5-2017

Accuracy and Stability of Integration Methods for Neutrino Transport in Core Collapse Supernovae

Kyle A. Gregory
kgrego12, kgrego12@vols.utk.edu

Accuracy and Stability of Integration Methods for Neutrino Transport in Core Collapse Supernovae

Kyle Gregory
Faculty Adviser: Dr. Mike Guidry

May 8, 2017

Abstract

When massive stars explode, an enormous amount of energy is released and carried by neutrino particles to unknown areas of phase space. By invoking the proper computation methods in simulations of these supernovae events, patterns and trends of the energy carrying neutrinos can be traced and further investigated. The Backward Euler and Explicit Asymptotic integration methods are analyzed for accuracy and stability performance by approximating simulations of the neutrino movement at varying time steps. Using constant energy columns and time step conditions provided a comparison of the numerical and analytic solutions concerning accuracy, error, stability, and density sum. Following investigation, the Backward Euler method provided highly accurate and stable results at all time steps but requires greater computational work and time while the Explicit Asymptotic approximation produced comparably accurate and stable results at all time steps while requiring less computational work but time constraints. With this conclusion, a combination of accurate and stable approximation methods can be implemented to improve the efficiency of computational results for neutrino transport in core collapse supernovae.

1 Introduction

In order to numerically approximate the integral of a given function, many computational techniques can be used to achieve the desired result. These integration techniques differ in theory and difficulty. Depending on the technique used, the accuracy of the result differs. For instance, using technique 'A' instead of technique 'B' may provide a solution more quickly, while sacrificing an amount of accuracy. For more complex computations, technique 'B' may be implemented in order to achieve a greater level of accuracy, with sacrificed speed.

Usually a topic introduced in introductory calculus, some of the primary methods for numerical integration form approximations using the area under a function curve. There are two basic methods for numerically approximating an integral. First, simply approximating an integral using the area under its curve requires the desired function be plotted, providing limits of integration for a definite integral of the function. Upon performing the definite integral of the function using those limits, the summation for the total area under the function curve is the result. In the second method known as the trapezium rule, area under the plotted function is separated into n number of intervals, with each interval size Δ. Each interval segment represents a trapezoid and the total area summation of n trapezoidal segments can be calculated to produce a result corresponding to the numerical integral. The trapezium rule is approximated with the following equation, where x_0, x_1, x_n represent coordinates along the x axis [3].

$$\int_a^b f(x)dx \approx \frac{\Delta x}{2}[f(x_0) + 2f(x_1) + 2f(x_2) + ...f(x_{n-1})]$$

where $\Delta x = \frac{b-a}{n}$.

As an example, we can use the above numerical integration methods on a differential equation of our own. An ordinary differential equation (ODE) related to the topic in Chapter 2 is

$$\frac{df}{dt} = \eta - kf. \tag{1}$$

We can further rewrite this equation to $\frac{df}{dt} = k(\frac{\eta}{k} - f)$, where η represents a positive production coefficient and k represents a positive extinction coefficient. We also supply the initial condition of $f_0 = f(t_0)$. We can still make our equation a bit more simple by substituting the factor $\frac{\eta}{k}$ with a new constant, calling it f_{Eq}. Now with an initial value f_0, we can integrate our simplified differential equation to find $f(t)$, which can then be used in the numerical integration techniques described above. Integrating from an initial time t_0 to a time t, the equation becomes

$$f(t) = f(t_0) + \int_{t_0}^{t} k(f_{Eq} - f)dT.$$

The integrated equation for $f(t)$ will then have a solution of the form

$$f(t) = f(t_0) + k\int_{t_0}^{t_1}(f_{Eq} - f)dt + k\int_{t_1}^{t_2}(f_{Eq} - f)dt + k\int_{t_2}^{t_3}(f_{Eq} - f)dt + ...$$

By dividing $f(t)$ into separate integrals, each with new limits of integration, we have also defined a time increment Δt that can be necessary during numerical integration.

Our goal is to obtain approximated solutions of the integrated equation above. However, as mentioned at the end of the paragraph above, a time step Δt can be used in lieu of the definite integrals making up the equation. In order to modify the existing equation containing definite integrals into an equation containing time steps of Δt, a procedure known as the Euler method is used. There are two common approaches using the Euler method, known as the explicit method and the implicit method. The explicit method (also known as the Forward Euler method) calculates an approximation for the state of a system at a later time $(t + \Delta t)$ using the state of the system at the current time, t. The implicit method (also known as the Backward Euler method) calculates an approximation using both the current state of the system (t) and the later system $(t+\Delta t)$. Upon using the explicit method to modify our equation, the subsequent equation containing a time step Δt is

$$f_1 = f_0 + \Delta t(k)(f_{Eq} - f_0).$$

Upon using the implicit method to modify the equation, the integrated equation becomes

$$f_1 = f_0 + \Delta t(k)(f_{Eq} - f_1).$$

Using the trapezium rule, the integrated equation becomes

$$f_1 = f_0 + \frac{\Delta t}{2}[(f_{Eq} - f_0) + (f_{Eq} - f_1)].$$

Using the explicit, implicit, and trapezium equations above for approximations, we can further generalize by replacing f_1 with f^{n+1} and f_0 with f^n. Our explicit approximation equation becomes

$$f^{n+1} = f^n + \frac{\Delta t}{2}(f_{Eq} - f^n) + \frac{\Delta t}{2}(f_{Eq} - f^{n+1})$$

and our implicit approximation equation becomes

$$f^{n+1} = f^n + \frac{\Delta t}{2} f^n + \Delta t f_{Eq} - \frac{\Delta t}{2} f^{n+1}.$$

Finally, our trapezium approximation equation becomes

$$f^{n+1} = \frac{(1 - \frac{\Delta t}{2}) f^n + \Delta t f_{Eq}}{1 + \frac{\Delta t}{2}}.$$

This concludes our first example, where we found numerical integration approximations for the differential equation $\frac{df}{dt} = \eta - kf$

Let's use these same approximation equations to estimate an actual solution to the function $\frac{df}{dt} = 1 - f$. We can use these equations to create a single, combined equation containing a variable directly corresponding to the type of approximation (explicit, implicit, or trapezium rule) performed. We will call this variable theta, θ. When θ is zero, the explicit method is used. When θ is one, the implicit method is used. When θ is one-half, the trapezium method is used. Introducing our new variable θ, the combined approximation equation becomes

$$f^{n+1} - f^n = \Delta t(1-\theta)(1-f)(f^n)\theta(1-f)(f^{n+1}).$$

Solving for f^{n+1}, we now have $f^{n+1} = \frac{\Delta t\theta + f^n + \Delta t(1-\theta)(1-f)}{1+\theta\Delta t}$. We are now ready to approximate f(t), using different values for θ and different time steps of Δt.

To illustrate the differing characteristics of the three approximation methods described, the combined approximation equation can be implemented using different values for theta and time increments. For the differential equation in question, the analytic solution is $f(t) = f_0 e^{-t} + (1 - e^{-t})$. Using MATLAB software, solutions of the combined approximation equation were computed at θ values of 0,0.50,1, and 0.49. Time step (Δt) values of .01,.10,1.0, and 3.0 seconds were used, while the final time (t) remained constant at 100 seconds. These numerical solutions were then plotted with the

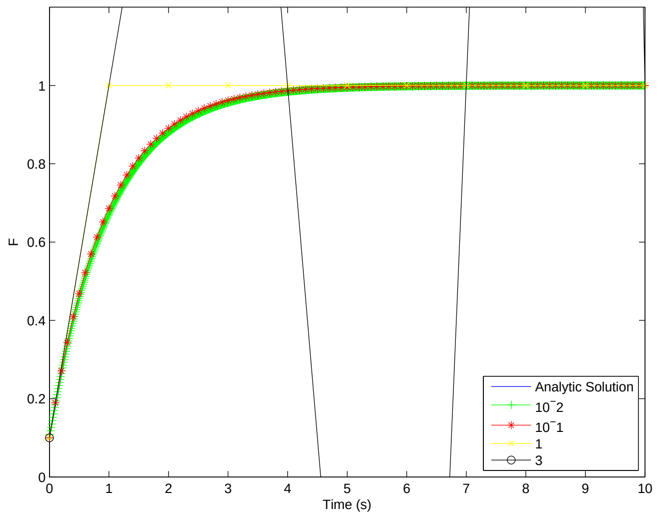


Figure 1: Explicit method approximation versus Analytic solution. The oscillating curve represents a time increment (Δt) of 3 seconds, demonstrating a lower time increment is necessary to maintain accuracy when using this method.

analytic solution versus time to assess the accuracy of the three approximation methods.

