon')) dV_{\epsilon'} \quad (2.5)$$

Here, much like with phase space density $\mathcal{F}$, $\mathcal{N}$ takes on a value between 0 and 1. We have also absorbed a factor of 4π into $dV_{\epsilon'}$. We can interpret the integrals as being over the surface area of the energy shell. Note that we must have conservation of particles during the scattering process. From Cernohorsky [3], this is to say:

$$\mathcal{R}^{in}(\epsilon, \epsilon') = \mathcal{R}^{out}(\epsilon', \epsilon) \quad (2.6)$$

And we also must have the equilibrium condition:

$$\mathcal{R}^{in}(\epsilon, \epsilon') = \mathcal{R}^{out}(\epsilon', \epsilon) e^{\beta(\epsilon' - \epsilon)} \quad (2.7)$$

so that when $\frac{d\mathcal{N}}{dt} = 0$, the exchange in energy and momentum in the scattering process disappears which corresponds to when:

$$\mathcal{N}_{Eq}(\epsilon, \mathbf{u}) = \frac{1}{4\pi} \int_0^{2\pi} \int_0^\pi \mathcal{F}_{Eq} \sin \vartheta d\vartheta d\phi = \frac{1}{1 + e^{\beta(\epsilon - \mu_\nu)}} \quad (2.8)$$

where μ_ν is the neutrino chemical potential.

2.2 Discretization of the Problem

In this section we discretize the energy domain into N_b energy bins for computational purposes with each bin having a center given by:

$$\epsilon_i = \frac{\epsilon_{i-1/2} + \epsilon_{i+1/2}}{2} \quad (2.9)$$

The volume of each bin is:

$$\Delta V_i^\epsilon = \int_{\epsilon_{i-1/2}}^{\epsilon_{i+1/2}} dV_\epsilon = \frac{4\pi}{3}(\epsilon_{i+1/2}^3 - \epsilon_{i-1/2}^3) \quad (2.10)$$

Next we use the finite-volume approach to discretize Eq.(2.5) as follows:

$$\frac{d\mathcal{N}}{dt} \approx (1 - \mathcal{N}) \int_{D^\epsilon} \mathcal{R}^{in}(\epsilon, \epsilon') \mathcal{N}(\epsilon') dV_{\epsilon'} - \mathcal{N} \int_{D^\epsilon} \mathcal{R}^{out}(\epsilon, \epsilon') (1 - \mathcal{N}(\epsilon')) dV_{\epsilon'} \quad (2.11)$$

Where the integrals are now over a finite energy domain D^ϵ . Next, we sum each term over the total number of energy bins, noting that for the i^{th} bin which has a corresponding $\mathcal{N}_i$ number density, the equations take the form:

$$\frac{d\mathcal{N}_i}{dt} = \sum_{k=1}^{N_b} (1 - \mathcal{N}_i) \int_{\epsilon_{k-1/2}}^{\epsilon_{k+1/2}} \mathcal{R}_{ik}^{in}(\epsilon, \epsilon') \mathcal{N}_k(\epsilon') dV_{\epsilon'} - \sum_{k=1}^{N_b} \mathcal{N}_i \int_{\epsilon_{k-1/2}}^{\epsilon_{k+1/2}} \mathcal{R}_{ik}^{out}(\epsilon, \epsilon') (1 - \mathcal{N}_k(\epsilon')) dV_{\epsilon'} \quad (2.12)$$

Assuming that we have constant scattering rates in each bin, we can pull the kernels outside of the integrals in Eq.(2.12). Using the expression for the volume of each bin given by Eq.(2.10) and also making use of the expression:

$$\mathcal{N}_i(t) = \frac{1}{\Delta V_i^\epsilon} \int_{\epsilon_{i-1/2}}^{\epsilon_{i+1/2}} \mathcal{N}(\epsilon, t) dV_\epsilon \quad (2.13)$$

which is the volume averaged particle density in each respective energy bin², we arrive at the expression:

$$\frac{d\mathcal{N}_i}{dt} = (1 - \mathcal{N}_i) \sum_{k=1}^{N_b} \hat{\mathcal{R}}_{ik}^{in} \mathcal{N}_k - \mathcal{N}_i \sum_{k=1}^{N_b} \hat{\mathcal{R}}_{ik}^{out} (1 - \mathcal{N}_k) \quad (2.14)$$

Where $\hat{\mathcal{R}}^{in/out} = \mathcal{R}^{in/out} \Delta V_k^\epsilon$. Before moving forward, let's describe exactly what Eq.(2.14) represents. This equation is describing the rate of change of neutrinos in the i^{th} energy bin of our system. As mentioned earlier, we will be working with a system that has 40 equations so i takes on a value between 1 and 40. During the neutrino-electron scattering process, there will be neutrinos scattering in to and out of a respective i^{th} bin. The neutrinos in the i^{th} bin will interact with the neutrinos from the other bins in the system. So we must sum over all of the other bins to ensure that we are including all of the interactions. These other bins are represented by the k index which also ranges from 1 to 40. The scattering kernels $\hat{\mathcal{R}}^{in/out}$ mediate the transitions in and out of the i^{th} bin, and since that bin interacts with all of the other bins, the kernels have double indices to represent scattering between the i^{th} bin, which has density $\mathcal{N}_i$, and a given k^{th} bin which has density $\mathcal{N}_k$. When $N_i = 1$, corresponding to when the i^{th} bin is full, the first term in Eq.(2.14) vanishes and we no longer have scattering into that bin. This term also vanishes when the distribution equals the local Fermi-Dirac distribution corresponding to when the bin is in equilibrium. When a respective N_k is full, that contribution in the summation in the 2nd term vanishes which means there are no particles scattered in to a k bin as a result from collisions with the i bin. When all of the bins are full, the right hand side of Eq.(2.14) vanishes and there no neutrinos scattering in to or out of the i^{th} bin. These terms vanish when the bins are full because we've included Pauli blocking; a full bin adheres to the Pauli exclusion principle. Lastly, if $i = k$, the right hand side of the equation vanishes due to the symmetry of the kernels which represents that there is no scattering in or out of the i^{th} bin as a result of the bin interacting with itself. As can be seen this system of differential equations are coupled.

²The expression in Eq.(2.13) is given using i as the subscript to be visually consistent with the expressions provided in Eq.(2.9) and Eq.(2.10). In the context of applying it to Eq.(2.12) the subscripts will be in k .

Further manipulating Eq.(2.14) we get:

$$\frac{d\mathcal{N}_i}{dt} = \sum_{k=1}^{N_b} \hat{\mathcal{R}}_{ik}^{in} \mathcal{N}_k - \mathcal{N}_i \sum_{k=1}^{N_b} \left[\hat{\mathcal{R}}_{ik}^{out} + (\hat{\mathcal{R}}_{ik}^{in} - \hat{\mathcal{R}}_{ik}^{out}) \mathcal{N}_k \right]$$

Let's define $F_i^+ = \sum_{k=1}^{N_b} \hat{\mathcal{R}}_{ik}^{in} \mathcal{N}_k$, and $\kappa_i = \sum_{k=1}^{N_b} \hat{\mathcal{R}}_{ik}^{out}$. Lastly, we define:

$$\tilde{\kappa}_i = \sum_{k=1}^{N_b} \left[\delta_{ik} \kappa_k + (\hat{\mathcal{R}}_{ik}^{in} - \hat{\mathcal{R}}_{ik}^{out}) \mathcal{N}_k \right] \quad (2.15)$$

and get the expression:

$$\frac{d\mathcal{N}_i}{dt} = F_i^+ - \tilde{\kappa}_i \mathcal{N}_i = C_i \quad (2.16)$$

Where δ_{ik} is the Kronecker delta. F_i^+ in Eq.(2.16) can be interpreted as the flux of neutrinos in to the i^{th} bin and $\tilde{\kappa}_i$ is a rate parameter tied to neutrino-electron collisions. We can ascertain the importance of the collision rate parameter $\tilde{\kappa}_i$ by considering Eq.(2.16) if F_i^+ and $\tilde{\kappa}_i$ are constant. In this special case, the differential equation has an exact solution which takes the form:

$$\mathcal{N}_i(t) = \mathcal{N}_0 e^{-\tilde{\kappa}_i t} + \frac{F_i^+}{\tilde{\kappa}_i} (1 - e^{-\tilde{\kappa}_i t}) \quad (2.17)$$

Where $\mathcal{N}_0$ represents an initial number density. Here we see that the inverse of the rate parameter $\frac{1}{\tilde{\kappa}_i}$ represents the timescale over which the bin reaches a steady state. We can also define a mean free path by approximating the neutrinos to be traveling at light speed:

$$\ell = \frac{c}{\tilde{\kappa}_i} \quad (2.18)$$

2.3 Matrix Formulation and Stiffness

We can define a vector $\mathcal{N} = (\mathcal{N}_1, \mathcal{N}_2, \dots, \mathcal{N}_{N_b})^T$ such that:

$$\frac{d\mathcal{N}}{dt} = \mathbf{M}(\mathcal{N})\mathcal{N} = \mathbf{C}(\mathcal{N}) \quad (2.19)$$

Where $\mathbf{M}$ is a matrix whose elements are defined by:

$$M_{ik} = \hat{\mathcal{R}}_{ik}^{in} - \delta_{ik}\tilde{\kappa}_k \quad (2.20)$$

Emphasizing the matrix multiplication in Eq.(2.17) with $\dot{\mathcal{N}}_i = \frac{d\mathcal{N}_i}{dt}$, we have the matrix equation:

$$\begin{pmatrix} \dot{\mathcal{N}}_1 \\ \dot{\mathcal{N}}_2 \\ \dot{\mathcal{N}}_3 \\ \vdots \\ \dot{\mathcal{N}}_{N_b} \end{pmatrix} = \begin{pmatrix} M_{11} & M_{12} & M_{13} & \dots & M_{1N_b} \\ M_{21} & M_{22} & M_{23} & \dots & M_{2N_b} \\ M_{31} & M_{32} & M_{33} & \dots & M_{3N_b} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ M_{N_b1} & M_{N_b2} & M_{N_b3} & \dots & M_{N_bN_b} \end{pmatrix} \begin{pmatrix} \mathcal{N}_1 \\ \mathcal{N}_2 \\ \mathcal{N}_3 \\ \vdots \\ \mathcal{N}_{N_b} \end{pmatrix} \quad (2.21)$$

The square matrix, denoted by $\mathbf{M}(\mathcal{N})$, we call the "Collision Matrix" and for our purposes is a 40×40 matrix. This matrix formulation is the basis for how we solve the problem in the code. For explicit methods, this requires a matrix-vector multiplication with each time step. Implicit methods require algebraic methods and thus some need to involve matrix inversions to solve the problem; like in the case of using Newton-Raphson method for Backward Euler for example. For the thermonuclear calculations in Guidry [5], this matrix is sparse. However in our case the matrix is dense, so there will be additional cost concerns particularly as the size of the matrix increases for more complex systems. This is explored more in Fig.(2.1). This figure is a plot of the average time taken to do 10,000 matrix-vector multiplications versus 10,000 matrix inversions for different matrix sizes ranging from 10×10 to 1000×1000 . We use a random number generator to build the elements in the matrices. These calculations are performed in MATLAB, thus the trends highlighted here may be different when using an optimized linear algebra package on a supercomputer. However, here it is clear that on average, matrix inversions for the same size as matrix-vector multiplications take considerably longer, and the time increases with the size of the matrix. Matrix inversions generally will asymptotically scale cubically with the size of the matrix as opposed to quadratically in the case of matrix-vector multiplications. The implications of this will be explored in more detail in Chapter 4.

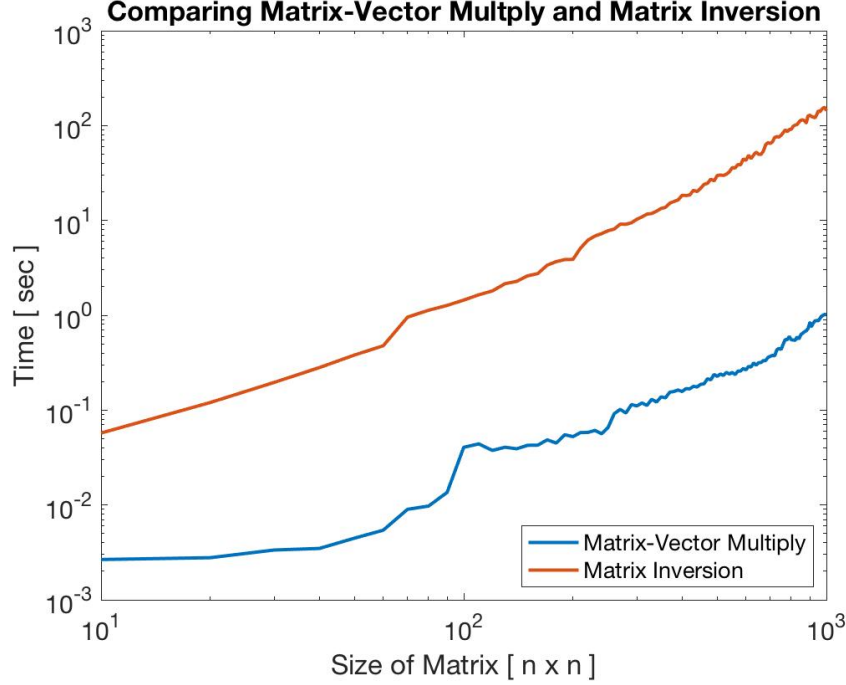


Figure 2.1: Plot of average time taken to do matrix-vector multiplications versus matrix inversions for $n \times n$ matrices. Here it is evident that matrix inversions take considerably longer.

The rate parameters found in the matrix in Eq.(2.21) will change over time due to the occupancy of the bins changing over time due to collisions. Moreover, the varying occupancy of the bins means that the timescales over which these collisions occur can differ substantially from one another. This is a source of stiffness in the system and is ultimately induced by the scattering kernels. To illustrate this, we consider the low occupancy limit in Eq.(2.15) whereby $\mathcal{N}_k \ll 1$. Fig.(2.2) shows a plot of the mean collision time $\frac{1}{\kappa}$ for all energies in the system for various proto-neutron star densities taken from models in spherically symmetric supernova simulations found in [7], the details of these models is illustrated in the table below.

Table 2.1: Table of models used in thesis. Further information can be found in [7]

Name	Density [g/cm ³]	kT [MeV]
High	1.0×10^{14}	20.5399
Intermediate	1.0×10^{12}	7.7141
Low	1.0×10^{10}	3.1448

The names of these models are designated for the purposes of this thesis. For comparison, we also include the explosion time which will be on the order of seconds (we've set it to 1 second in the plot), and a hydrodynamic timescale. We use the definition of the hydrodynamic timescale as the amount of time it takes a sound wave in the star material to propagate some distance. For our purposes, we assume a wave travels at 10 percent the speed of light over a distance of 1 km. For the highest density case, we see that the mean collision time is $\sim 10^{-6}$ for the lowest energies and $\sim 10^{-9}$ for the highest indicating the stiffness induced by the scattering kernel (we discuss what effect this has on explicit calculations in chapter 3). Furthermore, the middle density case represents a transition period where the mean collision time exceeds the hydrodynamic timescale and thus neutrino-electron collisions are occurring slower than sound wave propagation in the star material at low energies.

Likewise, Fig.(2.3) shows the mean free path for all of the energies. Intuitively, we see that collisions are more common at higher densities, particularly at higher energies. An estimate of the size of a proto-neutron star and shock radius given by Bruenn [2], which we consider to be 40 km and 100 km respectively, shows that neutrinos are able to escape the proto-neutron star and the shock radius for low densities at low energies. The transition between where neutrinos are trapped inside the star and where they can escape occurs in the middle density scenario much like when the collision times are longer than wave propagation in the star. If the neutrinos are able to disperse from the star, collisions will be less frequent as a result of the surrounding material being less dense. Now that we've constructed our model and have the equations that must be solved, let's introduce methods that can be used to solve this system.

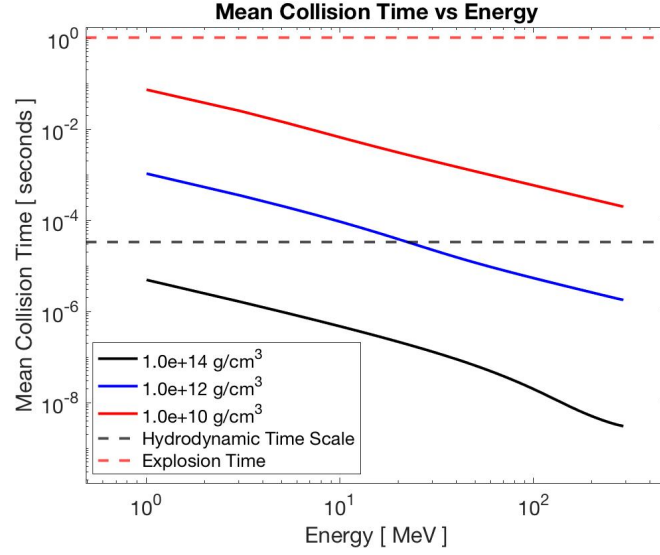


Figure 2.2: Plot of mean collision times for all energy bins in the low occupancy limit. The collision times for all densities differ by orders of magnitude across all energies, indicating stiffness that's induced by the scattering kernel.

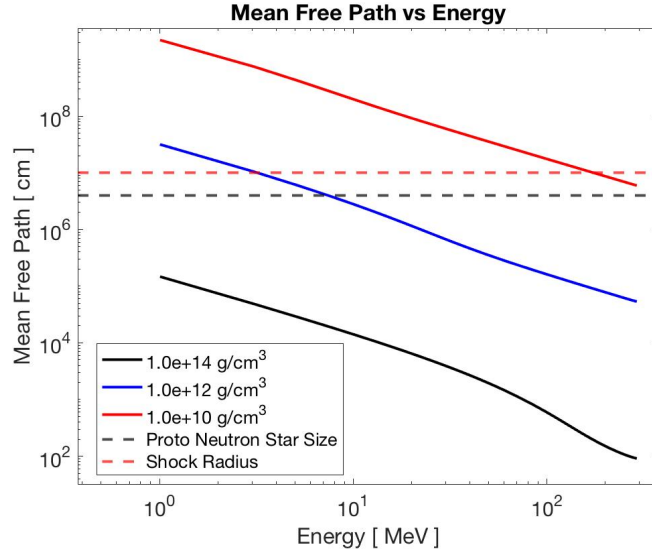


Figure 2.3: Plot of mean free path for all energy bins in the low occupancy limit. For the higher energy bins, collisions are substantially more common than for the lower energy bins.

Chapter 3

Numerical Methods

In this chapter we will be exploring the different methods that will be used to solve the equations. For the purposes of simplicity, some of the derivations shown in this chapter will be for only a single equation represented by Eq.(2.16). The actual applications of the methods to solve a system of equations taking the form of Eq.(2.16) will be done using the matrix formulation described in Chapter 2 Section 3.