Using a θ value of 0, the resultant solution curves did demonstrate varying levels of accuracy when compared to the analytic solution curve. Featured below, Figure 1 illustrates the approximation accuracy provided by the explicit Euler method. With a time step of 3 seconds, an oscillating curve illustrates a definite amount of error inherently produced due to the larger time increment. The diverging solution is also a demonstration of instability at this large time step, where the numerical solution does not converge toward the analytic solution. With a time step of 1 second, a more accurate and asymptotic curve is produced. Much smaller discrepancies compared to the analytic solution can be seen using time steps of .1 and .01 second.

Using a θ value of 0.5, Figure 2 depicts the more accurate trapezium rule that produces a plot with better approximations illustrated by the more asymptotic curves. With a time step of 3 seconds, an oscillating but asymptotic curve is generated. Figure 1, a time step of 1,.1, and .01 second provides curves with smaller differences compared to the analytic solution.

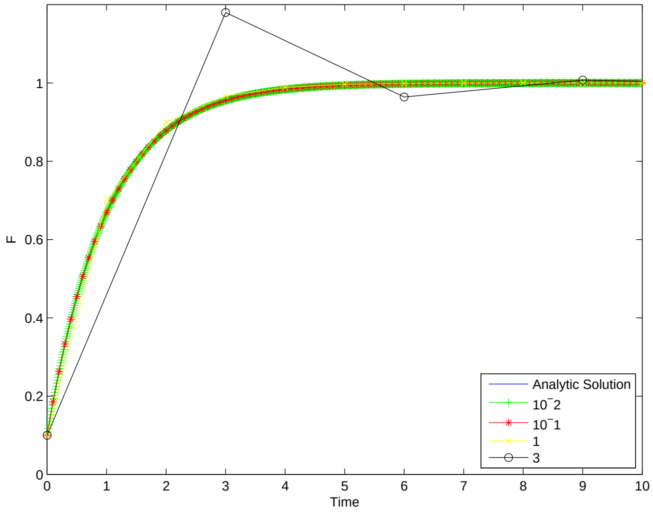


Figure 2: Trapezium rule approximation versus Analytic solution. The oscillating curve becomes asymptotic to the predicted solution and represents a time increment (Δt) of 3 seconds. At some larger time increments, the trapezium rule can still provide an accurate approximation.

Figure 3 shows a similar plot to the trapezoid method approximation used above and was generated using a θ value of 0.49. Similar to Figure 2, these curves feature an oscillating but asymptotic curve, along with more accurate curves showing little divergence from the analytic solution curve.

Using a θ value of 1, the accuracy of the implicit Euler method is illustrated below in Figure 4. This plot differs from the previous figures by not containing an oscillating curve for the 3 second time step. Every curve is asymptotic, increasing in accuracy compared with the analytic solution curve while descending in time increment value with a .01 second time increment providing the best approximation.

The four figures above are good illustrations of the varying accuracy when using different numerical methods. Better approximations using the explicit method require smaller time steps. The implicit method provides approximations that are asymptotic at every time step used, illustrating better accuracy than the explicit method. The trapezium method provides the best numerical approximations, with all asymptotic curves and an oscillating curve at the larger time step of 3 seconds. Another useful way to display the accuracy of the three methods is to view the actual error of each approximation by

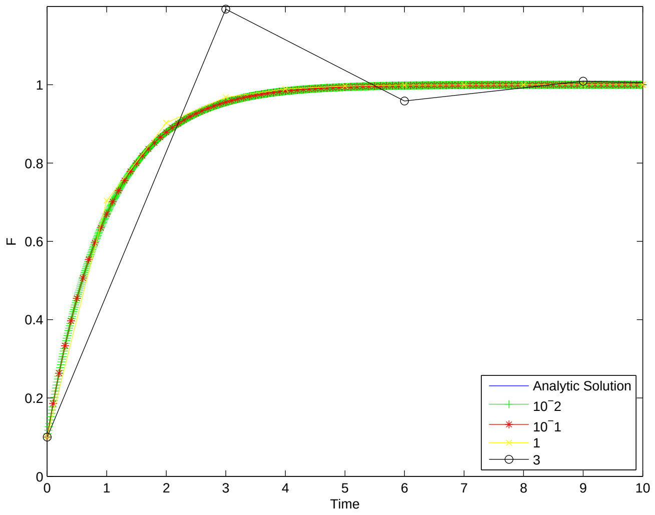


Figure 3: Variation of the trapezium rule approximation versus Analytic solution. The oscillating curve also becomes asymptotic to the predicted solution and represents a time increment (Δt) of 3 seconds.

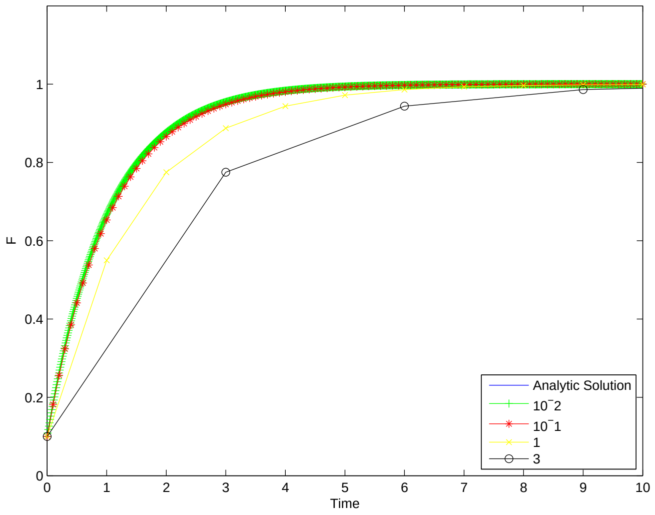


Figure 4: Implicit method approximation versus Analytic solution. Note: this method does not produce an oscillating curve for the time increment (Δt) of 3 seconds. Every curve is asymptotic to the predicted solution.

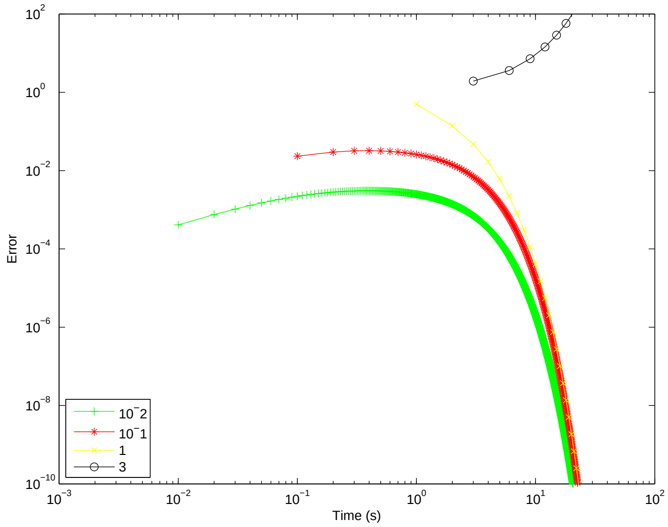


Figure 5: Explicit approximation error by time increment (Δt). The increasing error curve corresponds to the larger 3 second time increment.

comparing each curve to the analytic solution curve.

Below, Figure 5 shows the approximation error using the explicit method. This plot features three curves showing decreasing error trends, with one curve actually showing an increasing error trend. The lowest error curve is represented by the lowest time increment of .01 second, followed by .10 and 1 second. The increasing error curve corresponds to the larger, oscillating 3 second time increment from Figure 1.

Figure 6 contains the approximation error using the trapezium rule. This plot features four decreasing curves, each at different error levels with varying trends. A time increment of .01 second produces the lowest error curve indefinitely, with a result lower by up to four levels of magnitude than the other time increments. The remaining time increments show increasing error by one order of magnitude as increment number increases. The highest error curve represents a 3 second increment approximation.

Figure 7 displays the approximation error using the implicit method. Like the trapezium rule approximation error, the implicit method approximation error contains similar decreasing error curves. Like Figure 6, the lowest error curve represents the smallest time increment of .01 second. This curve also shows one order of magnitude higher error than the same increment in Figure 6. The other curves also then differ by one order of magnitude higher

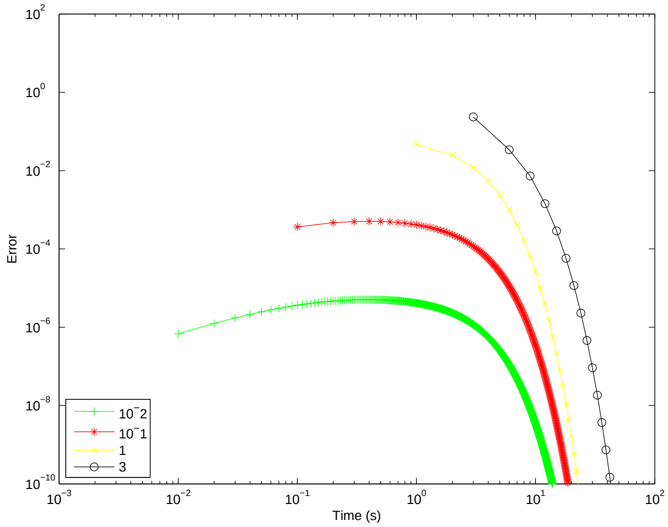


Figure 6: Trapezium rule approximation error by time increment (Δt). The error curves appear in ascending order by increasing time increment amount. The lowest error approximation is found using a .01 second time increment.

error.

Another way to view the error of each approximation type is to simply assess the maximum error produced from each method. In Figure 8 below, the table displays error results similar to results found in Figures 5,6, and 7. The table shows that the best approximations, featuring the lowest error, occurred using the trapezium method ($\theta = 0.50$). The figure also shows that the highest error occurred in the approximations using the explicit method ($\theta = 0$).

Comparing the error curves produced by each approximation type, it is clear that the trapezium rule produces the best approximation. Comparing the error levels and trends for each approximation type and time increment also show that the implicit and and explicit methods are similar but differ slightly with greater time increments. The 3 second time increment displays a noticeable difference under both methods. Under the explicit method, the larger time increment produces increasingly greater approximation error and instability. The implicit method, however, produces an initially high amount of error that decreases with time. The implicit method remained accurate and stable at all time steps, differing from the explicit method. For higher time steps, the implicit method produced accurate and stable solutions while

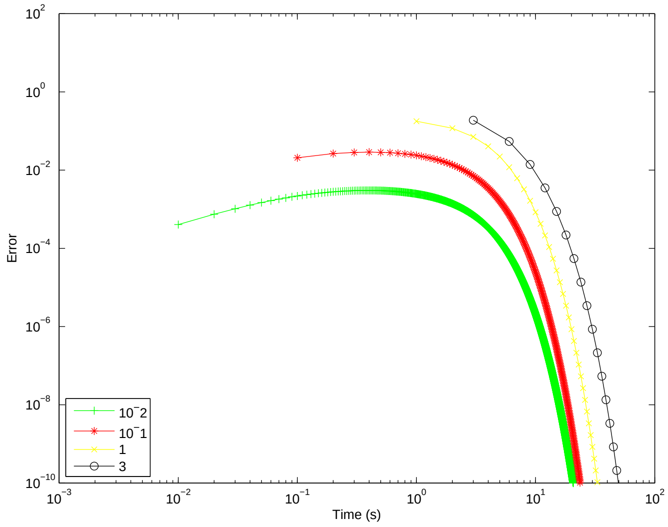
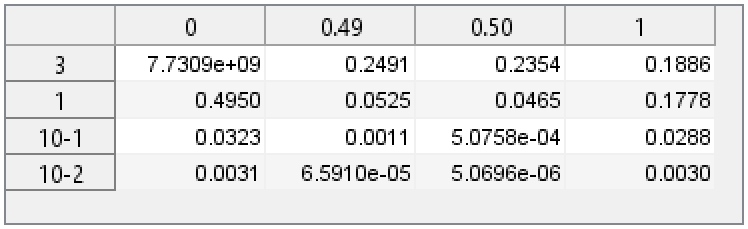


Figure 7: Implicit method approximation error by time increment (Δt). The error curves appear in ascending order by increasing time increment amount. Each error curve is one degree magnitude higher than the error curves produced using the trapezium rule. The curves using this method are comparable to those produced using the explicit method.



	0	0.49	0.50	1
3	7.7309e+09	0.2491	0.2354	0.1886
1	0.4950	0.0525	0.0465	0.1778
10-1	0.0323	0.0011	5.0758e-04	0.0288
10-2	0.0031	6.5910e-05	5.0696e-06	0.0030

Figure 8: Table of maximum error for each approximation type.

the explicit method experienced decreased accuracy and instability. This prominent difference illustrates an important conflict in numerical approximation, stability versus accuracy.

2 Approximating: Backward-Euler and Explicit Asymptotic

During Chapter 1, some of the primary forms of approximation used in computational physics were described. These three methods (Backward Euler or Implicit, Forward Euler or Explicit, and Trapezium) can be extrapolated for specific uses in real world scenarios. An expansive problem requiring powerful computational methods is on the subject of neutrino transport in core collapse supernovae, a cosmic event resulting in the creation of a neutron star or black hole following the complex destruction of a massive star. A core collapse supernova occurs after the iron rich core of a massive star becomes unstable, collapses on itself, undergoes key elemental characteristic changes, and releases a detrimental shock wave eventually leading to the star's destruction. During the shock wave, energy is released through the compression and heating of core nuclei, as well as through the transport of massless particles called neutrinos. These energy "stealing" particles are abundantly found in three areas of the core following the ignition of the shock wave, where an inner "neutrinosphere" has formed that radiates additional neutrinos. Deep within the core (below the neutrinosphere), the "neutrino-thick" region requires energy-carrying neutrinos to diffuse a greater distance to reach the outer edge of the star. Located in the neutrinosphere, the "semitransparent" region provides an easier diffusion course. Finally, at radii above the neutrinosphere, the "neutrino-thin" region provides the easiest and shortest diffusion passage for the energy-carrying particles. Considered a significant interest during this cosmic event, the exchange of energy among neutrinos in stellar fluid is analyzed to better understand the evolution of energy during the core collapse supernovae.

In order to investigate the possible energy passages during this event, computational simulations are used to repeat multiple scenarios, unfolding possible patterns among these transfers. The reliability of these solutions is greatly determined by the accuracy and stability of the approximation method (or combination of methods) used. The numerical solutions produced using the various approximation methods are not the true solution. Each numerical solution produces a finite amount of errors introduced during different times of the simulation. Error that is introduced during the approximation after a single time step is called local error [3]. Error that has accumulated during the course of many time steps during the approximation is called global error. To define local and global error, the set $(x_n, y_n) : n = 0, ..., N$ will be considered an approximation to the solution $y(x)$. The global error at time x_n is $E_n = y(x_n) - x_n$, where $y(x_n)$ is the true solution and y_n is the approximation. The absolute error for the approximation is $|E_n|$, the absolute value of the global error. The local error is $y(x_{n+1}$, the difference between the true solution and the approximation following one time step. In computational methods, the stability of the solution is partly determined by how the errors progress throughout the computation. Generally speaking, a stable approximation contains errors that, at all stages of the computation, have not amplified as the computation has progressed. An unstable approximation contains errors that become amplified with time progression. A sign of instability is a diverging solution, like the result shown in Figure 1 of Chapter 1. However, approximation accuracy and stability are also affected by the type of equation being approximated. Stiff equations are equations that progress through a wide range of time scales, sometimes occurring over several orders of magnitude. Depending on the desired level of accuracy and stability, these equations complicate the choice of approximation method used. Method "A" may be conditionally stable, requiring specific time step choices to remain stable throughout computation but be easy to compute. Method "B" may remain unconditionally stable throughout the computation but require a much greater amount of work and time to compute. This computation cost is a significant consideration when performing simulations. For instance in the neutrino transport mechanism, computations concerning the electron scattering are the most intensive calculations occurring. Obtaining an efficient approximation method for these particular computations is significant to the entire simulation. As discussed in Chapter 1, each of the three approximation methods maintains a characteristic measure of accuracy and stability during simulations. For the course of this study, we evaluate some of the useful approximation techniques for their accuracy and stability in the investigation of neutrino energy transport described above.