All of the methods described in this chapter are designed to solve the problem using a time step that evolves from $t = 0$ to a specified t_{end} which will vary depending on the density condition. The idea is to run the calculations until we achieve a steady state so the time required will be different for low and high PNS densities. Our initial time step is 10^{-15} seconds. With each iteration, we grow our time step by a factor which we call dt_{grow} such that $\Delta t^{n+1} = dt_{grow} \Delta t^n$. Values for dt_{grow} commonly used in this study are 1.003, 1.03, and 1.20 which we refer to as small, intermediate, and aggressive (or large) time stepping. We use a Gaussian function as the initial condition centered on 100 MeV with PNS mass densities of 1.0×10^{14} g/cm³, 1.0×10^{12} g/cm³, and 1.0×10^{10} g/cm³ which we refer to as high, intermediate (or medium), and low densities, respectively. This initial condition is far from equilibrium. We also set t_{end} to be 0.01, 0.1, and 1 second for the high, intermediate, and low PNS density cases, respectively. We design the code so that it outputs 100 data values over the course of a run. Building the collision matrix and rate parameters uses expression

provided by Bruenn [1] in Appendix C to calculate $\mathcal{R}_{ik}^{in}$ for neutrino-electron scattering. We impose the symmetry condition for the scattering kernels in Eq.(2.6) to construct $\mathcal{R}_{ik}^{out}$.

3.1 Explicit Methods

Explicit methods are numerical treatments in which $\mathcal{N}_i^{n+1}$, in our case, is calculated strictly from terms on the right hand of the iteration equation which depend on $\mathcal{N}_i^n$. As detailed in the introduction of this thesis, standard explicit methods have stability constraints which limit the size of the time step we can use and thus limits how fast we can solve the equations.

3.1.1 Forward Euler Method

The first explicit method that will be detailed is the Forward Euler (FE) method which when applied to our differential equations given by Eq.(2.16) takes the form:

$$\mathcal{N}_i^{n+1} = \mathcal{N}_i^n + \Delta t C_i^n \quad (3.1)$$

where $C_i^n = \frac{d\mathcal{N}_i^n}{dt} = F_i^+ - \tilde{\kappa}_i \mathcal{N}_i^n$ and Δt is the time step. Once the time step has reached the stability constraint $\tau_{FE} = \frac{1}{\tilde{\kappa}_i}$, we progress each step with τ_{FE} . Because of the stability condition of FE, this is not a method that's practical to use for a realistic neutrino transport calculation. The mean collision times for different energies, which is given by the expression which is also FE's largest stable time step, is reflected in Fig.(2.1). We can see from the figure that for the highest density case the mean collision time at high energies is on the order of 10^{-9} seconds. If we wanted to integrate these equations over the course of an entire explosion time of 1 second, it would take on the order of 10^9 runs using FE's largest stable time step which makes this method costly. However, for our scaled down model it can be used as an accuracy test for the other methods in the subsequent sections. A small growth factor of $dt_{grow} = 1.0001$ is used for this method to minimize computational error as this will be important when comparing other methods and determining their accuracy. An important question to address for all of the methods used in this study is whether or not they conserve

the total particle number during each iteration. In the case of FE, we can demonstrate this noting that the total particle number at a particular time t is given by the expression:

$$N_{tot}(t) = \int_{\mathbb{R}^+} \mathcal{N}(\epsilon, t) dV_\epsilon \approx \sum_{i=1}^{N_b} \mathcal{N}_i(t) \Delta V_i^\epsilon \quad (3.2)$$

Multiplying both sides of Eq.(3.1) by ΔV_i^ϵ and summing over all i indices we arrive at the expression:

$$N_{tot}^{n+1} = N_{tot}^n + \Delta t \sum_{i=1}^{N_b} C_i^n \Delta V_i^\epsilon \quad (3.3)$$

Focusing now on the summation in the 2nd term we get:

$$\begin{aligned} \sum_{i=1}^{N_b} C_i^n \Delta V_i^\epsilon &= \sum_{i=1}^{N_b} \sum_{k=1}^{N_b} [(1 - \mathcal{N}_i) \mathcal{R}_{ik}^{in} \mathcal{N}_k - (1 - \mathcal{N}_k) \mathcal{R}_{ik}^{out} \mathcal{N}_i] \Delta V_i^\epsilon \Delta V_k^\epsilon \\ &= \sum_{i=1}^{N_b} \sum_{k=1}^{N_b} (\mathcal{R}_{ik}^{in} - \mathcal{R}_{ki}^{out}) (1 - \mathcal{N}_i) \mathcal{N}_k \Delta V_i^\epsilon \Delta V_k^\epsilon \\ &= 0 \end{aligned}$$

Which we arrive from expanding the summations and gathering like terms of $(1 - \mathcal{N}_i) \mathcal{N}_k$. The 2nd term in Eq.(3.3) vanishes due to the symmetry of the scattering kernels, recalling Eq.(2.6)¹. Thus we get that $N_{tot}^{n+1} = N_{tot}^n$ meaning that FE conserves particle number at each step and can be safely used. Lastly, for illustrative purposes, Fig.(3.1) shows the time evolution of the neutrino distribution for all energies in the high PNS density case starting from our initial time step. These plots were produced using a FE calculation. We can see in the first plot our Gaussian initial condition which then evolves over the course of the run to a Fermi-Dirac distribution. This is to better highlight exactly how the distribution function evolves.

¹ Here, $\mathcal{R}_{ik}^{in/out} = \mathcal{R}^{in/out}(\epsilon_i, \epsilon_k)$

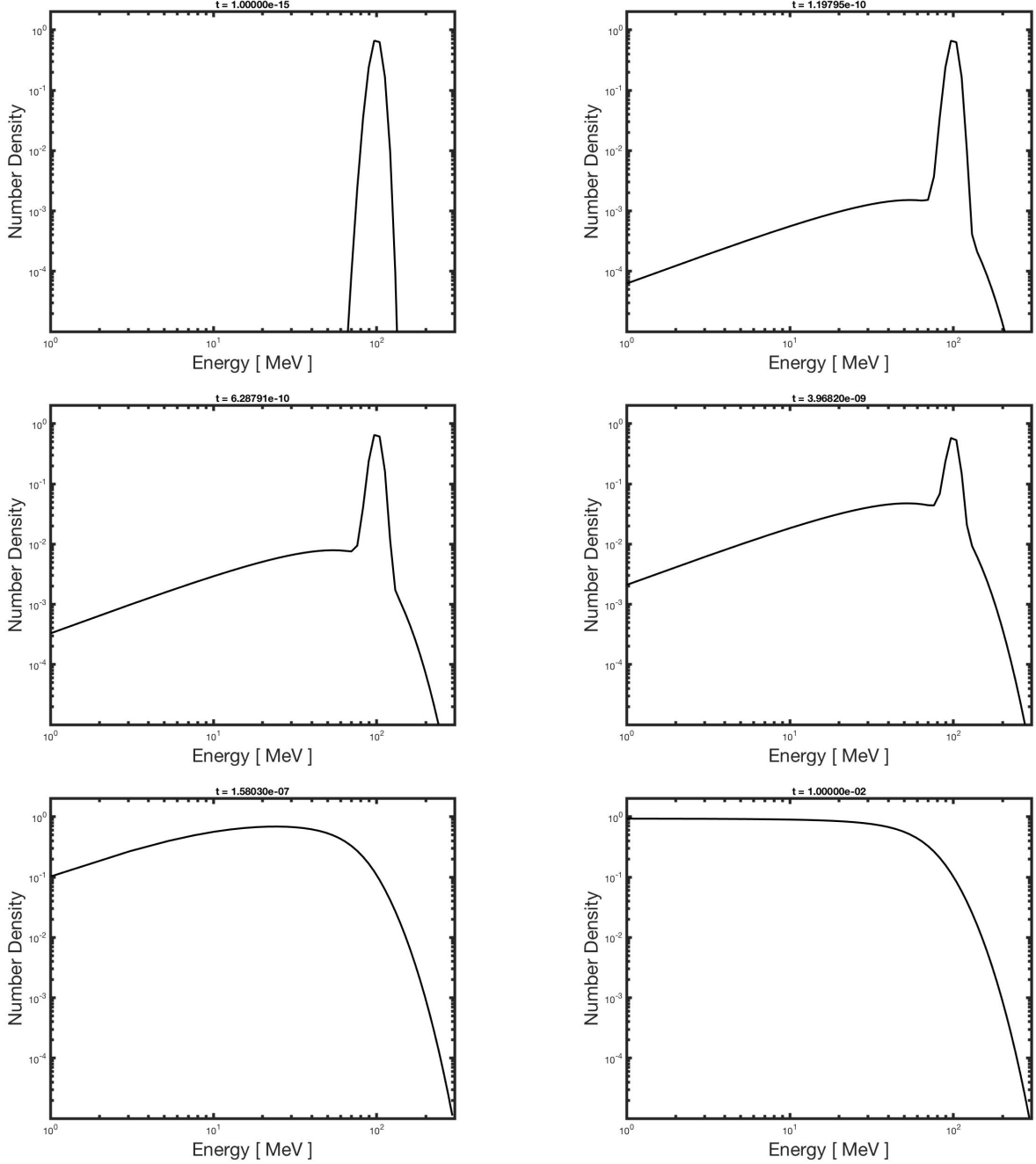


Figure 3.1: Time evolution of neutrino distribution function for all energies.

3.1.2 Explicit Asymptotic Approximation

The Explicit Asymptotic method (EA) as highlighted by [5] is an approach that builds an iterative expression that is an approximation when the entire system is far from equilibrium. This method is applied to expressions that are of the form of Eq.(2.16) where we can solve

for the density and arrive at the expression:

$$\mathcal{N}_i = \frac{1}{\tilde{\kappa}_i} \left(F_i^+ - \frac{d\mathcal{N}_i}{dt} \right) \quad (3.4)$$

and consider the asymptotic limit where $\frac{d\mathcal{N}_i}{dt} \rightarrow 0$ corresponding to when $F_i^+ \approx \tilde{\kappa}_i$. Using a finite-difference approximation for the derivative at a particular iteration we get:

$$\frac{d\mathcal{N}_i^n}{dt} = \frac{\mathcal{N}_i^{n+1} - \mathcal{N}_i^n}{\Delta t} - \frac{\Delta t}{2} \frac{d^2\mathcal{N}_i^n}{dt^2} + \dots \quad (3.5)$$

where we truncate at the first term given that the derivative is small. Using this we can build an approximate expression for the $\mathcal{N}_i^{n+1}$ step as follows:

$$\begin{aligned} \mathcal{N}_i^{n+1} &= \frac{1}{\tilde{\kappa}_i} \left(F_i^+ - \frac{d\mathcal{N}_i^n}{dt} \right) \\ &= \frac{F_i^+}{\tilde{\kappa}_i} - \frac{1}{\tilde{\kappa}_i} \frac{d\mathcal{N}_i^n}{dt} \\ &= \frac{F_i^+}{\tilde{\kappa}_i} - \frac{1}{\tilde{\kappa}_i} \left(\frac{\mathcal{N}_i^{n+1} - \mathcal{N}_i^n}{\Delta t} \right) \end{aligned}$$

Solving for $\mathcal{N}_i^{n+1}$:

$$\begin{aligned} \mathcal{N}_i^{n+1} \left(1 + \frac{1}{\tilde{\kappa}_i \Delta t} \right) &= \frac{F_i^+}{\tilde{\kappa}_i} + \frac{\mathcal{N}_i^n}{\tilde{\kappa}_i \Delta t} \\ \mathcal{N}_i^{n+1} &= \frac{\mathcal{N}_i^n + \Delta t F_i^+}{1 + \tilde{\kappa}_i \Delta t} \\ &= \mathcal{N}_i^n + \frac{\Delta t}{1 + \tilde{\kappa}_i \Delta t} (F_i^+ - \tilde{\kappa}_i \mathcal{N}_i^n) \\ &= \mathcal{N}_i^n + \frac{\Delta t C_i^n}{1 + \tilde{\kappa}_i \Delta t} \end{aligned} \quad (3.6)$$

As can clearly be seen, in order to calculate the $n^{th} + 1$ step in the iteration we just need to know the n^{th} step, starting from our initial condition and progressing forward. Hence, this indeed is an explicit method. As stated by Guidry, this method is expected to provide

stable solutions for $|\tilde{\kappa}_i \Delta t| \gg 1$, when the time step is larger than FE's stability condition. Another thing to note that if we define $\tau_{EA} = \frac{\Delta t}{1 + \tilde{\kappa}_i \Delta t}$, Eq.(3.6) looks similar to FE thus emphasizing the simplicity in implementation. Now we must check whether EA conserves particle number. Using the same method to check if FE was conservative we get:

$$N_{tot}^{n+1} = N_{tot}^n + \Delta t \sum_{i=1}^{N_b} \frac{C_i^n}{1 + \tilde{\kappa}_i \Delta t} \Delta V_i^\epsilon \quad (3.7)$$

We expand the denominator in τ_{EA} using the Taylor Series expansion:

$$\frac{1}{1 + \tilde{\kappa}_i \Delta t} = \sum_{m=0}^{\infty} (-1)^m (\tilde{\kappa}_i \Delta t)^m \quad (3.8)$$

And Eq.(3.7) becomes:

$$\begin{aligned} N_{tot}^{n+1} &= N_{tot}^n + \Delta t \sum_{i=1}^{N_b} \sum_{m=0}^{\infty} C_i^n \Delta V_i^\epsilon (-1)^m (\tilde{\kappa}_i \Delta t)^m \\ &= N_{tot}^n + \Delta t \sum_{i=1}^{N_b} C_i^n \Delta V_i^\epsilon + \sum_{i=1}^{N_b} \sum_{m=1}^{\infty} C_i^n \Delta V_i^\epsilon (-1)^m (\tilde{\kappa}_i)^m (\Delta t)^{m+1} \end{aligned} \quad (3.9)$$

Here it is clear that EA agrees with FE up to $\mathcal{O}(\Delta t^2)$. The 2nd term in Eq.(3.9) will vanish due to the symmetry of the kernels as demonstrated in the FE case. The issue is the 3rd term. Because EA is expected to be valid for $|\tilde{\kappa}_i \Delta t| \gg 1$, the expansion of τ_{EA} does not converge. Applying this method naively is potentially problematic as we cannot guarantee that EA conserves particle number because the result of those summations is undefined. In order to address this concern, we introduce a tolerance parameter $tolC$ and impose a condition that checks the fractional difference of total particles from one iteration to the next with the constraint that:

$$\left| \frac{N_{tot}^{n+1} - N_{tot}^n}{N_{tot}^n} \right| \leq tolC \quad (3.10)$$

As before with FE, and for the implicit methods in the subsequent sections, we evolve the time step by a factor dt_{grow} . Because the non vanishing terms in Eq.(3.9) depend on the size of the time step, if the tolerance condition in Eq.(3.10) is not satisfied we need to readjust

the time step and redo that iteration with the modified time step. For the purposes of this study, we've decided to reduce the time step by a factor of 0.90 every time the tolerance condition isn't met. Depending on how strict we want to conserve particle number, this can cause our evolving time step to diverge from linearity when plotted on a logarithmic scale. We also want to reduce the error in density between one time step and the next. Having stable solutions does not guarantee that they are accurate. In order to do this, we implement another tolerance condition whereby²:

$$\max\left[\frac{|\mathcal{N}^{n+1} - \mathcal{N}^n|}{\max(\mathcal{N}^n, 10^{-8})}\right] \leq \text{tol}\mathcal{N} \quad (3.11)$$

Where we set $\text{tol}\mathcal{N} = 0.01$ and set the denominator to be no smaller than 10^{-8} . If this condition is violated, we again redo the time step using the same method as with the particle conservation condition. Furthermore, in order to keep record of how fast the method is solving the problem, we decided to count the number of right hand evaluations in Eq.(3.6) that are required in the calculation. We define a parameter called "truecycle" that counts the number of times the algorithm iterates which also includes the steps where the time step is adjusted due to violation of our tolerance condition. We also define a parameter "cycle" which counts the number of iterations excluding the steps where the time step is adjusted. This will be important when comparing this method to standard implicit methods used to solve this system of differential equations. However, because this method is non-conservative, we don't want to use it if the time steps are small enough that FE can be used. Because of that, we've structured the algorithm so that as long as the time steps are below the stability condition of FE, use FE to solve the problem. After the time steps have exceeded this constraint, update using the EA. A more comprehensive algorithm is given below:

1. Update truecycle by an increment of 1.
2. Build collision matrix and calculate rate parameters $\tilde{\kappa}_i$.
3. Check if $\Delta t \leq \frac{1}{\tilde{\kappa}_i}$. If so, update with FE. If not, update with EA.

²Here the number densities are in bold to indicate that they are vectors.

4. Check tolerance conditions in Eq.(3.10) and Eq.(3.11). If either condition is violated, decrease time step by 0.90 and loop back to step one using adjusted Δt . If not, increase Δt by dt_{grow} factor and loop back to step 1.

We expect this to cut down the cost of solving the problem; the region where we are no longer updating the time step linearly using FE because of the stability constraint is the most costly because the steps remain constant in time. Fig.(3.2) shows a plot of the particle number density vs. time for an EA calculation that used a $tolC = 1.0 \times 10^{-6}$ and $dt_{grow} = 1.01$ for the highest density model. This plot is for the energy bins of 1.0064, 10.0489, 44.40869, 104.1867, and 210.3028 MeV. In solid lines are the EA curves and the FE results in the same plot are dashed. As can be seen, the overlap of the curves suggest that the EA method is solving the problem. To quantify the accuracy of this method compared to FE, Fig.(3.3) shows the relative difference of the number densities for the same energy bins compared to FE. We compute this difference using the expression:

$$Error = \frac{\mathcal{N}_{FE} - \mathcal{N}}{\mathcal{N}} \quad (3.12)$$

Where $\mathcal{N}_{FE}$ is the particle number density from the FE calculations. We also define a change in particle number from one iteration to the next which is given by:

$$dN = \frac{N_{tot}^{n+1} - N_{tot}^n}{N_{tot}^n} \quad (3.13)$$

For the purposes of this study, we define acceptable particle number conservation if Eq.(3.12) and Eq.(3.13) are $\sim 10^{-2}$.

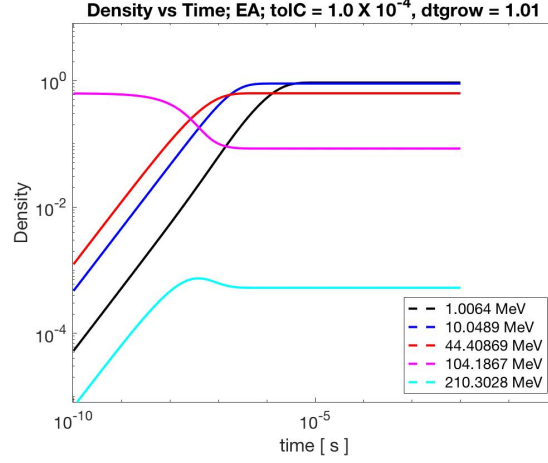


Figure 3.2: Plot showing the time evolution of number densities for five arbitrary energy bins. EA is solid and FE is dashed and they appear to overlap.