The neutrino transport mechanism can be described using an ordinary differential equation appearing similar to Equation (1) introduced in Chapter 1. A neutrino containing initial energy ν_i collides with an electron of initial energy e_i. Following the law of conservation of energy, the resultant neutrino is scattered carrying a final energy ν'_f while the electron carries a final energy of e'_f. Using this system, each neutrino particle's energy (E) and movement at a time (t) can be measured to form a neutrino function density variable, $N(E, x, t)$ [5]. Storing and plotting multiple neutrino particles of $N(E, x, t)$ effectively illustrates an energy spectrum. This spectrum can be used to help interpret the movement taken by the neutrinos. The ODE describing the change in neutrino density during transport contains a growth term and a diminising term. This equation is given as

$$\frac{dN}{dt}(E_{i,t}) = \int_0^{E_{max}} R^{in}(E_i, E')N(E'_i)dE' - N(E_i)\int_0^{E_{max}} R^{out}(E_i, E')dE'. \tag{2}$$

The growth term contains a rate $(R^{in}(E_i, E'))$ for the energy being transported into a new energy "bin" while the diminishing term contains a rate $(R^{out}(E_i, E'))$ for the energy being transported away from the current energy bin. The diminishing term also contains a coefficient $(N(E_i))$ corresponding to the present energy in the current energy bin. This coefficient preceded by the subtraction sign signifies the dimenishing term as only a reduction from the current energy bin. When the coefficient is zero signifying no present energy in the current energy bin, the diminishing term is respectively zero. Rather than using integrals to describe this movement, summations can be used in the equation instead. In summation form, the equation reads

$$\frac{dN_i}{dt}(E_i, t) = \sum_{k=1}^{N_g} R^{in}(E_i, E_k)N(E_k)\Delta E_k - N(E_i)\sum_{k=1}^{N_g} R^{out}(E_i, E_k)\Delta E_k. \tag{3}$$

The equation can be further reduced by simplifying each term accordingly. Within the growth term, a variable $\hat{R}_{in}$ represents $\hat{R}_{in} = \Delta E_k R^{in}(E_i, Ek)$. From the diminishing term, a similar variable $\hat{R}_{out}$ represents $\hat{R}_{out} = \Delta E_k R^{out}(E_i, Ek)$. Redefining the summation equation using these new variables, the differential equation becomes

$$\frac{dN_i}{dt}(E_i, t) = \sum_{k=1}^{N_g} \hat{R}^{in}_{ik} N_k - N_i \sum_{k=1}^{N_g} \hat{R}^{out}_{ik}. \tag{4}$$

Lastly, the diminishing term can be simplified by introducing a variable to represent the summation, $R_i = \sum_{k=1}^{N_g} \hat{R}^{out}_{ik}$. The differential equation now reads

$$\frac{dN_i}{dt}(E_i,t) = \sum_{k=1}^{N_g} \hat{R}_{ik}^{in} N_k - N_i R_i.$$

Upon refactoring N, the equation becomes

$$\frac{dN_i}{dt}(E_i,t) = \sum_{k=1}^{N_g} (\hat{R}_{ik}^{in} - \delta_{ik} R_k) N_k.$$

In this form, the term located inside the parenthesis can be expressed as a matrix. This matrix term will be denoted as L_{ik}. The term outside of parenthesis, N_k, can be expressed as a vector. This vector term is $\vec{N} = (N_1, N_2, ..., N_{Ng})$. Redefining the differential equation using these forms, the equation becomes

$$\frac{d\vec{N}}{dt} = L\vec{N}. \tag{5}$$

Using Equation (5) as a basis, the next step is deriving and implementing the relevant approximation techniques discussed in Chapter 1.

2.1 Euler Methods

For the energy transport mechanism through neutrino particles, it is appropiate to derive two of the popular techniques discussed in Chapter 1. For the Implicit or Backward Euler approximation method described in Chapter 1, the Backward Euler technique as Equation (5) is

$$\vec{N}^{n+1} - \vec{N}^{n} = \Delta t L \vec{N}^{n+1}. \tag{6}$$

Solving Equation (6) for $\vec{N}$, the equation reads $\vec{N} = (I - \Delta t L)\vec{N}^{n+1}$ [1]. Lastly, upon solving for the new energy bin term $\vec{N}^{n+1}$, the implicit form of Equation (5) is

$$\vec{N}^{n+1} = (I - \Delta t L)^{-1} N^{n}. \tag{7}$$

Likewise for the Explicit or Forward Euler method, Equation (7) instead has the form $\vec{N}^{n+1} = \vec{N}^{n} + \Delta t L \vec{N}^{n}$. Upon simplifying, the explicit form of Equation (5) is

$$\vec{N}^{n+1} = (I + \Delta t L)\vec{N}^{n}. \tag{8}$$

With an equation for two valuable approximation techniques, it is now possible to begin computing numerical solutions that describe the energy transport while focusing on the error and stability in each method.

In order to guarantee correctness for the analysis of the approximation techniques, it is necessary to have an exact solution to compare the approximation results with. For this research, the exact solutions were computed using an analytic equation for the neutrino density of various columns (energy bin columns). As an exact solution to the neutrino energy transport mechanism, this equation offers a suitable comparison against other non-exact, numerical solutions approximating the neutrino density for the same energy bin columns. Using Equation (5), matrix (L) in the equation was made diagonal as $L = R\lambda R^{-1}$. Equation (5) now reads

$$\frac{d\vec{N}}{dt} = R\lambda R^{-1}\vec{N}. \tag{9}$$

By multiplying Equation (9) by R^{-1}, the equation becomes

$$R^{-1}\frac{d\vec{N}}{dt} = R^{-1}R\lambda R^{-1}\vec{N}.$$

Where $R^{-1}R$ becomes an identity matrix, the equation becomes

$$\frac{d\vec{W}}{dt} = \lambda\vec{W} = \frac{dW_i}{dt} = \lambda_i W_i.$$

In this equation, $i = 1, ..., N_g$. Using the above matrix relation in Equation (6), an analytic form of the equation to generate results is

$$N_i^{n+1} = W_i^n + \Delta t\lambda_i W_i^n = (1 + \Delta t\lambda_i)W_i^n. \tag{10}$$

Using Equation (10), a MATLAB program was created to generate analytic solutions, containing the appropriate elements to compute the analytic neutrino density results for different time ranges (t) and time steps (dt). Next, a MATLAB program was created to compute the Backward Euler numeric solutions utilizing Equation (7). This program required the input of a minimum and maximum time step, as well as an initial and final computation time. For this research, all simulations (analytic and numerical solutions) were performed using the same five time steps of 10^{-9}, 10^{-8}, 10^{-7}, 10^{-6}, and 10^{-5} seconds. In addition to these constant time steps, the program offered a factor that could employ different levels of a varying time step. For an introductory analysis of the Backward Euler method, the numerical solutions for five different energy bin columns (1,10,15,25,and 35) using the five time steps discussed above were computed using this program. These numeric solutions were then plotted with the analytic solutions comparing identical energy columns at equal time step conditions. Just as the five time steps remained constant throughout every simulation, the same five energy bin columns of 1,10,15,25, and 35 remained constant throughout every analysis as well. This allowed the time step conditions and energy bin columns to act as control variables during analysis.

The resulting five time step plots for the Backward Euler (BE) simulation are presented in Figures 9-13. One key characteristic of the BE numerical solutions is illustrated in every time step plot. The BE numerical solutions maintain consistently high accuracy throughout the entire computation time while compared to the analytic solution. In Figures 9 and 10, the numerical solutions remain greatly comparative to the analytic solutions throughout the total time. The use of a smaller time step generated a larger range of results in these simulations. With more results to analyze, the figures provide a greater depiction of accuracy than the remaining time steps. To determine what kind of accuracy to expect for the Backward Euler method, the local and global error orders can be derived using the equation defining the method. Consider the BE method to be summarized as $y(x + \Delta t) = y(x) + \Delta t y^{(1)}(x + \Delta t)$ [3]. To find these expected values, $y(x)$ in the equation is expanded in a Taylor series around x_1 to become

$$y(x_1 - \Delta t) = y(x_1) - \Delta t y^{(1)}(x_1) + \frac{1}{2}\Delta t^2 y^{(2)}(\chi). \tag{11}$$

In this equation, $y(x_1 - \Delta t) = y(x_0) = y_0, y(x_1) = y(x_0 + \Delta t)$, and $y^{(1)}(x_1) = f(x_1, y_1)$. Using these relations, Equation (11) can now be redefined as

$$y(x_0 + \Delta t) = y_0 + \Delta t f(x_1, y_1) - \frac{1}{2}\Delta t^2 y^{(2)}(\chi).$$

Using this definition, the local error for the BE method is

$$E^{local} = -\frac{1}{2}\Delta t^2 y^{(2)}(\chi) \propto \Delta t^2. \tag{12}$$

For the Backward Euler method, the local error decreases as the square of the time step size. This explains that the local error for the smaller time step is 1000000 higher than the accuracy of the 10^{-5} time step. How does this compare to the global error? If $y(0) = y_0$ and we need to find $y(X)$, the appropriate amount of steps is $\frac{X}{h}$. With a constant local error for $0 \leq x \leq X$, and $E(X)$ is the global error for x=X, the global error $E(X)$ is

$$E(X) \propto numberofiterations \times E^{local} \propto \frac{X}{\Delta t} \times \Delta t^2 \propto \Delta t. \tag{13}$$

Using this logic, the global error for the Backward Euler method decreases linearly with the time step size. This verifies that the simulations of magnitude smaller time step sizes (Figures 9 and 10) actually produce magnitudes less of global error. Figures 11, 12, and 13 contain an increasingly smaller range of results for the neutrino density, inherently carrying more local and global error. However, from those figures, the accuracy of the numerical solutions still remain high when compared to the analytic solutions. It is also apparent that stability is not sacrificed at during any of the time steps, as no signs of a diverging solution appear in Figures 9-13. The solutions in Figures 12 and 13 are consistently stable, converging to obvious equilibrium levels for each energy bin. Figure 13 greatly depicts a stable numerical solution by containing almost linear plots.