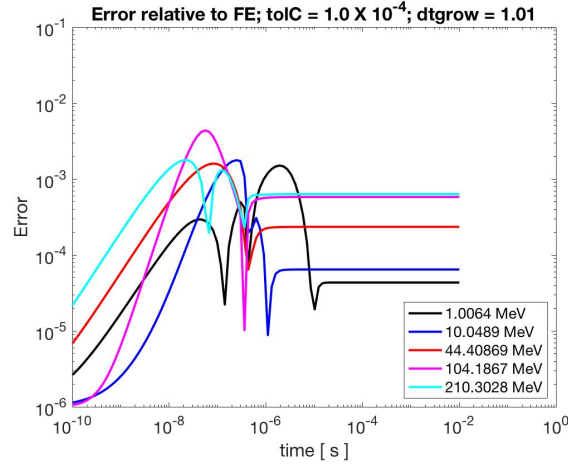


Figure 3.3: Relative difference between EA and FE number densities for the same energy groups in Fig.(3.2). The error never exceeds 10^{-2} for any energy bin. As such the EA method solves the equations while being reasonable accurate with FE even for a tight tolerance on particle number error.

The accuracy of EA with respect to FE depends on the aggressiveness of our time stepping and how tight we want our tolerance to be. For example, Fig.(3.4) and Fig.(3.5) shows a density and relative error plot, respectively, for a case where the $dt_{grow} = 1.20$ and $tolC = 1.0 \times 10^{-2}$, aggressive time stepping with a loose tolerance.

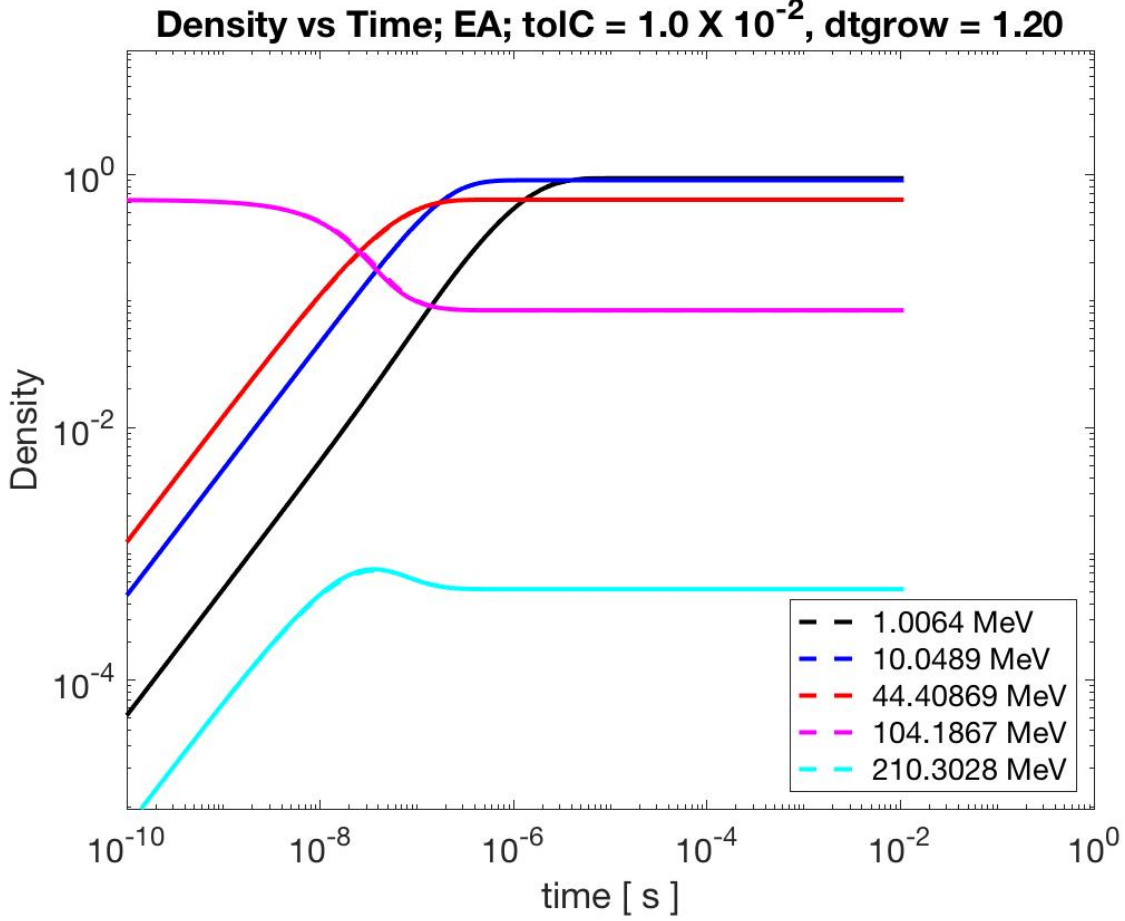


Figure 3.4: Plot showing the time evolution of number densities for five arbitrary energy bins. EA is solid and FE is dashed and they appear to overlap.

We can see from Fig.(3.3) that the offset from FE still isn't very noticeable, a consequence of the tolerance condition in Eq.(3.11). In conjunction, the relative error for the two highest energy bins maxes at around 5%, noticeably higher than before. Next we will see what implications these parameters can have when it comes to time stepping and particle number conservation. In principle, we want to have as competitive time stepping with implicit methods as possible, meaning we want to minimize how often the algorithm has to redo steps, while also conserving particle number to a reasonable tolerance and minimizing error relative to FE.

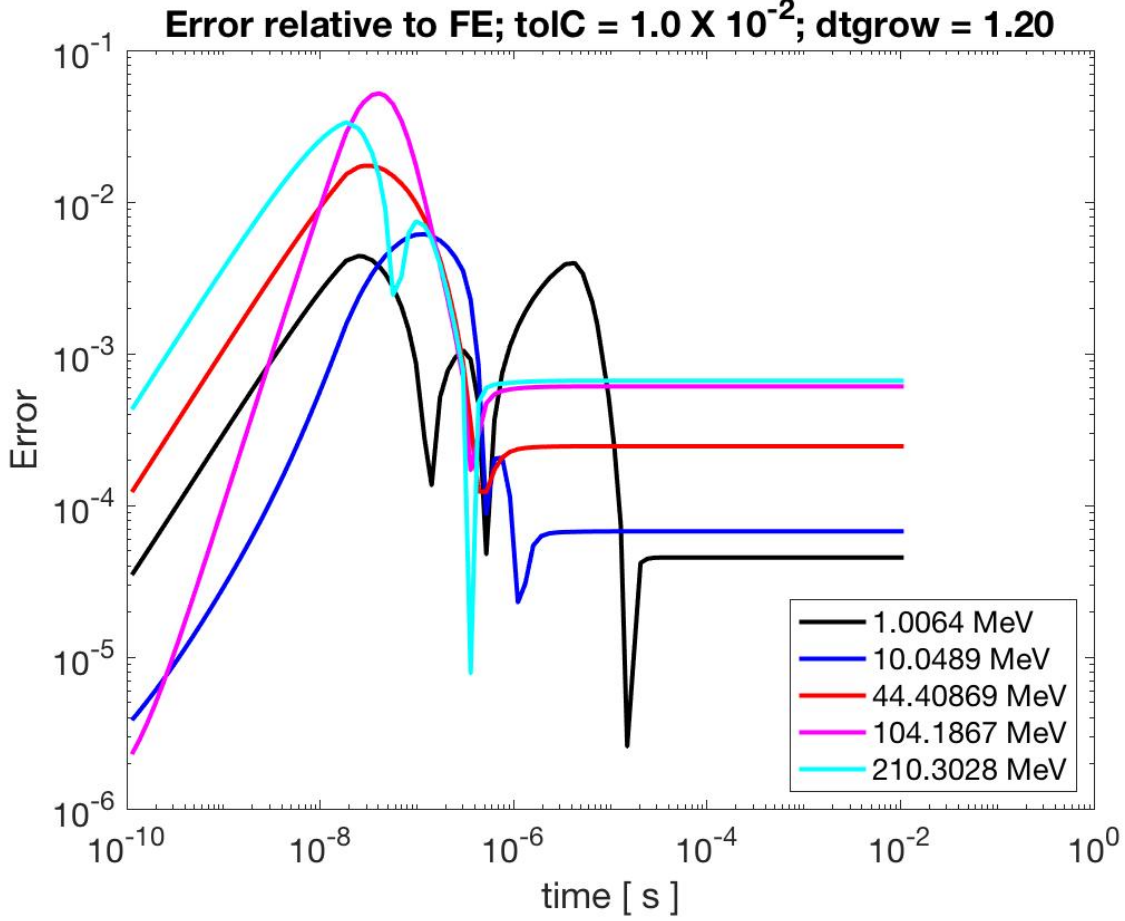
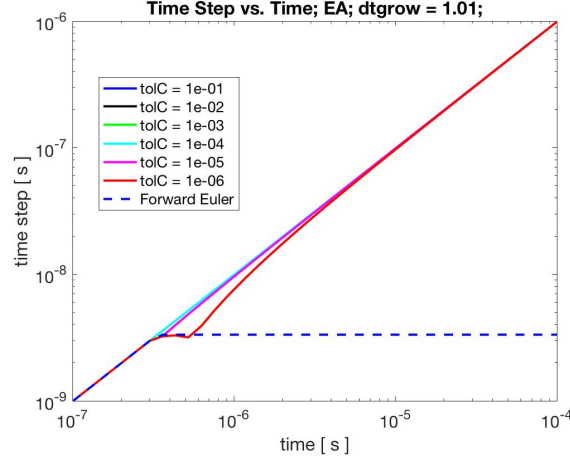


Figure 3.5: Relative difference between EA and FE number densities for the same energy groups in Fig.1. The error maxes out at $\sim 5\%$. As such the EA method solves the equations while being reasonable accurate with FE even for a tight tolerance on particle number error.

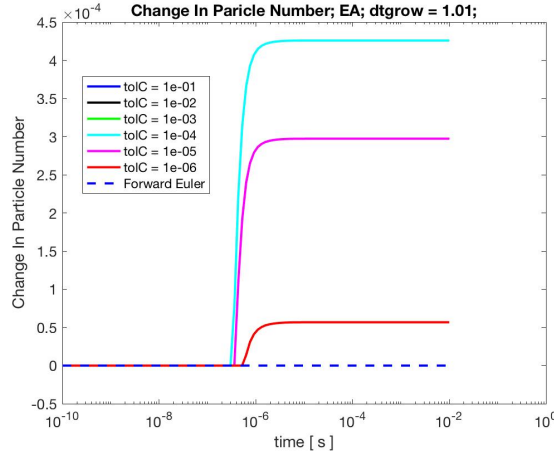
The time evolution of the time step for different tolerances as well as the change in particle number vs. time for different tolerances in the high density model is shown in Fig.(3.6a) and Fig.(3.5b), respectively. As shown in Fig.(3.6a), the time stepping evolves linearly at a $dt_{grow} = 1.01$ for $tolC \leq 10^{-4}$, afterward we see divergence from linearity. This is due to the EA method not conserving particle number so the algorithm has to adjust the time step more often for tighter tolerances. Having said that, even at the lower tolerances we see a change in particle number on the order of 10^{-4} which we deem acceptable for our calculations. We compute this change from one iteration to the next using the left hand expression in Eq.(3.10). The plots for FE are found in the same figure. We see the FE time steps evolve linearly until it flatlines due to the stability condition. The change in particle

number for FE is always zero because the method is conservative. We compare and contrast time step evolutions in the high density case for different dt_{grow} in Fig.(3.7) - (3.9), for 1.003, 1.03, and 1.20 respectively. For more aggressive time stepping we see that the time step evolution diverges from linearity for smaller tolerances. Because the non-vanishing terms in Eq.(3.9) depend on the time step, a more aggressive time step will lead to the algorithm having to readjust the time step more often. Because the time stepping is sensitive to dt_{grow} and $tolC$ we must be careful when selecting these parameters as some combination may lead to calculations slower than standard implicit methods. Furthermore, from Fig.(3.10) - (3.12) we see that the change in particle number in the high density case also varies with dt_{grow} and $tolC$. For example in Fig.(3.11) we have a change in particle number on $\sim 10^{-4}$ for $dt_{grow} = 1.03$ for all of the tolerances with the tighter tolerances generating the best conservation (though worst time stepping). Much the same when $dt_{grow} = 1.20$, as shown in Fig.(3.12). Looking at particle conservation alone suggests that it may be better to use dt_{grow} of 1.20 in a calculation over the other scenarios because even for large time steps we get acceptable particle number conservation. However, there are time stepping concerns as well, hence It may even be better to use 1.03 since the time stepping is more stable for all of the tolerances and the calculation conserves particle number to an acceptable limit as well. This will be addressed in full when we include the number of iterations in Chapter 4 for all methods discussed in this thesis. More is brought into clarity once we include how cycle and truecycle vary with these parameters. Tab.1 shows a comparison of cycle and truecycle for $dt_{grow} = 1.003$ for different tolerances. Here, cycle and truecycle are always equal which is why the time stepping is always linear in Fig.(3.7). The slowly evolving time stepping ensures that the algorithm never has to re-step which holds true even for tight tolerances. Thus, all calculations in this scenario take the same amount of time irrespective of how tight $tolC$ is set. Tab.2 shows cycle and true cycle for the 1.03 case. We see that for the tighter tolerance conditions the algorithm has to re-step as the time steps get larger. Contrasting that to the most aggressive time stepping found in Tab.3, the 1.20 case has to re-step earlier in the tolerance order and the offset between truecycle and cycle is proportionately larger. Ultimately how the time stepping compares to implicit methods and the consequences therein

will address which parameters are best to yield the desirable results. In saying that, let us now introduce the implicit methods used in this study.



(a) Logarithmic plot of time step vs. time for different tolerances. Time step evolution for tolerances tighter than 10^{-4} diverge from linearity due to the method being non conservative.



(b) Logarithmic plot of change in particle number vs. time. For all tolerances shown we get acceptable particle number conservation. Plots for 10^{-1} to 10^{-4} overlap.

Figure 3.6: Time stepping and particle conservation sample plots. The goal is to achieve competitive time stepping with implicit methods while also minimizing the change in particle number.

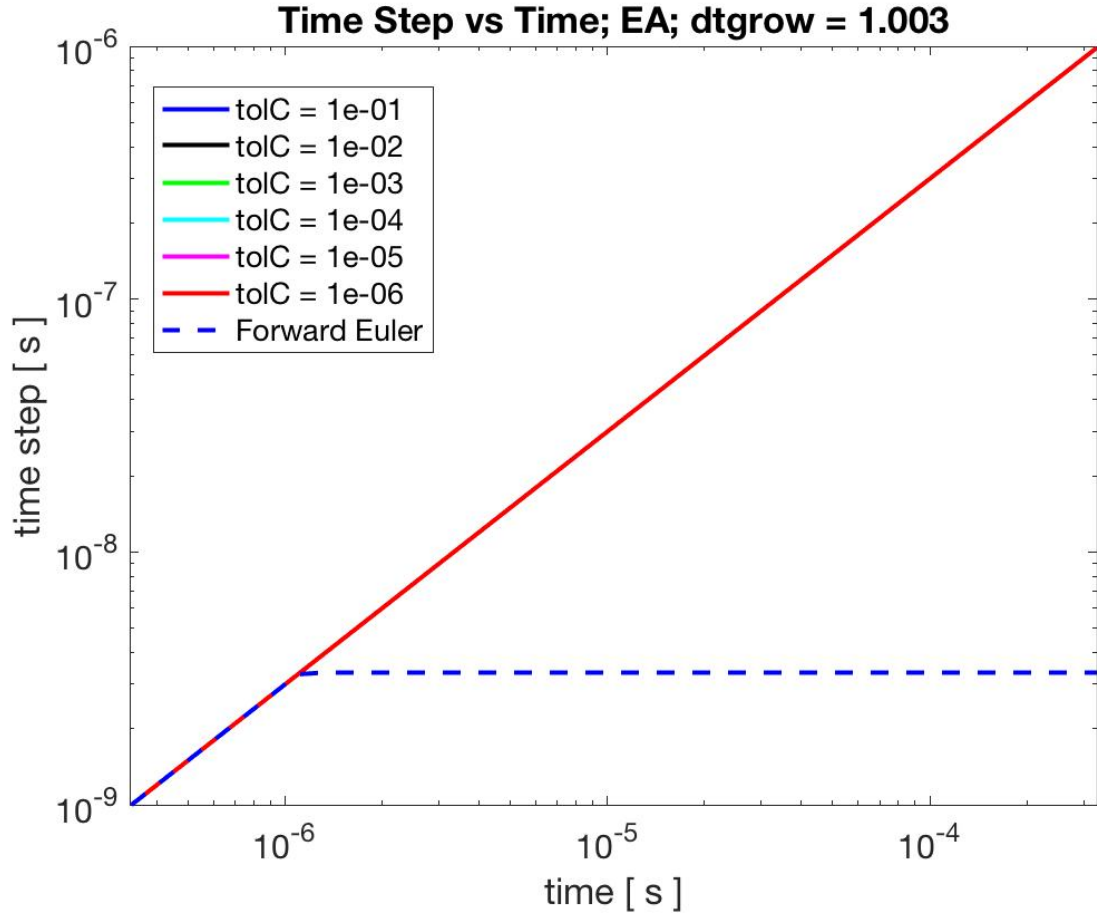


Figure 3.7: Time stepping evolution for EA in high density case. Small dt_{grow} . Evolution for all tolerances is linear due the algorithm never having to adjust the time step.

Table 3.1: Cycle and Truecycle for $dt_{grow} = 1.003$

tolC	Cycle	Truecycle
1.0×10^{-1}	8053	8053
1.0×10^{-2}	8053	8053
1.0×10^{-3}	8053	8053
1.0×10^{-4}	8053	8053
1.0×10^{-5}	8053	8053
1.0×10^{-6}	8053	8053

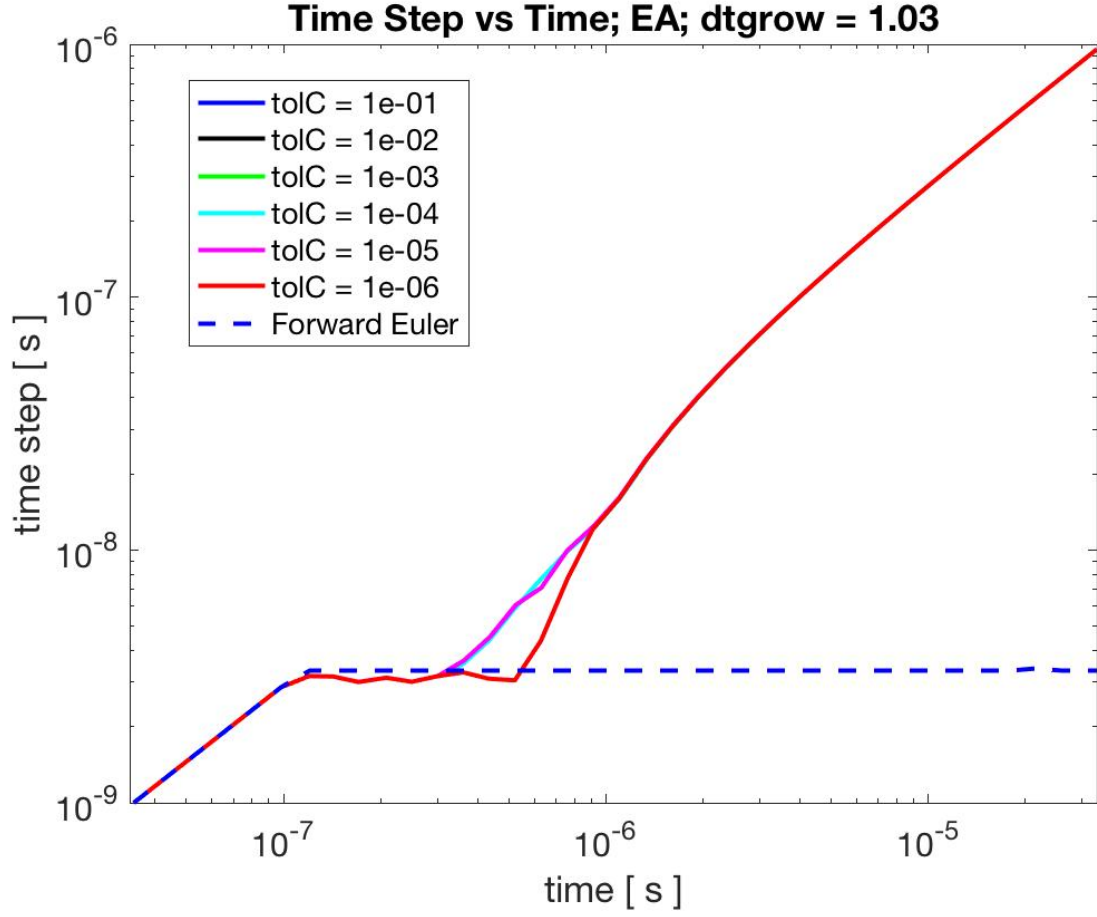


Figure 3.8: Time stepping evolution for EA in high density case. Intermediate dt_{grow} . We see more pronounced diverging from linearity due to time steps being too large and violating the tolerance conditions and thus needing adjustment.

Table 3.2: Cycle and Truecycle for $dt_{grow} = 1.03$

tolC	Cycle	Truecycle
1.0×10^{-1}	1011	1044
1.0×10^{-2}	1011	1044
1.0×10^{-3}	1011	1044
1.0×10^{-4}	1011	1044
1.0×10^{-5}	1011	1044
1.0×10^{-6}	1050	1094

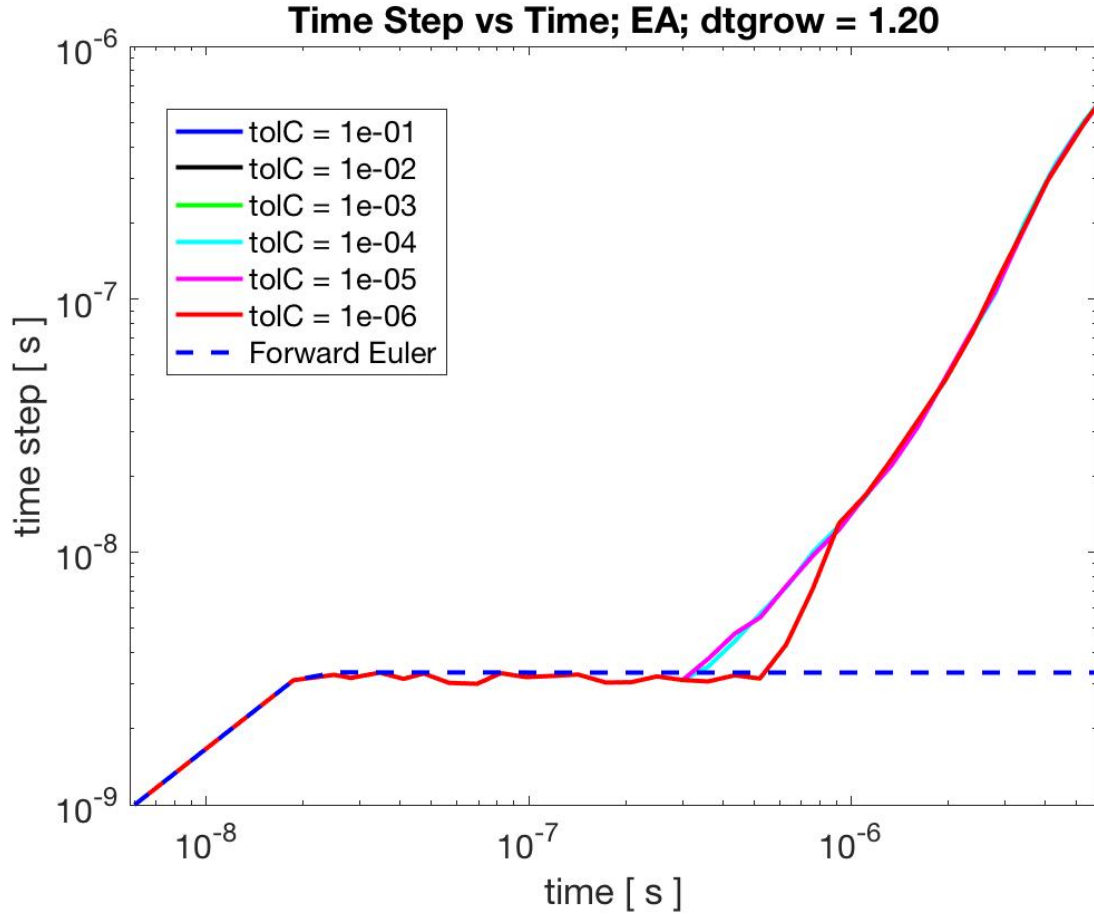


Figure 3.9: Time stepping evolution for EA in high density case. Large dt_{grow} . This divergence from a linear evolution exacerbates with more aggressive time stepping.

Table 3.3: Cycle and Truecycle for $dt_{grow} = 1.20$

tolC	Cycle	Truecycle
1.0×10^{-1}	379	767
1.0×10^{-2}	379	767
1.0×10^{-3}	379	767
1.0×10^{-4}	379	767
1.0×10^{-5}	380	769
1.0×10^{-6}	419	876

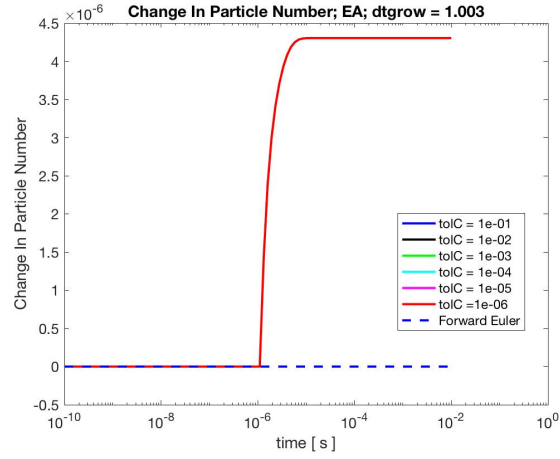


Figure 3.10: EA particle conservation for small time stepping. Small dt_{grow} yields better particle conservation due to the tolerance condition never being violated.

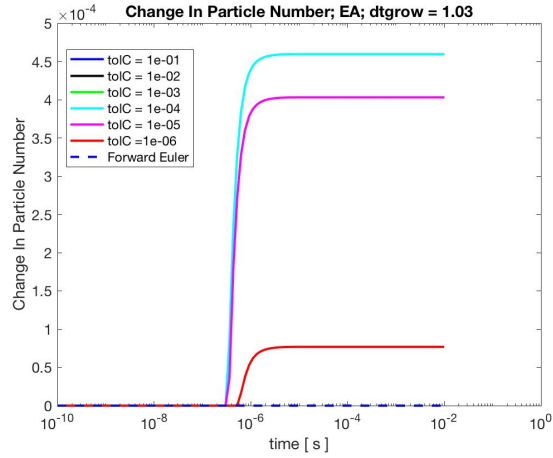


Figure 3.11: EA particle conservation for intermediate time stepping. Here the change doesn't exceed 1% for a given tolerance however tolerances greater than 10^{-3} don't yield ideal time stepping as shown in Fig.4(b).

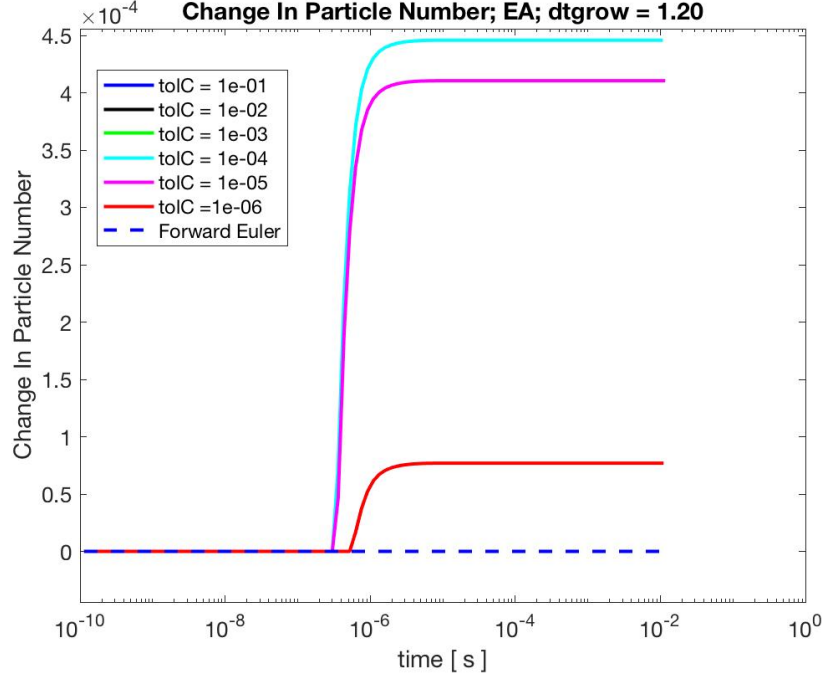


Figure 3.12: EA particle conservation for large time stepping. Aggressive time stepping can lead to the algorithm losing efficiency.

3.2 Implicit Methods

Unlike with explicit methods, implicit methods have $\mathcal{N}_i^{n+1}$ on both sides of the iterative expression, so the particle number density for future time steps will need to be solved by algebraic means of some sort. The benefit of these methods is that we can take larger time steps and maintain stable solutions than in the case of FE or some other standard explicit method. Thus, we can solve the problem faster. To reiterate, this is why implicit methods are standard for solving stiff systems of equations and why they will serve as the measuring stick for EA. However, in solving large systems there are efficiency concerns brought about by these algebraic calculations that must be performed per time step. In the context of a matrix formulation, this manifests in having to do a series of matrix inversions per time step which we've shown is more costly than simple matrix-vector multiplications. Much like with EA, we run FE in the region where the time steps are smaller than FE's stability condition.

Again, these solutions being stable does not mean they are accurate. So, much like in the case of EA, for all of these implicit methods we implement the condition highlighted in Eq.(3.11) to minimize the error in $\mathcal{N}_i$ between one step and the next, also instructing the algorithm to redo that time step with a smaller time step (multiplying dt_{grow} by 0.90) if this condition is violated. We also use the same $tol\mathcal{N} = 0.01$. Plots in this chapter will primarily focus on relative error from FE and particle number conservation to show that the methods are indeed solving the equations and producing acceptable results. We will explore the behavior of the time stepping for implicit methods in greater detail in Chapter 4. In saying that, let us now summarize the implicit methods used in this study in more detail.

3.2.1 Backward Euler Method

A standard implicit method that we will use as a speed test for EA is the Backward Euler (BE) method. BE method when applied to the equations we're solving will take the form:

$$\mathcal{N}_i^{n+1} = \mathcal{N}_i^n + \Delta t C_i^{n+1} \quad (3.14)$$

With the initial condition being the gaussian function detailed earlier. Unlike FE, BE conserves particle number only to a given tolerance. In order to solve for the density at a point in the future, it requires us knowing how the system behaves in the future. Because of this, implicit methods are generally more difficult to implement. Thus we need to solve an algebraic equation for $\mathcal{N}_i^{n+1}$. We do this using the Newton-Raphson Method and define a tolerance parameter $tolBE$ which is a convergence condition for the Newton-Raphson method. The algebraic expression we are to solve using this method takes the form:

$$\begin{aligned} \mathcal{N}_i^{n+1}(1 + \tilde{\kappa}_i(\mathcal{N}^{n+1})\Delta t) - \Delta t F_i^+(\mathcal{N}^{n+1}) - \mathcal{N}_i^n &= 0 \\ f(\mathcal{N}_i^{n+1}) &= 0 \end{aligned} \quad (3.15)$$

Where the flux and rate parameter are emphasized to be functions of $\mathcal{N}^{n+1}$, the $n + 1$ densities for all energy bins. The Newton-Raphson method will take the form:

$$\mathcal{N}_{i,k+1}^{n+1} = \mathcal{N}_{i,k}^{n+1} - \frac{f(\mathcal{N}_{i,k}^{n+1})}{f'(\mathcal{N}_{i,k}^{n+1})} \quad (3.16)$$

With the terminating condition that:

$$\frac{|f(\mathcal{N}_{i,k}^{n+1})/f'(\mathcal{N}_{i,k}^{n+1})|}{\mathcal{N}_{i,k+1}^{n+1}} < tolBE \quad (3.17)$$

Here, k is an integer designating the iteration step³ and $f'(\mathcal{N}_{i,k}^{n+1})$ is the derivative of $f(\mathcal{N}_{i,k}^{n+1})$ with respect to $\mathcal{N}_{i,k}^{n+1}$. For the first guess using Newton-Raphson method, we use the last particle number density that the algorithm calculated when FE was still running. This was demonstrated for a single equation in which you would be dealing in scalars. The corresponding vector form Eq.(3.14) would require a Newton-Raphson calculation of the form:

$$\mathcal{N}_{k+1}^{n+1} = \mathcal{N}_k^{n+1} - \frac{f(\mathcal{N}_k^{n+1})}{f'(\mathcal{N}_k^{n+1})} \quad (3.18)$$

In which the Newton-Raphson calculation will require matrix inversions; recall from Eq.(2.19) that $\mathbf{C}(\mathcal{N}^{n+1})$ is a product of a our square "Collision Matrix" $\mathbf{M}(\mathcal{N}^{n+1})$ and $\mathcal{N}^{n+1}$. We keep track of the number of iterations required to solve the problem with a parameter called "nTrueIteration" which is analogous to truecycle for EA. Each time step requires a certain number of iterations to satisfy the condition in Eq.(3.16); nTrueIteration is the sum of all the iterations over all time steps. We can see BE's behavior in the high density case looking at Fig.(3.13) - (3.16) . Here we choose the loosest tolerance for the relative errors and show different tolerances for particle conservation. For the relative errors, we chose the same energy bins as with EA. This is to illustrate the point that BE using intermediate and large time stepping, or a dt_{grow} of 1.03 and 1.20, still produces acceptable results, even for larger tolerances, and thus stricter tolerances will as well. The relative error for BE for

³Not to be confused with k which designated energy bins in the system earlier

the aggressive time stepping Fig.(3.14) maxes out at approximately 5% which mirrors this scenario for EA. Both time stepping cases conserve particle number to approximately 10^{-5} though BE actually underestimates the next iteration step in each case, explaining why the plots are negative in Fig.(3.15) and Fig.(3.16).

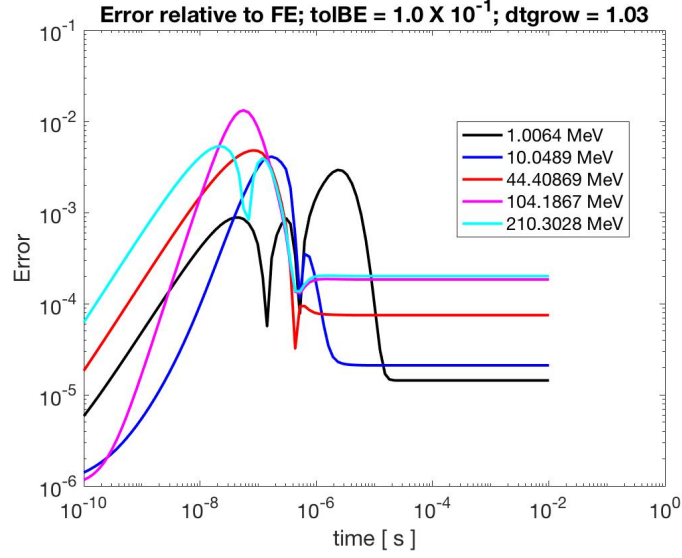


Figure 3.13: Relative Error for BE using intermediate time stepping.