After evaluating Figures 9-13, the Backward Euler method computes highly accurate results using the small and large time steps. The results using this method were also highly stable at all time steps. For the purpose of computational simulations, it appears the Backward Euler method can dependably be utilized under large and small time step conditions to approximate a numerical solution. However to formally ensure the capabilty of this method while also examining other significant neutrino behaviors revealed by the Backward Euler method, it will be necessary to further consider the mathematics surrounding Equation (4). Consider when $\frac{dN_i}{dt}$ is approximately equal to 0. When this occurs in Equation (4), $\vec{R^{in}}N$ equals $\vec{R^{out}}N$ or

$$\sum_{k=1}^{N_g} \hat{R}_{ik} N_k = N_i R_i. \tag{14}$$

The condition when $\frac{dN_i}{dt}$ approximately equals zero signifies an equilibrium status for the energy column, where generally the number of neutrinos entering the column is equal to the number of neutrinos exiting the column. Using this scenario, Equation (14) can be modified to progress the investigation of notable attributes for the various energy columns. To analyze the

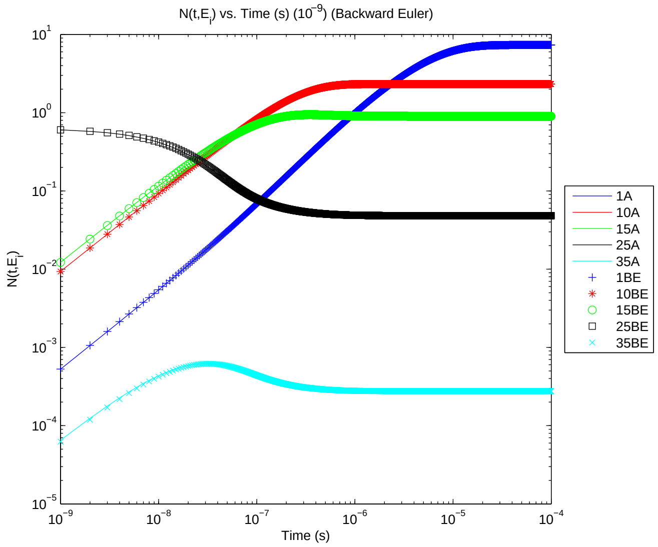


Figure 9: Backward-Euler method: Analytic vs. Numerical solution showing Neutrino density vs. Time at time step of 10^{-9}. Notice the very high accuracy throughout the large total time.

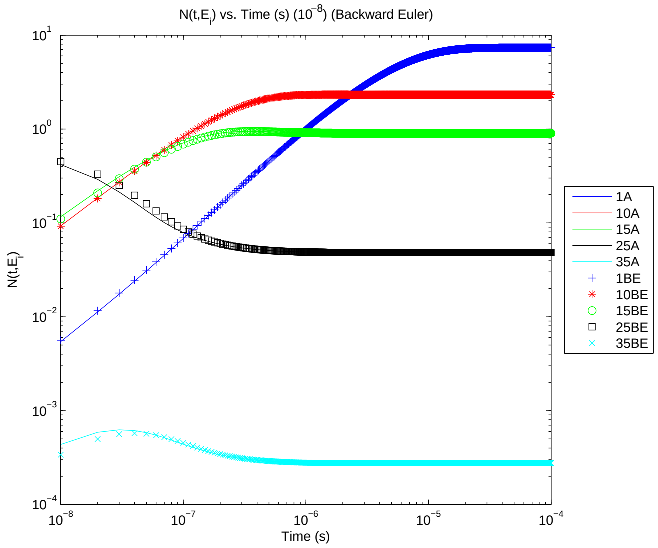


Figure 10: Backward-Euler method: Analytic vs. Numerical solution showing Neutrino density vs. Time at time step of 10^{-8}. The numerical solution remains very accurate throughout the total time.

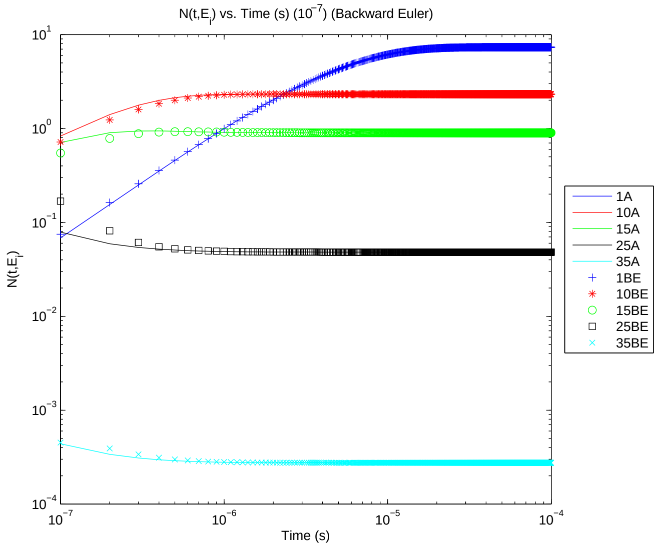


Figure 11: Backward-Euler method: Analytic vs. Numerical solution showing Neutrino density vs. Time at time step of 10^{-7}. As the time step (dt) increases, the amount of results decreases.

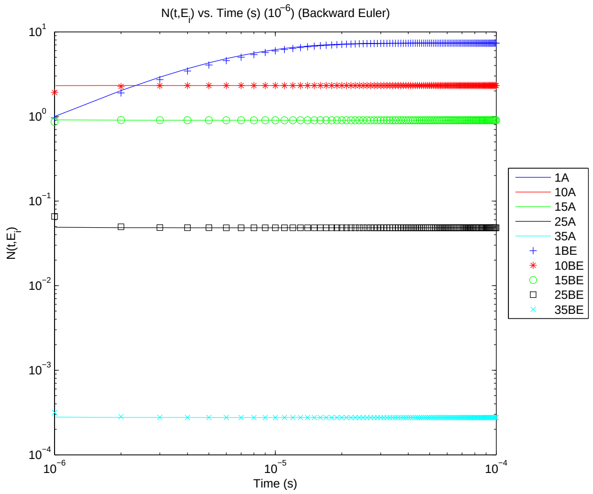


Figure 12: Backward-Euler method: Analytic vs. Numerical solution showing Neutrino density vs. Time at time step of 10^{-6}. Using a higher time step, the number of results is greatly decreased but accuracy is not sacrificed. The numerical solution maintains high stability.

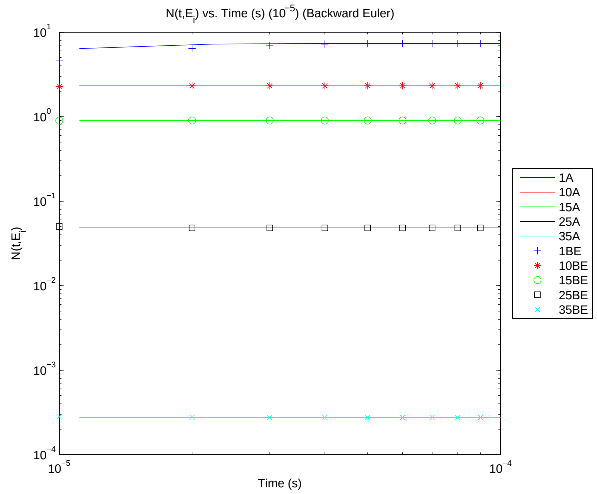


Figure 13: Backward-Euler method: Analytic vs. Numerical solution showing Neutrino density vs. Time at time step of 10^{-5}. Using this high time step, the stability of the numerical solution remains consistent.

condition of equilibrium for each energy column, it was necessary to distinguish the two terms in Equation (14). As discussed earlier in Chapter 2, the first term ($\sum_{k=1}^{N_g} \hat{R}_{ik} N_k$) is a growth term. For this reason, it will be characterized in the following figures as F_i^+. Contrarily, the other term ($N_i R_i$) is a diminishing term. Appropriately, this term will be characterized instead as F_i^-. Using MATLAB, a program was created to plot the two terms together, again illustrating the neutrino density versus time for the various energy bins and time step conditions. Using these plots, it was also possible to further analyze the significance of the equilibrium condition noted above.

In Figures 14-18, the numerical neutrino density solutions for F_i^+ and F_i^- are plotted against time. In these figures, each energy column reaches equilibrium at a varying time. However, the order that the columns reach equilibrium remains consistent in all figures. Figure 14 illustrates that energy bin column (35) reaches equilibrium first, while energy bin column (1) is noticeably the last column to reach equilibrium. Column (1)'s condition agrees with Equation (4), where the diminishing term (F_i^-) begins the simulation containing zero neutrinos before continuously increasing its amount of energy and reaching equilibrium status with the growth term. This pattern is reflected for every column except column 25. The F_i^+ term for each

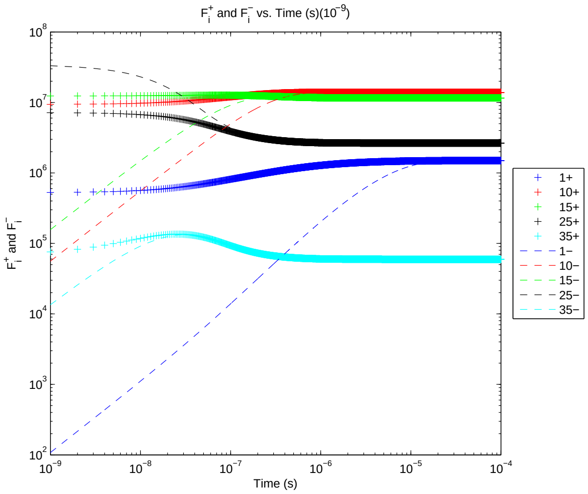


Figure 14: $F_i^+ = F_i^-$ [Backward-Euler method]: Numerical solution showing Neutrino density vs. Time at time step of 10^{-9}. At the lowest time step, the earliest equilibrium time is reached by the (35) energy bin group while the latest time is reached by the (1) energy bin group.

column begins the simulation with a finite amount of initial neutrinos before showing small neutrino movement prior to reaching equilibrium status with the F_i^- term. With an interest in limiting the error of the approximation results for this mechanism, the status of equilibrium also acts as key feature for error reduction as well. Upon evaluating the absolute error of the numerical results from Figures 14-18, the significance of achieving equilibrium is revealed. In Figure 19, the equilibrium point for each energy bin column marks a period for diminishing error. Prior to reaching equilibrium, the period of growth and loss occurring in Equation (2) carry higher error during the simulations. Using this unique point, perhaps an approximation method (or combination of methods) can be implemented to limit the error prior to and after the equilibrium point.

For approximating the neutrino energy transport mechanism, the Backward Euler method provided a highly accurate and stable approximation at small and large time step conditions. However, the extensive mathematics required to implement the Backward Euler method for large systems inherently create discrepancies for performing the simulations. A great amount of computational power and time are necessary to simulate large systems

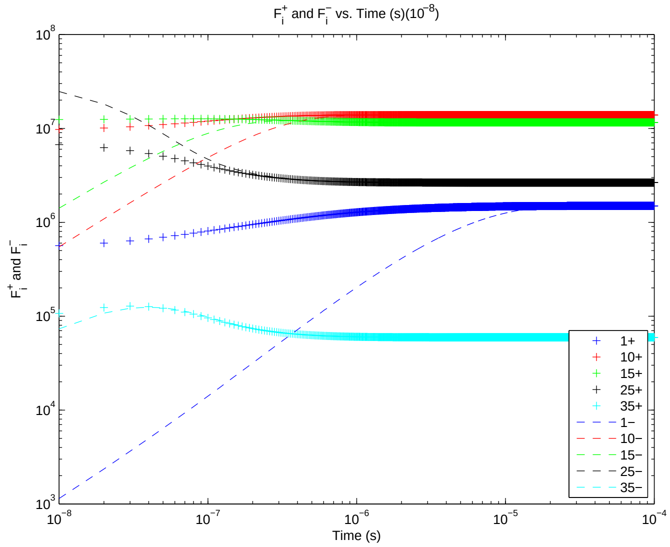


Figure 15: $F_i^+ = F_i^-$ [Backward-Euler method]: Numerical solution showing Neutrino density vs. Time at time step of 10^{-8}. As in Figure 14, the earliest equilibrium time is reached by the (35) energy bin group while the latest time is reached by the (1) energy bin group.

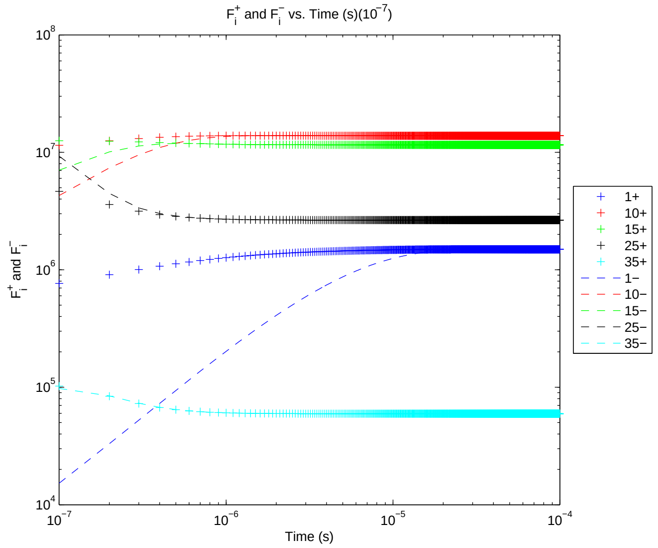


Figure 16: $F_i^+ = F_i^-$ [Backward-Euler method]: Numerical solution showing Neutrino density vs. Time at time step of 10^{-7}.

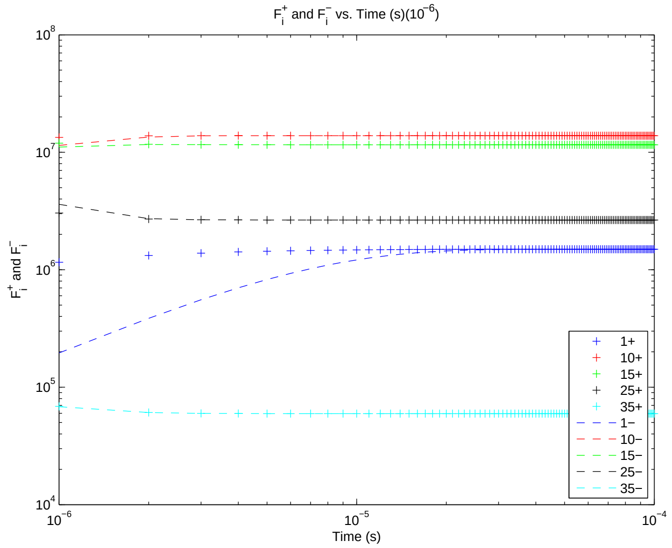


Figure 17: $F_i^+ = F_i^-$ [Backward-Euler method]: Numerical solution showing Neutrino density vs. Time at time step of 10^{-6}.

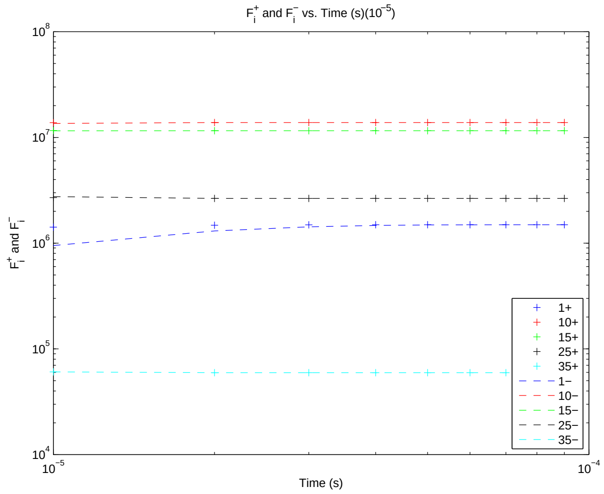


Figure 18: $F_i^+ = F_i^-$ [Backward-Euler method]: Numerical solution showing Neutrino density vs. Time at time step of 10^{-5}. Using this high time step, the only energy bin group to not already be in equilibrium is the (1) energy bin group.