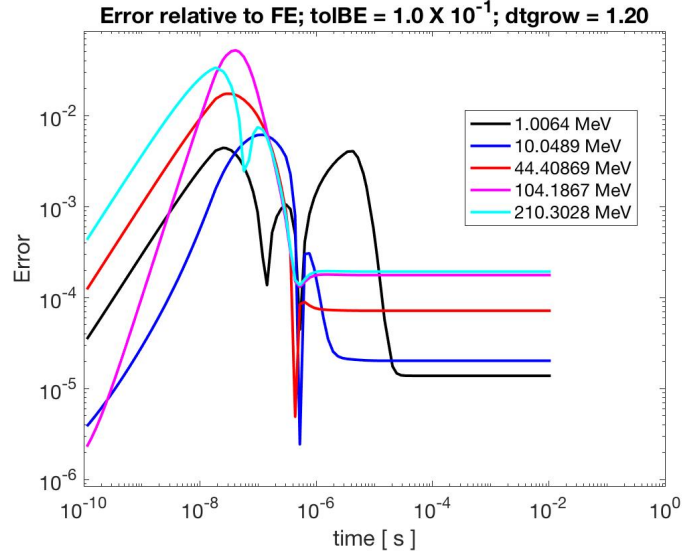


Figure 3.14: Relative Error for BE using large time stepping

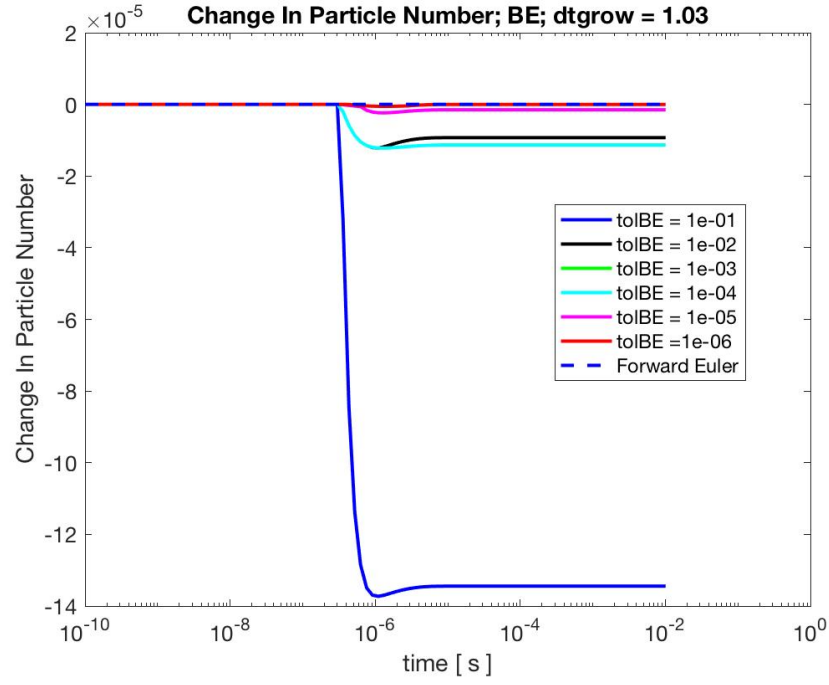


Figure 3.15: Change in particle number for intermediate stepping; BE.

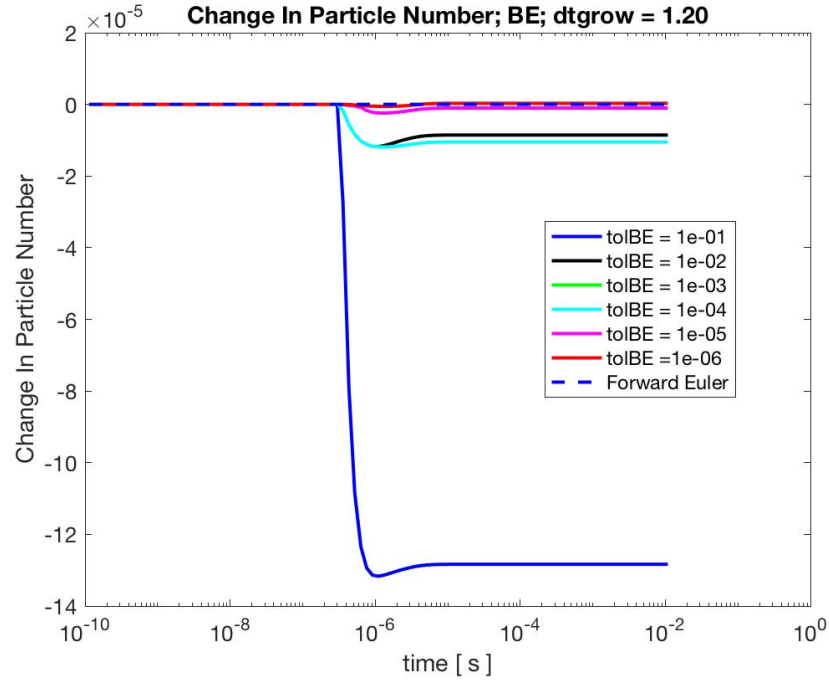


Figure 3.16: Change in particle number for large time stepping; BE.

3.2.2 Fixed Point Iteration

An alternative implicit approach that will be used for comparison is the Fixed Point (FP) iteration method. It involves finding $\mathcal{N}_i^{n+1}$ such that:

$$G(\mathcal{N}_i^{n+1}) = \mathcal{N}_i^{n+1} \quad (3.19)$$

Where $G(\mathcal{N}_i^{n+1})$ is a function we will have to specify. The idea is to let:

$$G(x) = x - f(x)$$

For some function f . If we let $x = \mathcal{N}_i^{n+1}$ the 2nd term vanishes because $\mathcal{N}_i^{n+1}$ is a root of f and we are left with a fixed point problem. If we are able to solve the fixed point problem given by Eq.(3.19), then we know that same $\mathcal{N}_i^{n+1}$ will be the solution to Eq.(3.14). If we look at Eq.(3.14), we can set up a fixed point problem by isolating $\mathcal{N}_i^{n+1}$ which yields:

$$\mathcal{N}_i^{n+1} = \frac{\mathcal{N}_i^n + \Delta t F_i^+(\mathcal{N}^{n+1})}{1 + \Delta t \tilde{\kappa}_i(\mathcal{N}^{n+1})} = G_i(\mathcal{N}_i^{n+1}) \quad (3.20)$$

Where again we've emphasized that the flux and rate parameter are functions of $\mathcal{N}^{n+1}$ and $G(\mathcal{N}_i^{n+1})$ is the function we iterate using the gaussian initial condition as our initial guess. The iteration relation is given by:

$$\mathcal{N}_{i,k+1}^{n+1} = G(\mathcal{N}_{i,k}^{n+1}) \quad (3.21)$$

where the tolerance for the convergence condition is defined as $tolFP$ such that:

$$\|\mathcal{N}_{i,k+1}^{n+1} - \mathcal{N}_{i,k}^{n+1}\| \leq tolFP \|\mathcal{N}\| \quad (3.22)$$

where $\mathcal{N}$ is the number density given by the last FE calculation before the stability condition was exceeded. Much like with BE, FP conserves particle number only to a given tolerance. One thing to note is the connection between FP and EA. Essentially, from looking at Eq.(3.20) and comparing that to the derivation of EA ending in Eq.(3.6), EA is a single fixed point iteration. Fixed point is an implicit method which does not require matrix

inversions, but may require multiple matrix-vector per time step multiplies to satisfy its tolerance condition as opposed to EA which only requires one. We also include the same methodology as described for BE to count the total number of iterations required to solve the problem. Also much like with BE, in Fig.(3.17) - (3.20) we show the relative errors for the loosest tolerance of FP (using the same energy bins) and the particle number conservations for different tolFP to illustrate that the method produces acceptable results even with a lax tolerance condition. Much like with BE, the aggressive stepping case produces a relative error that maxes out at approximately 5% while each case conserves particle number to $\sim 10^{-4}$ even for the loosest tolerance, though FP overestimates with each iteration as opposed to underestimates.

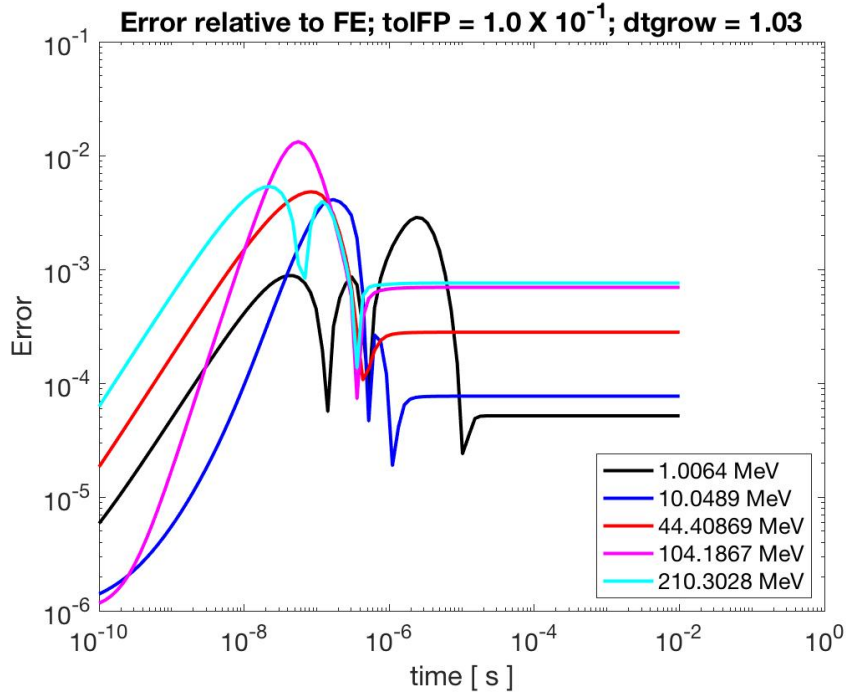


Figure 3.17: Relative Error for FP using intermediate time stepping.

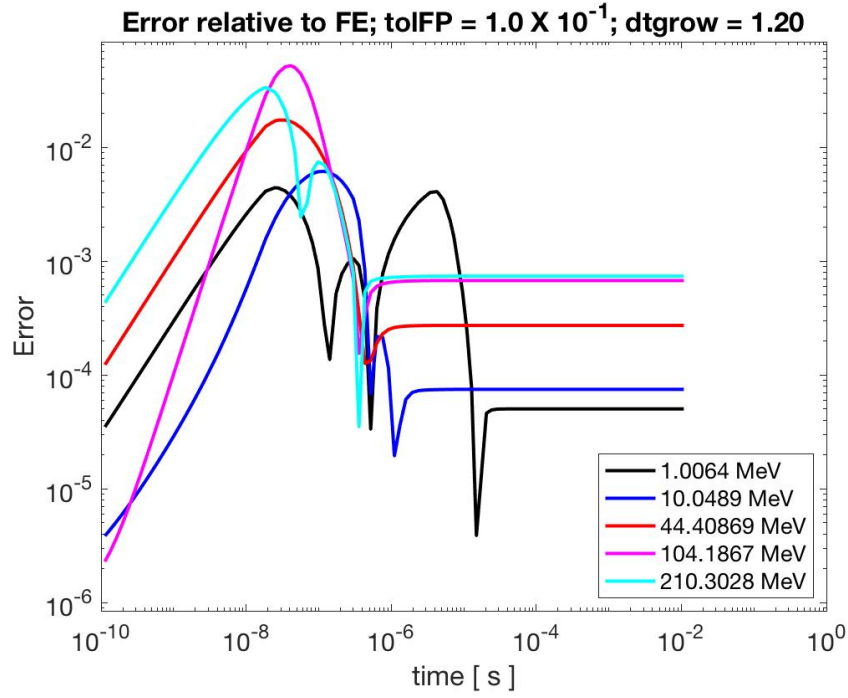


Figure 3.18: Relative Error for FP using aggressive time stepping.

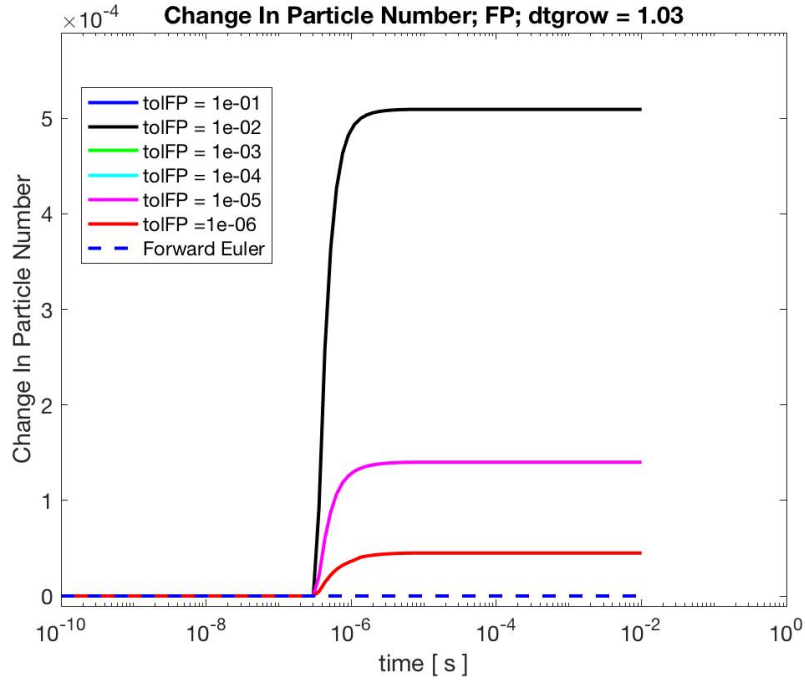


Figure 3.19: Change in particle number for intermediate time stepping; FP.

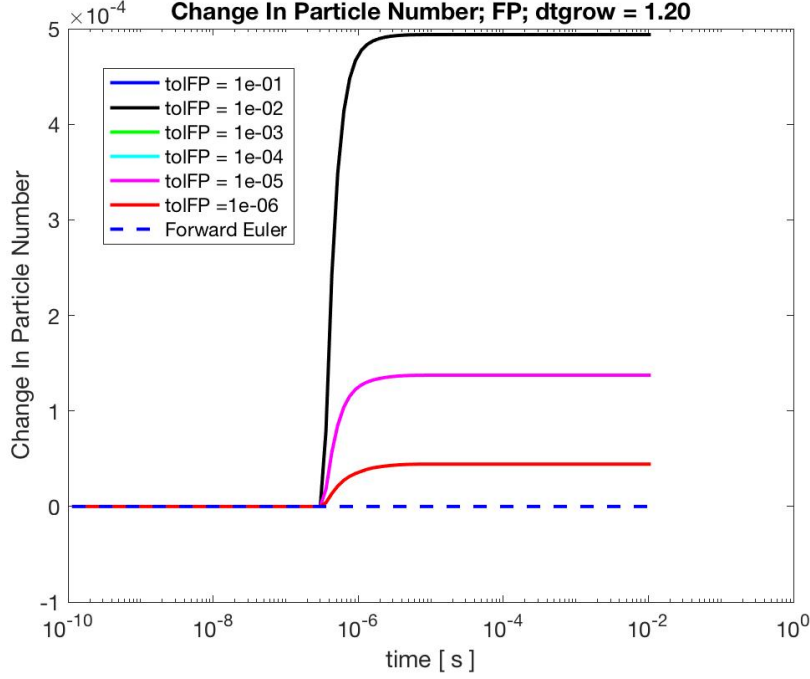


Figure 3.20: Change in particle number for large time stepping; FP

3.2.3 Anderson Accelerated Fixed Point Iteration

As described by Toth and Kelley [12], the Accelerated Fixed Point (AFP) method is designed to help FP converge faster by storing previous evaluations of the FP method and then summing the result to obtain the next iteration. As with FP, we use our Gaussian initial condition as our initial guess. The algorithm detailed by Toth and Kelley utilizes residuals which take the form:

$$f(\mathcal{N}_{i,k}^{n+1}) = G(\mathcal{N}_{i,k}^{n+1}) - \mathcal{N}_{i,k}^{n+1} \quad (3.23)$$

And defines a parameter $m_k = \min(m, k)$, for integers m and k , in which we want to find a linear combination that minimizes the residuals, i.e:

$$\left\| \sum_{j=0}^{m_k} \alpha_j^k f_{k-m_k+j} \right\| \quad (3.24)$$

where $\sum_{j=0}^{m_k} \alpha_j^k = 1$. The next particle density in the iteration is then given by:

$$\mathcal{N}_{i,k+1}^{n+1} = \sum_{j=0}^{m_k} \alpha_j^m G(\mathcal{N}_{i,k-m_k+j}^{n+1}) \quad (3.25)$$

For our calculations, $m = 3$ ⁴. We use the same tolerance condition as in the FP case with the parameter denoted as tolAFP. Justification for this choice of m is illustrated in Fig.(3.21) which shows the number of iterations required to satisfy the tolerance condition at each time step. This plot reflects calculations in which only AFP was run (it's not augmented with FE). Using a $dt_{grow} = 1.10$, it is found that there is no significant improvement when m (denoted mAA in the plot) is larger than 3, thus we use 3 as the default. In Fig.(3.22) - (3.25) again we show plots for relative error for the same energy bins as BE and FP for a loose tolerance and particle number conservation for different tolAFP. For these specific parameters there is negligible difference between FP and AFP, so the method produces similar and acceptable results. We will see the benefits of AFP in Chapter 4.

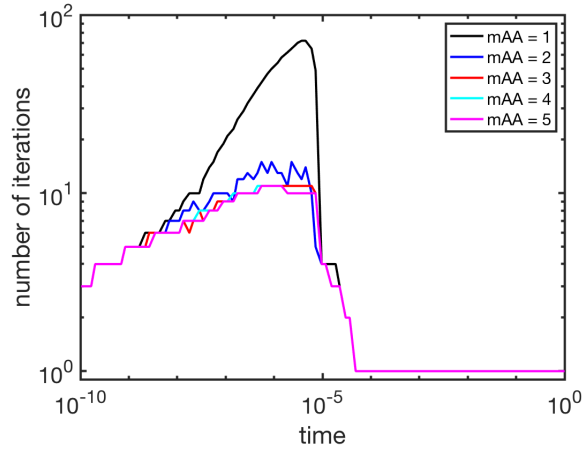


Figure 3.21: Number of iterations AFP takes for a given m .

⁴In the Toth and Kelley paper there is also a parameter β_k which we set to be equal to 1.

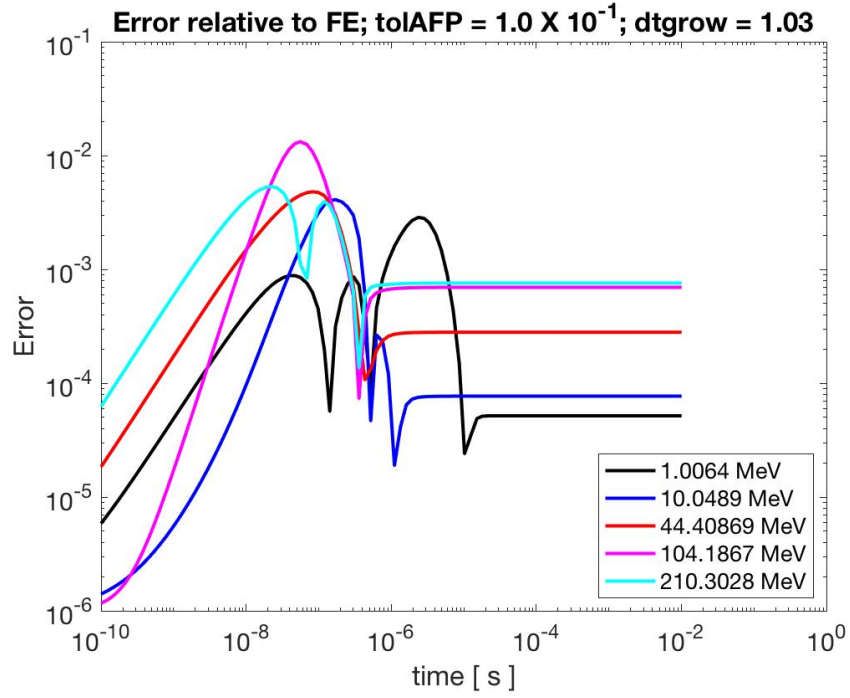


Figure 3.22: Relative Error for AFP using intermediate time stepping.