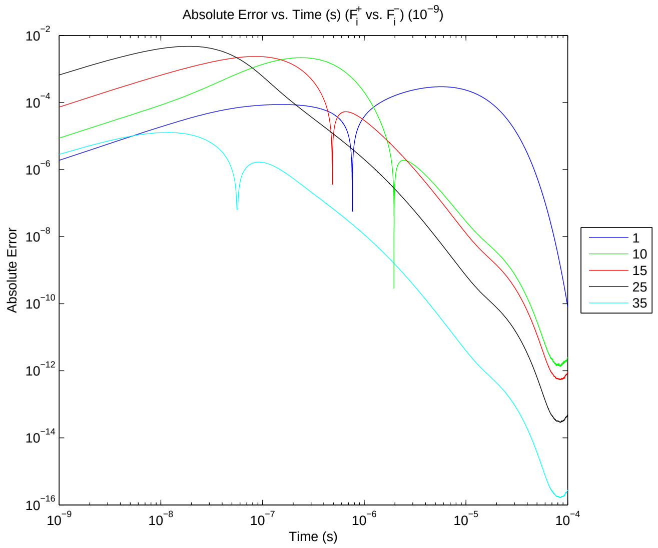


Figure 19: $F_i^+ = F_i^-$ [Backward-Euler method]: Absolute error vs. Time at time step of 10^{-9}. Upon reaching equilibrium status, the absolute error for every energy bin group follows a decreasing behavior.

using this method. According to Equation (7), the Backward Euler method contains the division operation of $\frac{1}{1-\Delta tL}$ at every time step. The effect of this operation becomes matrix inversion calculations, a computationally costly task. According to Equation (8), the Forward Euler method instead requires the multiplication operation of $1 + \Delta tL$ for each time step [3]. This operation leads to the computations of matrix multiplication, a much more efficient computation load. With computational costs in mind, other approximations technqiues must be evaluated and considered to compute numerical results for this expansive mechanism.

2.2 Explicit Asymptotic Approximation

Another method that can be utilized to approximate numerical solutions for neutrino energy transport is known as the Explicit Asymptotic (EA) approximation. This method focuses on the equilibrium condition discussed and plotted above. The method is based on the asymptotic limit that occurs when $\frac{dN_i}{dt} \approxeq 0$ [6]. From Equation (14), it is apparent F_i^- can be described as

$$F_i^- = (R_{i1}^{out} + R_{i2}^{out} + ...R_{iN_g}^{out})N_i. \tag{15}$$

As R_i^{out} characterizes a depleting rate factor to N_i, a characteristic timescale for this depletion can be derived using this rate factor. Using the definition of F_i^- in Figure 15, $F_i^- = N_i \sum_{ik}^{out} = N_i k_i$ and $\frac{dN_i}{dt} = F_i^+ - N_i k_i$. With algebraic manipulation, this equation becomes $\frac{1}{N_i}\frac{dN_i}{dt} = \frac{F_i^+}{N_i} - k_i$. The characteristic timescale is $\tau_j^i = \frac{1}{\vec{R_i^j}}$ [5]. Using this timescale and Equation (15), we can rewrite Equation (14) as

$$N_i = \frac{1}{\vec{R^{out}}}(F_i^+ - \frac{dN_i}{dt}). \tag{16}$$

Using a finite-difference approximation at a time step t_n, Equation (16) can be described as

$$N_i(t_n) = \frac{F_i^+(t_n)}{\vec{R^{out}}(t_n} - \frac{1}{N_i(t_n)}\Big|_{t=t_n}. \tag{17}$$

Applying the asymptotic limit for when $F_i^+ \approxeq F_i^-$ and implying $\frac{dNi}{dt} \approxeq 0$, a first approximation for Equation (17) is given by

$$N_i(t_n) = \frac{F_i^+(t_n)}{\vec{R^{out}}}. \tag{18}$$

Rewriting Equation (18) using a correction term, a second approximation for Equation (17) is

$$N_i^{n+1} = \frac{N_i^n + \Delta t F_i^+}{1 + k_i \Delta t}. \tag{19}$$

With Equation (19), the neutrino densities can now be approximated using this method. Using the same simulation procedure as for the Backward Euler method, plots of neutrino density versus time were created by implementing Equation (19) and the analytic solution in a MATLAB program. Upon plotting the neutrino density versus time for the same energy columns and time steps, there is a noticeable accuracy difference using the Explicit Asymptotic method. Beginning in Figure 21, energy bin column (25) highly differs from the analytic solution during the entire time scale. At this time step, the energy bin columns reached the incorrect equilibrium. This equilibrium difference occurs for every column in the figure. In Figure 22, this accuracy difference becomes larger for the energy bin columns. This accuracy and

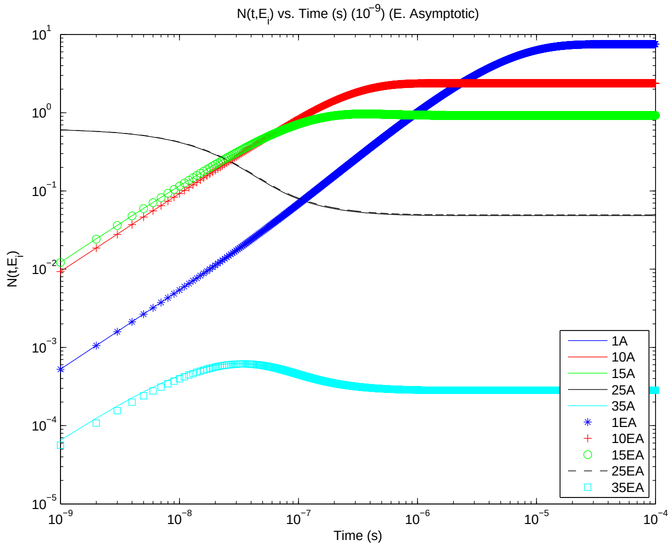


Figure 20: Explicit Asymptotic approximation (EA): Numerical/Analytic solution showing Neutrino density vs. Time at time step of 10^{-9}. Using this approximation with a small time step, good accuracy is maintained throughout the total time.

equilibrium difference remains in Figures 23 and 24. It is apparent from Figures 20-24 that the Explicit Asymptotic approximation provides less accuracy at greater time step conditions, as the accuracy loss was not noticeable for the 10^{-9} second time step. To view the extent of this accuracy loss, a plot of the absolute error versus time was plotted for the approximations of this method. Figure 25 presents the absolute error versus time for the energy bin columns using a time step of 10^{-9} seconds. In this figure, the error increases before reaching equilibrium for each energy bin column, an attribute also seen in the Backward Euler method approximations. This increase is most apparent for energy bin column (1), while energy bin columns (35) and (25) suffer a much shorter period of increase. In Figure 26, the same absolute error versus time for the Backward Euler method and Explicit Asymptotic approximation are plotted together for comparison. In this figure, the EA approximation produces an observable attribute that differs from the BE method. Upon reaching equilibrium, the error for the EA approximation does not show an appreciable drop but increases before entering a constant error amount.

Using a table featuring the maximum error from each time step of the

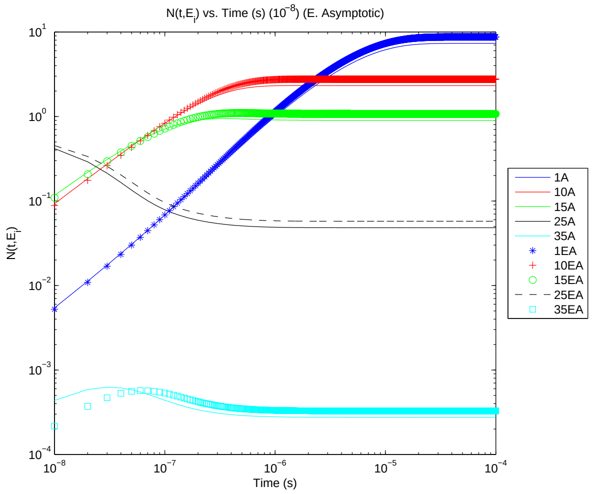


Figure 21: Explicit Asymptotic approximation (EA): Numerical/Analytic solution showing Neutrino density vs. Time at time step of 10^{-8}. The accuracy for the energy bin group (25) is noticeably weakened using this approximation.

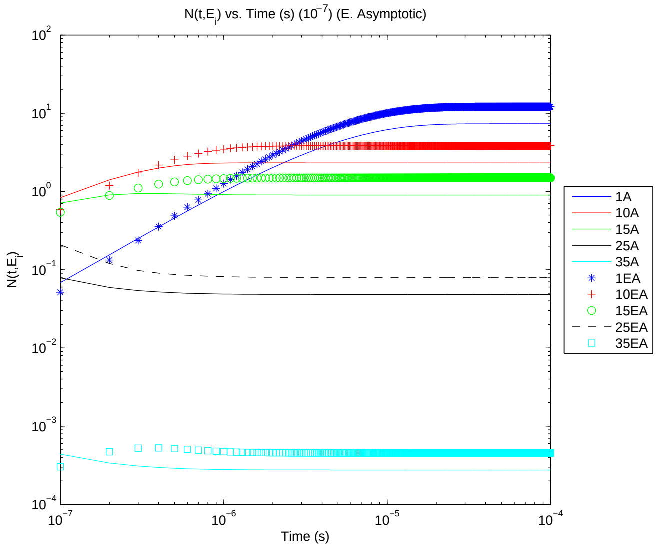


Figure 22: Explicit Asymptotic approximation (EA): Numerical/Analytic solution showing Neutrino density vs. Time at time step of 10^{-7}. As the time step becomes larger, accuracy is reduced using this approximation. An equilibrium point has not been reached by any energy bin group.