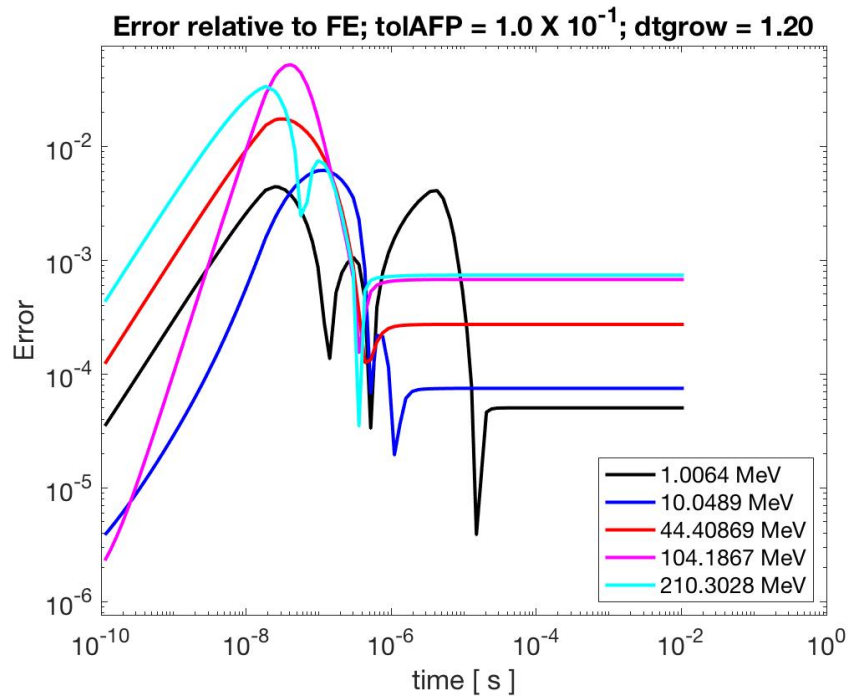


Figure 3.23: Relative Error for AFP using aggressive time stepping.

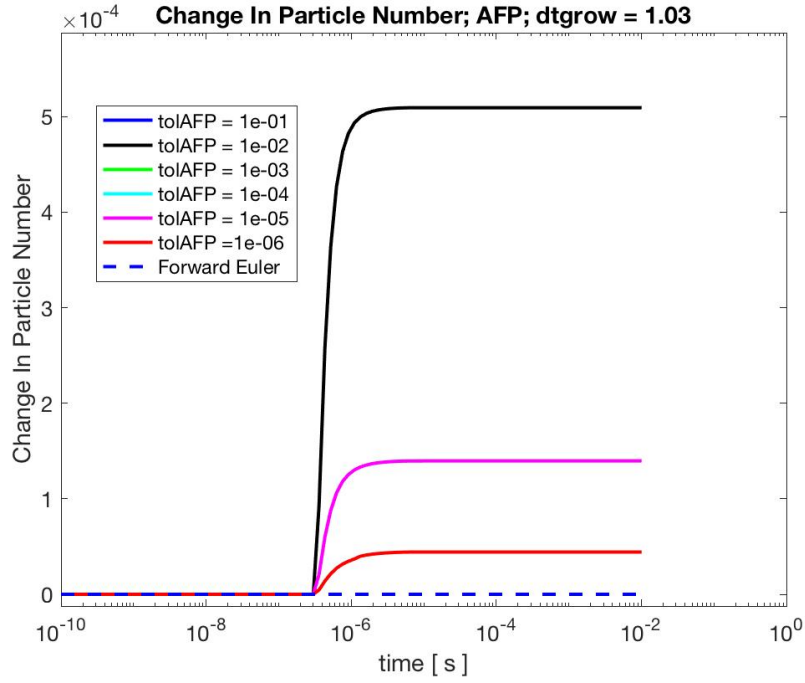


Figure 3.24: Change in particle number for intermediate time stepping; AFP

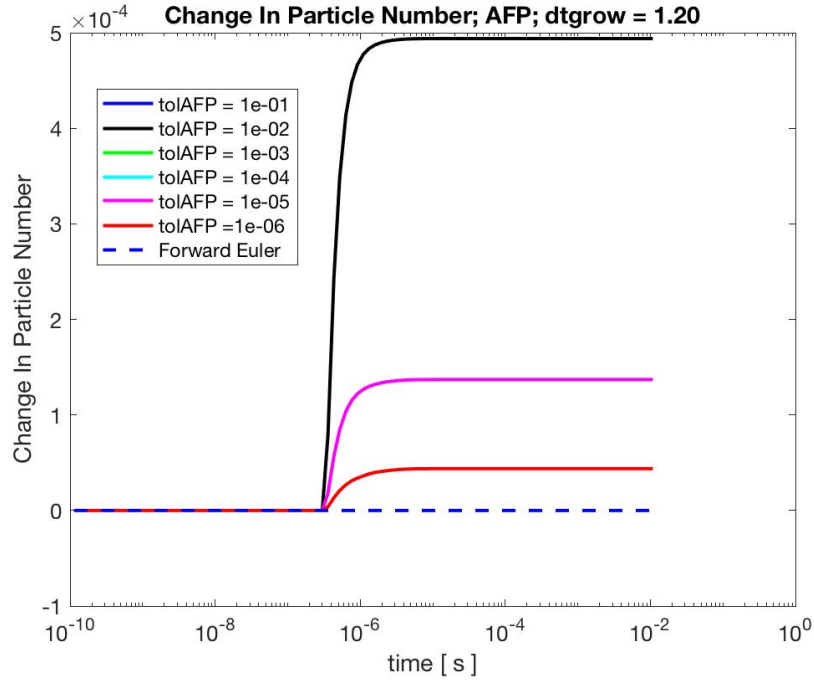


Figure 3.25: Change in particle number for large time stepping; AFP.

Chapter 4

Numerical Comparisons

This chapter will feature error, time stepping, and speed comparisons between the different methods. Also included are figures detailing the particle number conservation for each method. Here we show that for certain parameters the EA method is competitive with standard implicit methods.

4.1 Error And Time Stepping Comparisons

First and foremost, let's look at how EA compares to the implicit methods as it pertains to accuracy. For these comparisons, a $tolC = 1.0 \times 10^{-4}$ is used for EA as the standard that the other methods will be compared to. This tolerance yields acceptable particle number conservation $\sim 10^{-2}$ or smaller as demonstrated in Fig.(3.9) - (3.11), and it provides an opportunity to test the method even when the time stepping isn't ideal. EA may be competitive with implicit calculations even if the EA algorithm has to redo time steps due to violation of our particle conservation condition. However, in order to ascertain how the method performs as far as speed, we must be sure that our comparisons are fair. In order to do this, a comparison between the error in number densities relative to FE is necessary to ensure that we are comparing calculations that produce relatively the same results. We take the maximum error in density among all energy groups at each moment in time, given by:

$$\text{Max Error} = \max(\max(dN), \max[\frac{|\mathcal{N}_{FE} - \mathcal{N}|}{\max(\mathcal{N}_{FE}, 10^{-8})}]) \quad (4.1)$$

Where $d\mathcal{N}$ is defined by Eq.(3.12) and we set a stipulation that the denominator in the 2nd argument is never smaller than 10^{-8} . As per usual, we consider dt_{grow} of 1.003, 1.03, and 1.20 for our small, intermediate, and aggressive time stepping. Fig.(4.1) - (4.3) show the maximum errors versus time comparing implicit calculations to our standard EA calculation in the high density regime.

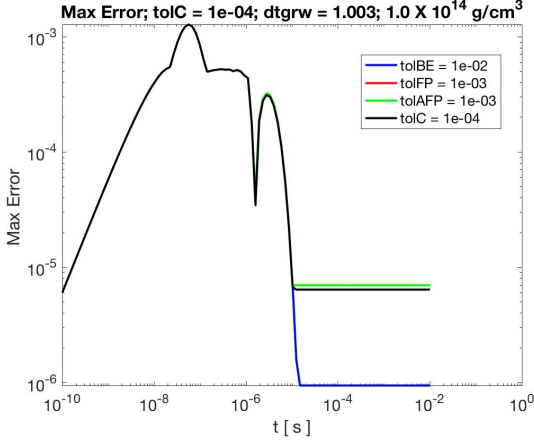


Figure 4.1: Max error from FE versus time compared to the error for our standard EA calculation for small dt_{grow} .

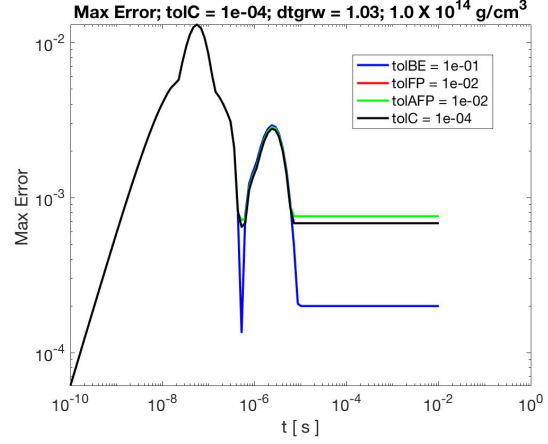


Figure 4.2: Max error from FE for a larger dt_{grow} . Here the EA, FP, and AFP are more competitive with BE.

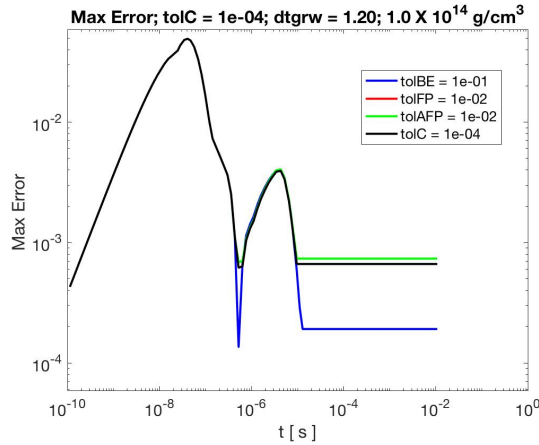


Figure 4.3: Max error from FE for an aggressive dt_{grow} .

The tolerance parameters for the implicit methods shown generate the closest error to EA. We see that all of the plots overlap with each other at the beginning of the run. This is due to all of the algorithms running FE until the time steps become so large that they exceed the stability condition; after which they diverge from each other at around 10^{-5} seconds. This similar behavior is due to the tolerance condition in Eq.(3.11) which limits the offset in number density between successive iterations; a tolerance condition that all of the methods have in common. Recall that the FE run that is used as an accuracy test uses a dt_{grow} of 1.0001; much smaller than the growth factors used for the methods we're comparing. This is the source of the error from FE in the region where all of the curves overlap. For the intermediate and aggressive time stepping, this error peaks around 5% and 8% respectively. Furthermore, BE noticeably outperforms EA, FP, and AFP as far as error for all time stepping scenarios, a behavior that we will see becomes less pronounced for other density cases. Having said this the errors are roughly still within the same order magnitude. This indicates that BE is a particularly robust method in this density case as increasing tolBE to values larger than what's given in the plot has negligible effect on the error. EA's error is due to both time stepping and the method not conserving particle number. The implicit methods conserve to their tolerances so taking such large steps compared to FE is the source of their error coupled with the tolerance conditions being lax. Furthermore, there is no discernible difference between FP and AFP at this point because the residuals that AFP uses are small for a dense model when NES isn't as diffuse throughout the star.

Fig.(4.4) shows the time stepping and particle conservations using the parameters found in the maximum error plots. All of the methods conserve particle number well beneath our accepted level of $\sim 10^{-2}$. Furthermore, all of the methods have virtually identical time stepping, as evident in the curves overlapping. This is a result of the methods having to adhere to the tolerance condition that limits the error in density between iterations. As the time stepping gets larger, this condition forces the algorithm to re-step more often (because the error gets larger), and thus the curves become less linear. This also shows that for EA, this condition dominates over the the particle number conservation condition detailed in Eq.(3.10) for these parameters. More importantly, the time stepping being identical helps

ensure that we are making as fair a comparison as possible once we examine the speed of these methods.

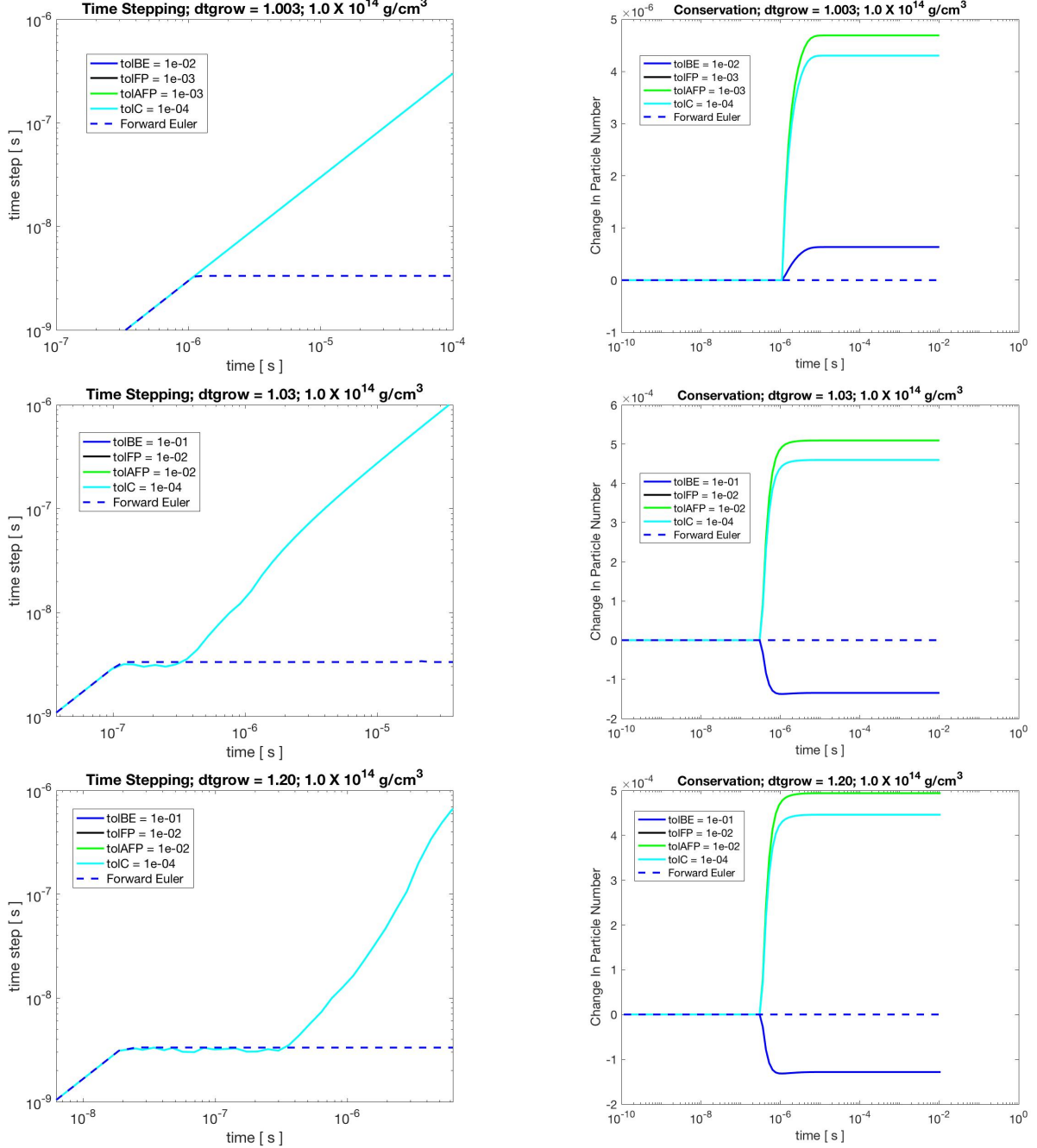


Figure 4.4: Time stepping and particle conservation for small, intermediate, and aggressive time stepping in the high density case. The time stepping for all methods are virtually identical and the methods all conserve particle number to an acceptable level.

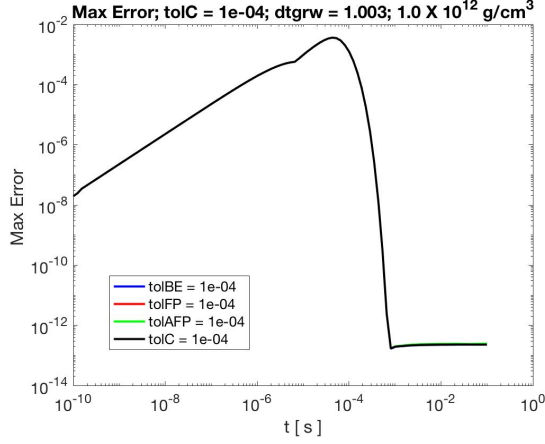


Figure 4.5: Max error from FE for a small dt_{grow} in the intermediate density case.

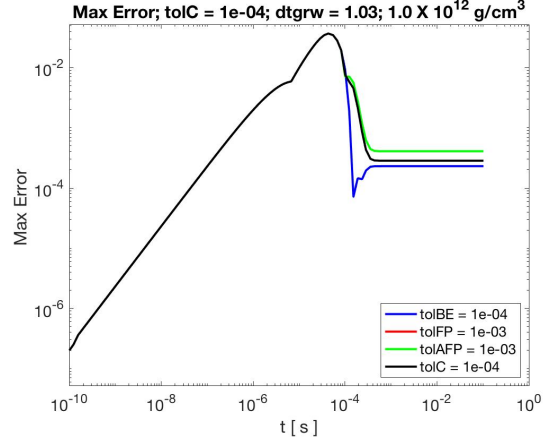


Figure 4.6: Max error from FE for a larger dt_{grow} in the intermediate density case.

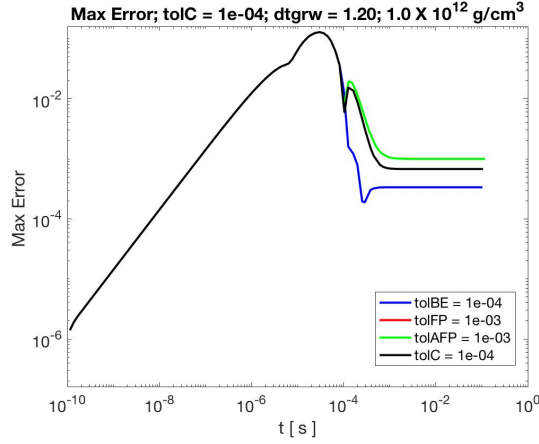


Figure 4.7: Max error from FE for an aggressive dt_{grow} in the intermediate density case.

Fig.(4.5)-(4.7) show the max error plots in the intermediate density scenario. From these figures we see that all of the methods overlap in the small time step scenario, which wasn't the case in the high density regime. This occurs because the stability condition is large enough for this model that it always runs FE for this growth factor. The implication of this is that the equations are not as stiff in this regime. The spread in error also isn't as large in the time stepping cases in comparison to the high density regime. Scattering is more diffuse

in this case than in the high density regime, so the difference in error output between the methods becomes less noticeable.

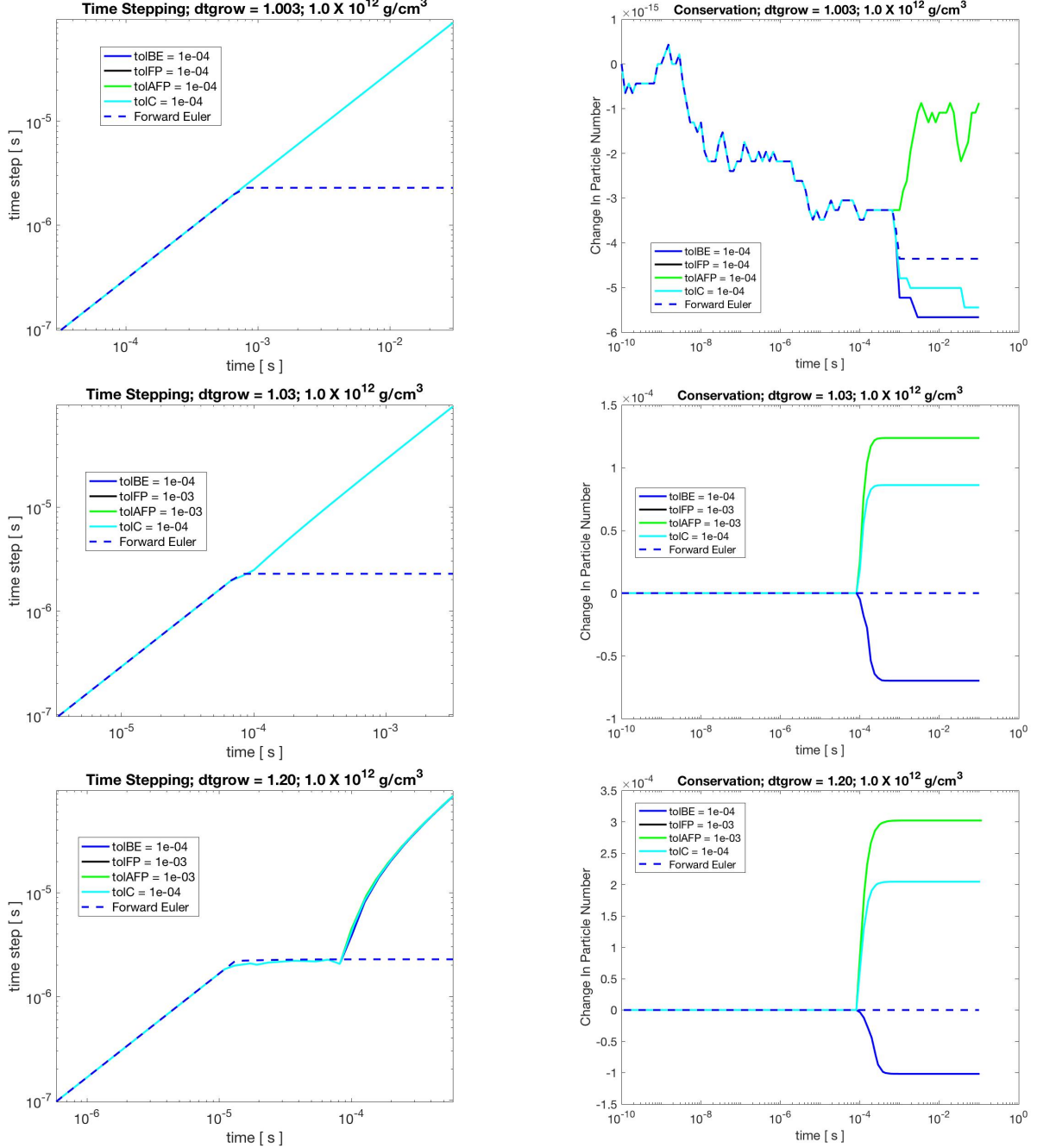


Figure 4.8: Time stepping and particle conservation for small, intermediate, and aggressive time stepping in the intermediate density case. The time stepping for all methods are virtually identical and the methods all conserve particle number to an acceptable level.

Again, in the intermediate and aggressive case this error maximizes at approximately 5% and 8% respectively. Fig.(4.8) shows the time stepping and particle conservation for these conditions. The conservation in the the small time stepping scenario is simply machine error precision as a result of FE always running. In the other cases we still have the same behavior in that the time stepping is virtually identical and the methods all conserve particle number to an acceptable level. As a result of the equations being less stiff, the algorithm doesn't have to re-step as often due to the amount of time each method spends running FE. Next Fig.(4.9)-(4.11) show the maximum errors in the low density regime. In the low density regime, collisions are more sparse further out than in the higher densities. and thus the error in calculating the particle number densities for the methods will be larger as a result. In conjunction to this, the implicit tolerances which yield error most similar to EA are closer in proximity than in the other cases. The general trend being that for a fixed tolC, the spread in error between EA and the implicit methods will be smaller, though the tolerance parameters which yield the closest error to EA may be difference in each case. The area where the curves overlap in the intermediate and aggressive time stepping cases always maximizes at approximately 5% and 8% respectively, which indicates that this a result of the $dt_{grow} = 1.03$ and 1.20 being much larger than in the FE calculations we use as a standard for accuracy. All of the methods have these choices for intermediate and aggressive time stepping in common, so all of them exhibit this behavior.

Lastly, Fig.(4.12) shows the time stepping and particle number conservation in the low density case. As a result of the condition to limit the offset in number density between iterations, the methods have to re-step more often due to the neutrino-electron interactions being more diffuse, thus producing more relative error. This is exacerbated when taking large time steps as indicated in the aggressive time stepping case. Having said that, we still have a situation where all of the time stepping curves are virtually identical so this along with the spread in error for this model shows that we can make fair comparisons regarding the speed of each method. Moreover, all of the methods conserve particle number to $\sim 10^{-5}$ which is well below our standard of acceptability. Noteworthy however is that there is noticeable separation between FP and AFP. For the most diffuse scattering we see that in

the intermediate and aggressive time stepping cases, AFP conserves particle number better than FP. This indicates that the residuals that AFP uses are large enough in this regime that the contributions of the method are more pronounced. Knowing all of this information, we can finally ascertain whether EA produces competitive results in the realm of speed.

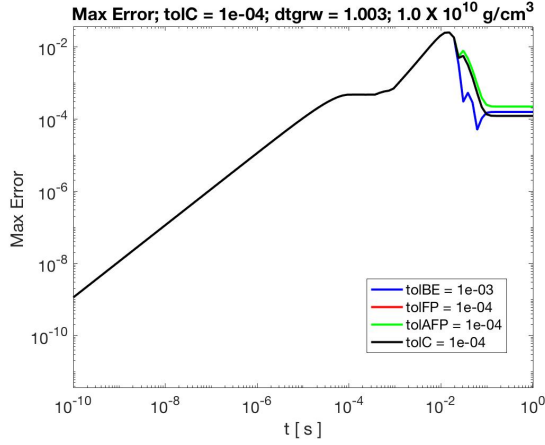


Figure 4.9: Max error from FE versus for the smallest dt_{grow} in the low density case.

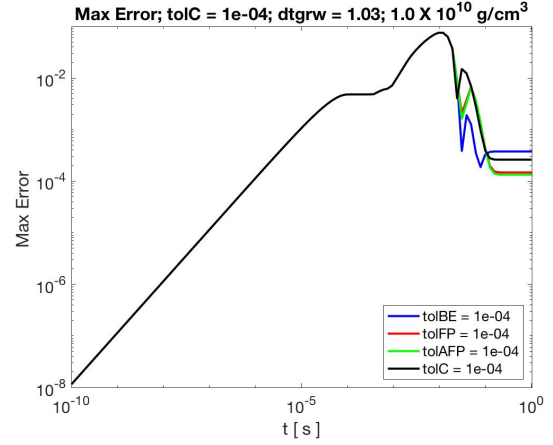


Figure 4.10: Max error from FE for a larger dt_{grow} in the low density case.

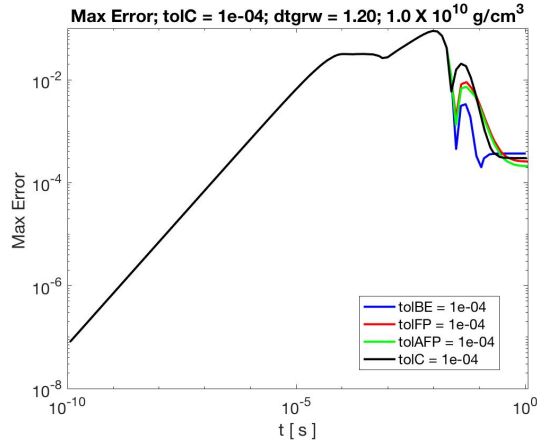


Figure 4.11: Max error from FE for the most aggressive dt_{grow} in the low density case.

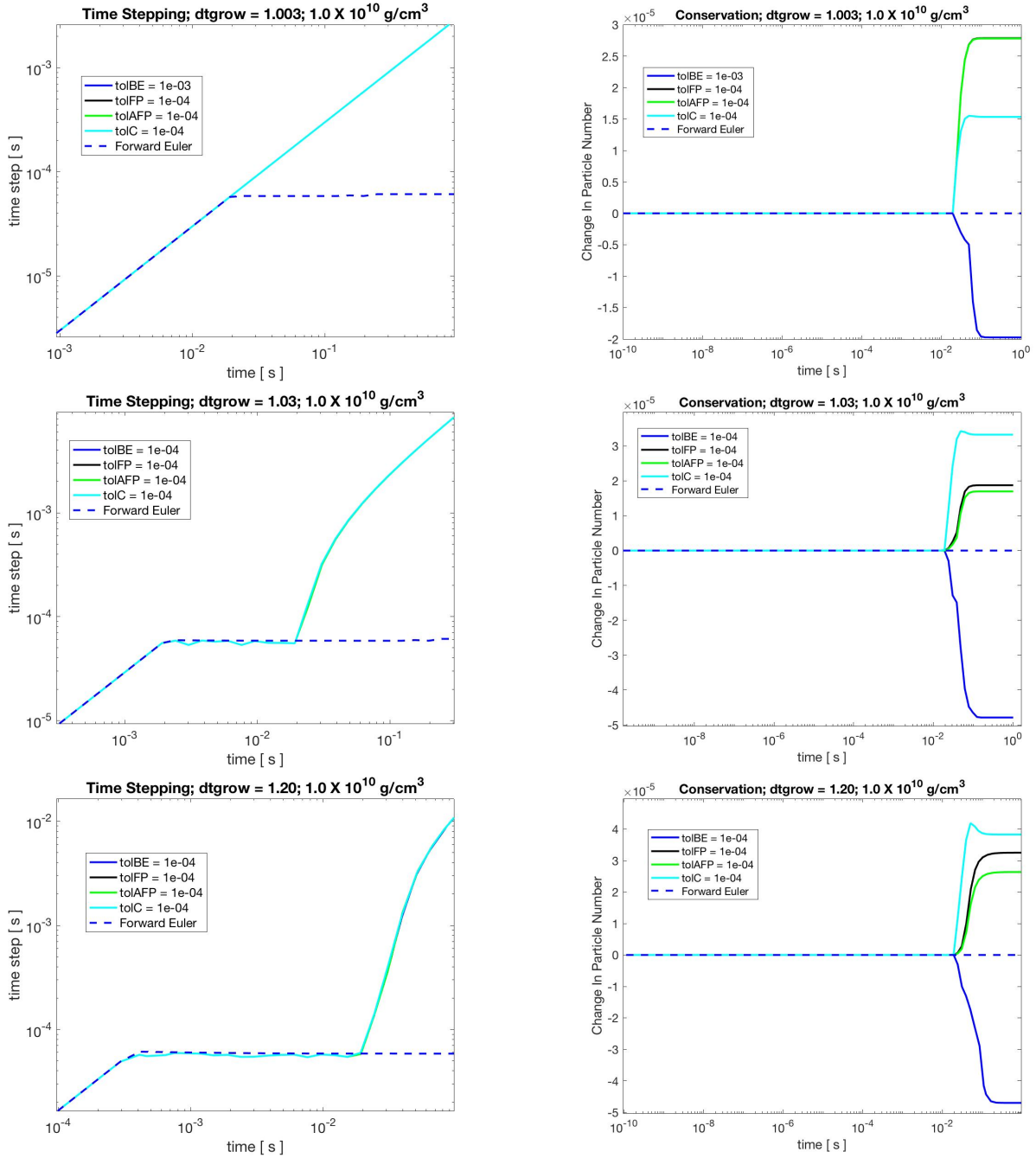


Figure 4.12: Time stepping and particle conservation for small, intermediate, and aggressive time stepping in the high density case. The time stepping for all methods are virtually identical and the methods all conserve particle number to an acceptable level.

4.2 Speed Comparisons

In the ideal case, some choice of parameters will allow EA to run faster than implicit methods. However it may not be necessary for EA to be faster, just "competitive". As stated earlier, we're comparing the number of iterations it takes to solve the equations using tolerance cases that yield similar error. In the last section we already detailed that these cases all yielded identical time stepping and conserved particle number well below our accepted standard. In order to gauge whether EA is competitive or not, we must do more than simply look at the number of iterations taken to solve the equations. There are also cost concerns related to matrix-vector multiplications in the case of EA, and matrix inversions in the case of the implicit methods like BE. This detail is not accounted for in counting the iterations, or righthand (RH) evaluations as they will later be called.

Recall that Chapter 2 Section 3, Fig.(2.1) shows the average time taken to do matrix-vector multiplications versus matrix inversions depending on the size of the matrix. We showed that, using dense matrices build using random number generations, for all matrix sizes it is more costly to do matrix inversions. In general the time it takes to do matrix inversions asymptotically scales cubically with the size of the matrix as opposed to quadratically for matrix-vector multiplication.

Fig.(4.13) shows a plot comparing "truecycle" vs. time for EA , and "nTrueIterations" vs. time for the implicit methods for the $dt_{grow} = 1.003$ case in the high density scenario. Both of these parameters as mentioned earlier are describing the same thing, the number of iterations (or RH evaluations) that each method is taking in solving the problem. Recall that each algorithm is designed to output 100 data values for the number densities throughout the duration of the run. The points on Fig.(4.13) represent the number of iterations required to produce the density calculation that was output at the corresponding time. We see that for small time stepping, all of the methods solve the problem in roughly the same number of iterations, hence the overlap with each method taking approximately 8000 iterations. But because of the detail referenced early about matrix-vector multiplications versus matrix

inversions, EA in this scenario is less costly than the BE method because it only does a single matrix-vector multiplication per iteration. Of course the smaller time stepping means the algorithms take longer to solve the problem and we want to take large time steps. Fig(4.12) shows the number of right hand evaluations for the 1.03 growth factor case. Again, just like in the previous scenario, EA solves this problem in the same number of iterative steps as the implicit methods, taking on the order of 1000 iterations. And much like with the least aggressive case, EA is less costly than BE which is an added bonus. FP and AFP mirroring EA indicates that each method is taking a single iteration per time step to satisfy their tolerance conditions (recall that EA is a single FP iteration). This also addresses why the errors almost overlap for these methods. In Fig.(4.13) we finally start to see some discernible offset, the most aggressive time stepping. Initially the plots of course overlap due the methods all running FE, and then each take a sharp jump around 10^{-8} seconds when the stability is reached and the number of iterations increases to account for the condition to keep the error in $\mathcal{N}$ small. In this regime we see that EA actually runs slightly faster than the implicit methods, taking 767 RH evaluations compared to 812 for BE and 837 for FP and AFP. Hence, EA is 6% faster than BE even without accounting for the cost due to matrix operations, and 9% faster than FP and AFP as a result of there needing to be multiple iterations to satisfy their tolerance conditions. As a result, FP and AFP having to do multiple matrix-vector multiplications makes them more costly than EA. In this model, it is more beneficial to use larger time steps because not only will you solve the problem faster than if you used a smaller dt_{grow} , EA is not only competitive with implicit methods, it slightly outperforms them. Much like with the error and particle conservation plots in Chapter 4 Section 1, the effects of AFP are not yet noticeable for this model, for these parameters.

The intermediate density cases are shown in Fig.(4.16) - (4.18). Recall that all of the methods for the smallest time stepping run FE throughout so these results are not inherently interesting outside of highlighting the detail that the equations are not as stiff in this scenario. The intermediate time stepping case however shows that BE is slightly slower than EA, FP and AFP taking 1047 iterations to solve the equations as opposed to 981 for EA and 982 for

FP and AFP. The detail of most importance is that EA performs well for this time stepping. This also reiterates the observation we detailed in the last section that BE is less robust compared to the other methods when scattering is more diffuse inside the star, which is the case for lower density models.

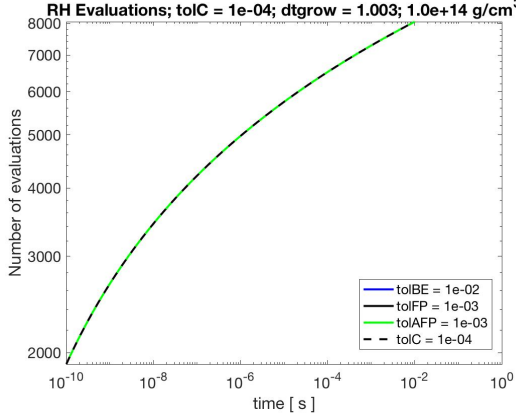


Figure 4.13: Number of right hand evaluations for the least aggressive time stepping in the high density scenario.

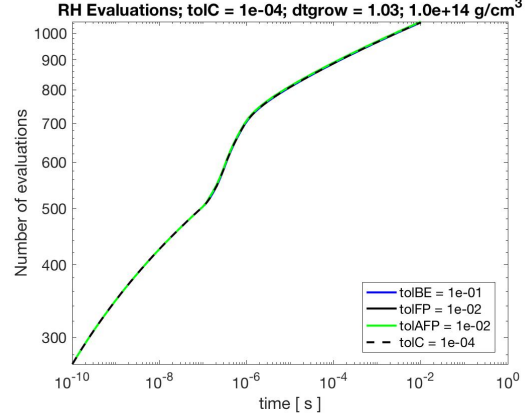


Figure 4.14: Number of right hand evaluations for a growth factor of 1.03 in the high density case.

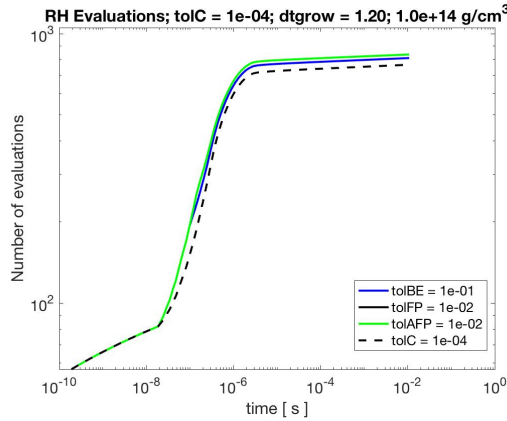


Figure 4.15: Number of right hand evaluations for the most aggressive time stepping. Here the EA method is not competitive with the implicit methods.

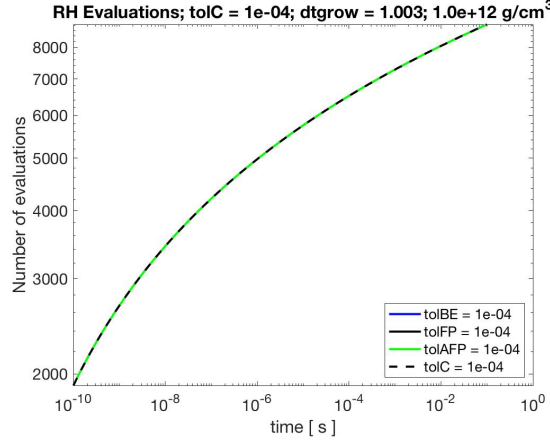


Figure 4.16: Number of right hand evaluations for the least aggressive time stepping in the intermediate density case.

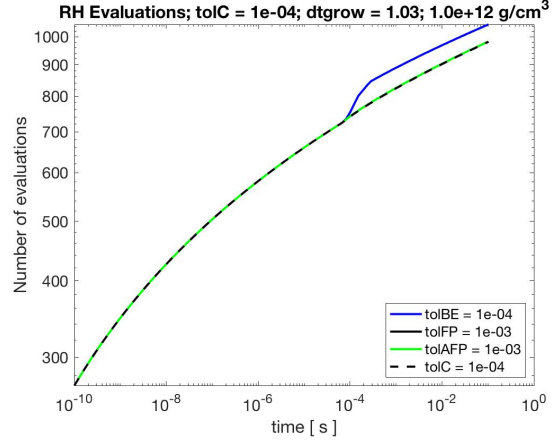


Figure 4.17: Number of right hand evaluations for a larger time step in the intermediate density case.

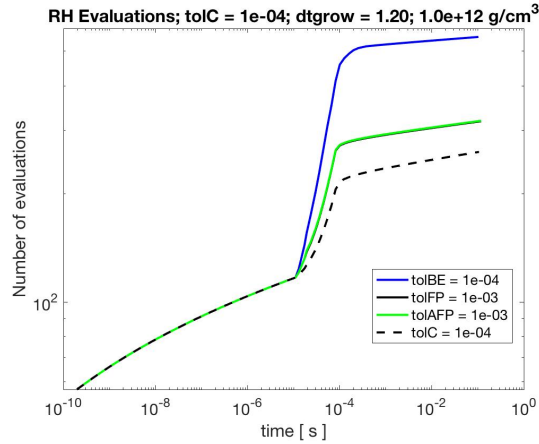


Figure 4.18: Number of right hand evaluations for the most aggressive time stepping in the intermediate density case.

For the most aggressive time stepping, we see that EA runs in fewer iterations than the implicit methods taking 262 RH evaluations to solve the equations compared to 319 for FP and AFP and 548 for BE. Not accounting for the cost due to matrix operations, this means EA runs 18% faster than FP and AFP, and 61% faster than BE. Much like with the high density model, EA out performs the implicit methods in number of iterations on top of not having the cost concerns associated with the linear algebra. Moreover, the degree to which

EA is faster in larger in this scenario. We can gain further insight into this by examining the RH evaluations for the low density model when the scattering is more diffuse. This is illustrated in Figs.(4.19) - (4.21).

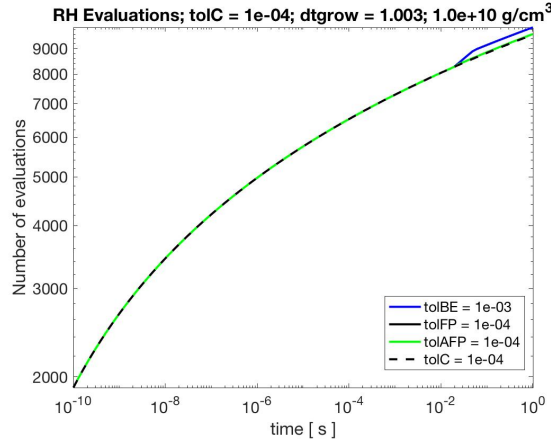


Figure 4.19: Number of right hand evaluations for the least aggressive time stepping for the low density scenario.

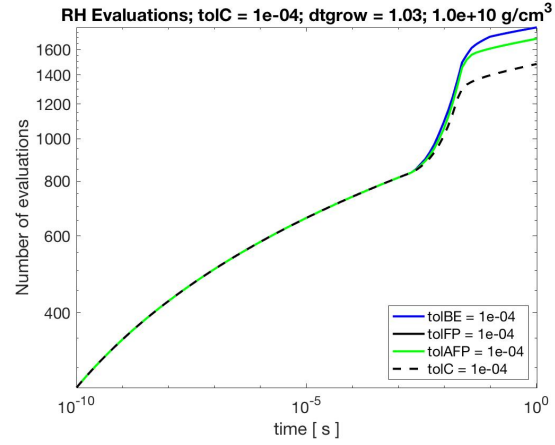


Figure 4.20: Number of right hand evaluations for larger time stepping in the low density scenario.

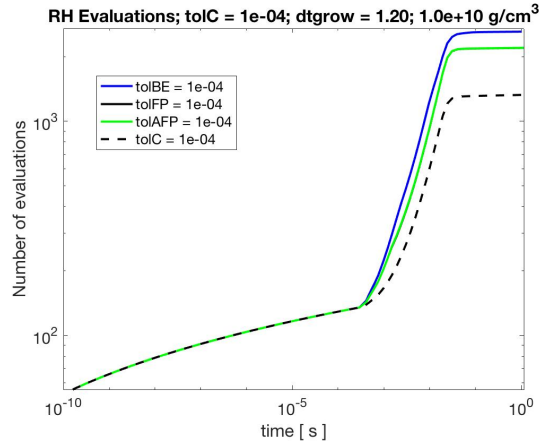


Figure 4.21: Number of right hand evaluations for the most aggressive time stepping in the low density scenario.

Again the smallest time stepping case produces a similar number of RH evaluations for all methods. In the intermediate case, there is more spread in the number of RH evaluations for each method in that EA runs faster than FP and AFP (along with BE) when in the intermediate density case these were approximately the same. In this scenario EA runs in 1482 iterations compared to 1693 for FP and AFP and 1797 for BE, a 12% difference and 17% difference respectively. Likewise for the most aggressive time stepping EA runs in 1327 iterations compared to 2200 for FP and AFP and 2620 for BE, a 40% difference and 49% difference respectively. Much like in the previous density models, the degree to which EA is faster than implicit methods becomes more pronounced with larger time stepping.

One brief addendum is that the tolerance parameter for implicit methods utilized in this thesis were only chosen because they yielded similar error to EA. One who is more comfortable using implicit methods may not choose these parameters because of their lax tolerance conditions. Fig.(4.22) and Fig.(4.23) show the time stepping and RH evaluations in the high density case using tolerance parameters that are more strict. Say on the order of 10^{-8} . Here we choose the intermediate time stepping arbitrarily. The point is to illustrate that the time stepping is still identical in this regime and that EA still outpaces the implicit methods. Fig.(4.24) shows the particle number conservation for these tolerances. So while the implicit methods significantly conserve particle number better than EA, it costs you more iterations to do so and EA already conserves to our standard of acceptability.

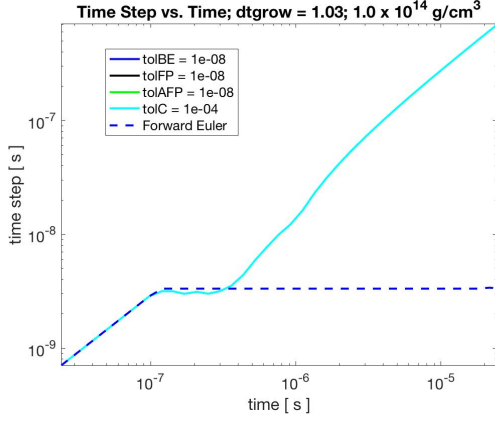


Figure 4.22: Time stepping in high density case using stricter tolerances for implicit methods.

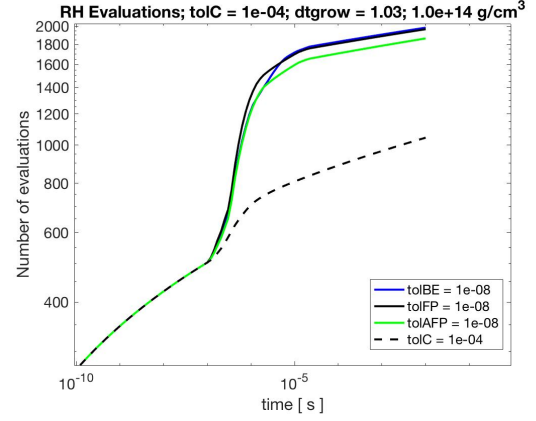


Figure 4.23: Number of right hand evaluations in the high density case using strict tolerances for implicit methods.

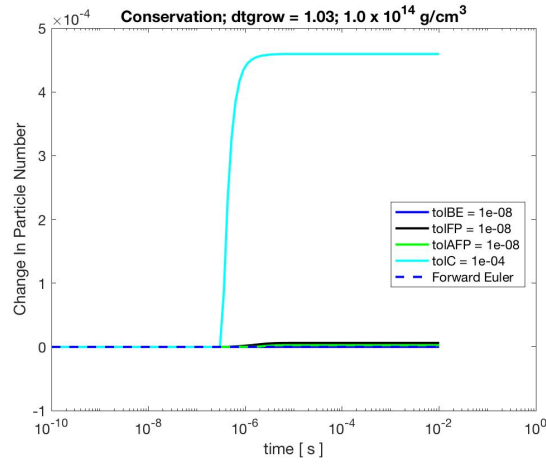


Figure 4.24: Particle conservation in the high density case using stricter tolerances for implicit methods.

Chapter 5

Conclusions

To summarize, the Explicit Asymptotic Approximation described by Guidry is shown to be able to solve the problem of neutrino-electron scattering as fast or faster than the standard implicit methods of Backward Euler, Fixed Point Iteration, and Anderson Accelerated Fixed Point, with the parameters presented in this thesis. In conjunction, because of the cost associated with doing matrix inversions for implicit methods like BE, and the possibility of doing multiple matrix-vector multiplications per time step in the case of FP and AFP, EA can serve as an alternative approach to solving stiff systems of equations represented by a matrix which is dense in the NES problem, or sparse in the context of thermonuclear networks highlighted by Guidry [5]. This is an essential detail because in a "realistic" calculation that is broader in scope than the ones presented in this thesis, the sizes of the system will be much larger and thus the cost associated with these matrix operations will be greater. These results may serve as groundwork to solving the neutrino transport problem more generally in which all neutrino-matter interactions are taken into account. However, with these results we cannot yet recommend a certain choice of parameters as a starting point for EA when applied to the neutrino transport problem more generally. More work will have to be done in the future to see if the parameters used in this thesis are universal. However in the context of isotropic NES, a $dt_{grow} = 1.20$ is sufficient as the time stepping is identical for all methods and the EA method solves the problem in the fewest number of iterations compared to the other time stepping cases, while also being faster than all of the implicit methods. Furthermore it is sufficient to use a $tolC = 10^{-4}$ with a $tol\mathcal{N} = 0.01$. These parameters will

yield acceptable particle number conservation and error. We also need to test the limits of EA and gauge how large a time step we can take while still maintaining results which are competitive and have sufficiently small error. Furthermore, in Guidry [5] they also introduce a method known as "Partial Equilibrium" which is used to deal with potential sources of stiffness near equilibrium. This could potentially improve the time stepping for an aggressive dt_{grow} in which there are more re-steps due to tolerance conditions. If such is the case, we may be able to augment EA with Partial Equilibrium in order to make a hybrid algorithm to optimize this method.

References

- [1] Bruenn, S. W. (1985). Numerical model and infall epoch. *Astrophysical Journal Supplement Series*, 58:771–841. [6](#), [17](#)
- [2] Bruenn, S. W. (2016). The development of explosions in axisymmetric ab initio core-collapse supernova simulations of 12-25 $M_{\odot}$ stars. *The Astrophysical Journal*, 818:123. [14](#)
- [3] Cernohorsky, J. (1994). Symmetries in neutrino–electron scattering. *Astrophysical Journal*, 433:247–249. [8](#)
- [4] Gear, C. W. (1971). Numerical initial value problems in ordinary differential equations. *Prentice Hall*. [1](#)
- [5] Guidry, M. (2012). Algebraic stabilization of explicit numerical integration for extremely stiff reaction networks. *Journal of Computational Physics*, 231:5266–5288. [v](#), [4](#), [12](#), [19](#), [61](#), [62](#)
- [6] Lambert, J. (1991). Numerical methods for ordinary differential equations. *Wiley*. [1](#)
- [7] M. Liebendorfer, O. E. B. M. and Mezzacappa, A. (2004). A finite difference representation of neutrino radiation hydrodynamics in spherically symmetric general relativistic spacetime. *The Astrophysical Journal Supplement Series*, 150:263–316. [viii](#), [13](#)
- [8] Mezzacappa, A. (2005). Ascertaining the core collapse supernova mechanism: The state of the art and the road ahead. *Annual Review of Nuclear and Particle Science*, 55:467–515. [4](#)
- [9] Oran, E. and Boris, J. (2005). Numerical simulation of reactive flow. *Cambridge University Press*. [1](#)
- [10] Smit, J. M. and Cernohorsky, J. (1996). Legendre expansion of the neutrino–electron scattering kernel. *Astronomy & Astrophysics*, 311:347–351. [7](#)

- [11] Thorne, K. S. (1981). Relativistic radiative transfer – moment formalisms. *MNRAS*, 194:439–473. [8](#)
- [12] Toth, A. and Kelley, C. (2015). Convergence analysis for anderson acceleration. *SIAM Journal on Numerical Analysis*, 53:805–819. [4](#), [41](#)
- [13] W.H. Press, S.A. Teukolsky, W. V. and Flannery, B. (1992). Numerical recipes in fortran. *Cambridge University Press*. [1](#)

Appendix

Summary of Equations

Chapter 1

1.1 Sample exponential decay differential equation	pg.2
1.2 Exact solution to Eq.(1.1)	pg.2
1.3 Forward Euler applied to Eq.(1.1)	pg.2
1.4 Backward Euler applied to Eq.(1.1)	pg.2
1.5 Iterative expression to calculate the $n + 1$ iteration from the Backward Euler method applied to Eq.(1.1)	pg.2

Chapter 2

2.1 Boltzmann Equation for spatially homogenous inelastic neutrino-electron scattering in terms of phase space density	pg.6
2.2 Legendre expansion of scattering kernels	pg.7
2.3 Orthogonality relationship of Legendre polynomials in terms of scattering kernels ..	pg.7
2.4 Angular moments integral of the distribution function	pg.8
2.5 Number density representation of the Boltzmann Equation	pg.8
2.6 Symmetry condition of scattering kernels	pg.8
2.7 Equilibrium condition of scattering kernels	pg.8
2.8 Equilibrium expression of number density	pg.8
2.9 Center of each energy bin	pg.9
2.10 Volume of each energy bin	pg.9
2.11 Equation 2.5 approximated over a finite energy domain	pg.9
2.12 Discretized expression of the number density evolution including interactions with all energy bins	pg.9
2.13 Volume average integral of number density	pg.9
2.14 Number density evolution where we use equation 2.13 to remove the integrals ..	pg.10
2.15 Expression for collision rate parameter	pg.11
2.16 Compact form of equation 2.14 in terms of flux and rate parameter	pg.11

2.17 Solution to Eq.(2.16) in the special case where F_i^+ and $\tilde{\kappa}_i$ are constant	pg.11
2.18 Mean free path of neutrino-electron collisions	pg.11
2.19 Matrix form of equation 2.16	pg.11
2.20 Definition of matrix $\mathbf{M}$ in Eq.(2.19)	pg.12
2.21 Matrix multiplication explicitly shown	pg.12

Chapter 3

3.1 Forward Euler method applied to our differential equations	pg.17
3.2 Approximation of total particle number for our discretized system	pg.18
3.3 Forward Euler method in terms of total particle number	pg.18
3.4 Our differential equations algebraically solved for the $\mathcal{N}$	pg.20
3.5 Finite difference approximation of the derivative of $\mathcal{N}$ with time	pg.20
3.6 Explicit Asymptotic method	pg.20
3.7 Explicit Asymptotic method in terms of total particle number	pg.21
3.8 Taylor series expansion of denominator in 2nd term of equation 3.7	pg.21
3.9 Demonstration that Explicit Asymptotic does not conserve particle number	pg.21
3.10 Particle conservation tolerance condition	pg.21
3.11 Tolerance condition to limit the error in $\mathcal{N}$ between iterations	pg.22
3.12 Relative error to FE	pg.23
3.13 Change in particle number from iteration to the next	pg.23
3.14 Backward Euler Method	pg.34
3.15 Algebraic equation that must be solved numerically for the unknown density	pg.34
3.16 Newton-Raphson method	pg.35
3.17 Backward Euler terminating condition	pg.35
3.18 Matrix form of Newton-Raphson Method	pg.35
3.19 Definition of a fixed point for some function G	pg.38
3.20 Definition of G	pg.38
3.21 Fixed point iteration method	pg.38
3.22 Terminating condition for Fixed point	pg.38
3.23 Residuals used for Accelerated Fixed Point	pg.41

3.24 Linear combination of residuals that must be minimized	pg.41
3.25 Accelerated Fixed Point method	pg.42

Chapter 4

4.1 Max error utilized to make fair comparison between Explicit Asymptotic and implicit methods.	pg.45
---	-------

Vita

Aaron M. Lackey - Stewart was born December 5th, 1993 in Frankfort, Kentucky where he graduated from Franklin County High School in the year 2012. After the completion of high school, he attended Morehead State University where he majored in physics with an area of concentration in astrophysics as well as mathematics. There, his research specialized in x-ray astronomy performing data analysis on blazars and supernova remnants. After graduating from undergraduate in 2016, he attended The University of Tennessee in Knoxville, TN for graduate education from 2016 to 2020 where his research focused on computational astrophysics.