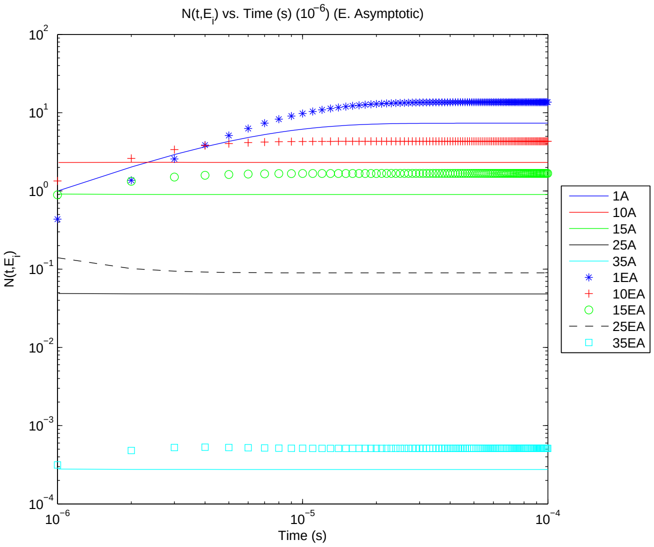


Figure 23: Explicit Asymptotic approximation (EA): Numerical/Analytic solution showing Neutrino density vs. Time at time step of 10^{-6}.

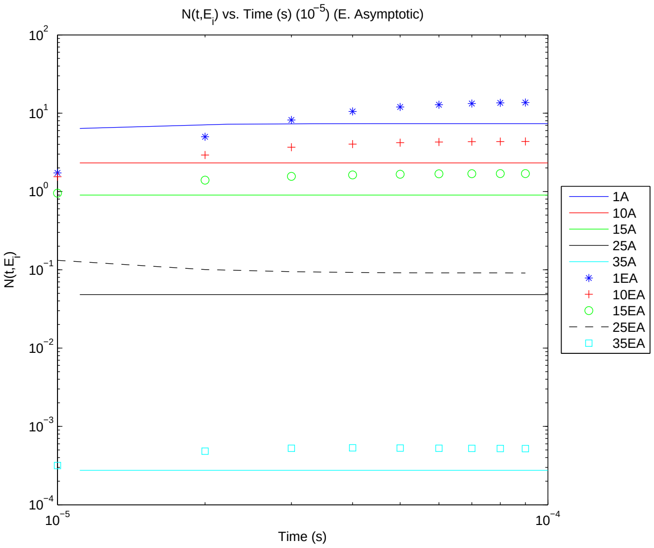


Figure 24: Explicit Asymptotic approximation (EA): Numerical/Analytic solution showing Neutrino density vs. Time at time step of 10^{-5}. Using this larger time step, the numerical solution is noticeably different than the analytic solution.

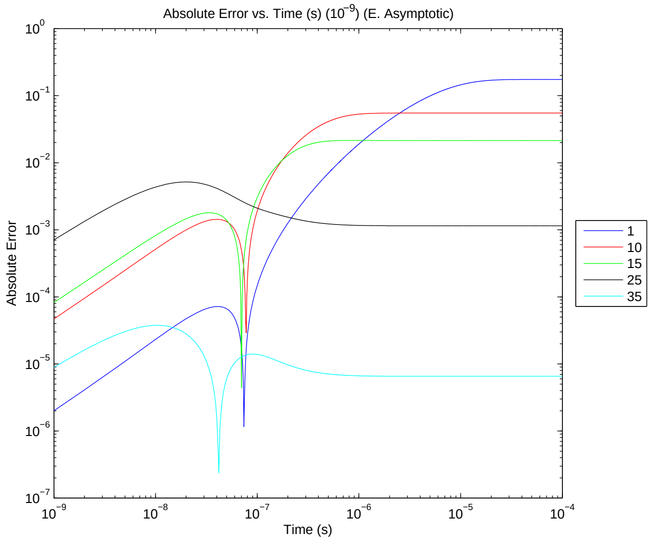


Figure 25: Explicit Asymptotic approximation (EA): Absolute error vs. Time at time step of 10^{-9}. Prior to reaching equilibrium, the absolute error increases before leveling for every energy bin group.

EA approximation plot, it is also simple to show the error variation between energy columns and time steps. Figure 27 presents these maximums, where energy bin group (35) consistently contains less error. This figure also shows a trend of decreasing maximum error for descending order of energy bin column. With the highest maximum error for every column, the largest time step at 10^{-5} seconds was also evaluated for error trends. When viewing this time step's error in Figure 28, the method retained the consistent error trend reached following equilibrium in Figure 25. This figure helps illustrate that greater but constant error is present at large time steps. From Figures 20-24, the Explicit Asymptotic approximations produced stable numeric solutions throughout the entire time scale. Another notable condition for this approximation method can be found within a plot of the characteristic timescale. Considering the characteristic timescale definition discussed prior to establishing Equation (16), $\tau_j^i = \frac{1}{\vec{R}_i^j}$, Figure 29 contains the plot of this equation and displays the evolution of F_i^+ and F_i^- from 10^{-9} to 10^{-4} seconds. In the figure, the difference evolves in a logarithmic decay scale where energy bin column (1) continuously changes and does not show this decay until a time of approximately 10^{-6} seconds. The remaining energy columns evolve at a

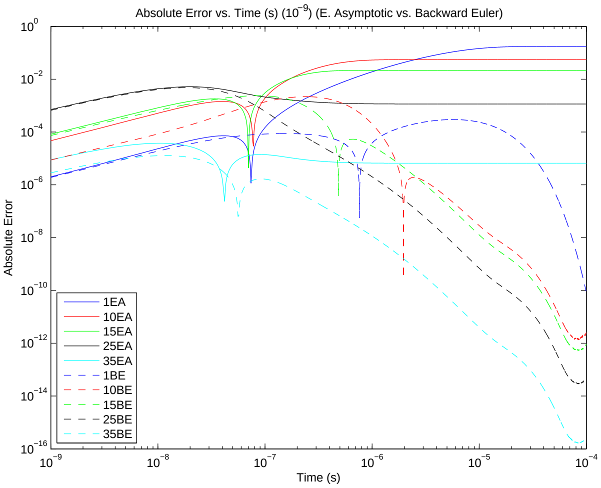
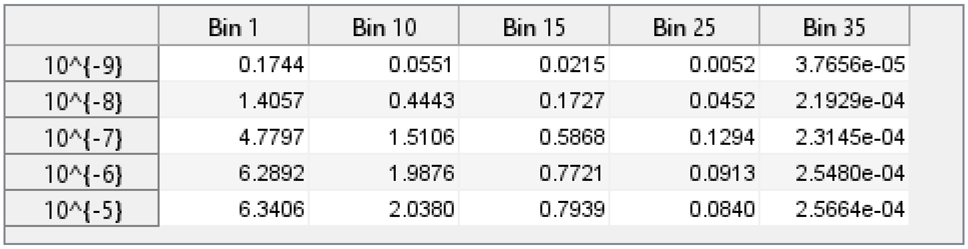


Figure 26: Explicit Asymptotic approximation (EA) vs. Backward Euler (BE) method: Absolute error vs. Time at time step of 10^{-9}. The EA approximation does not demonstrate a period of diminishing error following equilibrium status, as seen in the BE method. The EA approximation produces and maintains more error than the BE method during the total time scale of the simulation.

much greater decay rate throughout the entire time scale. Another characteristic feature to observe for the EA approximation is the density sum, or the count of total neutrino particles throughout the simulation. Figure 31 displays the density sum versus time for the EA approximation and the BE method during a simulation with a time step of 10^{-9} seconds. In this figure, the EA approximation contains a short period of decrease and a long period of increase before approaching equilibrium and becoming constant at a substantially higher final density sum. The BE method displayed a constant density sum throughout the entire simulation. Computing the numerical results using this fluctuating amount of neutrino particles likely contributed to some of the error found in the numerical solutions. Using a higher time step of 10_{-6} seconds, Figure 32 illustrated that the density sum fluctuation increases with increasing time step. Though the density sum for the EA approximation fluctuates away from equilibrium, particle conservation does occur near equilibrium. The Explicit Asymptotic approximation contained some noticeable error in the numerical solutions of Figures 22 and 23, using an adaptive (or varying) time step could improve the results for this method. Figure 30 contains a plot of neutrino density versus time using the EA approximation and BE method at an adaptive time step of 10^{-6} to 10^{-11} seconds. In the figure, the numerical solutions for both methods are highly similar. Compared to the results of Figures 22 and 23, the results in this figure demonstrated the EA approximation has the ability to produce results comparable to results from the Backward Euler method.

3 Conclusions and Outlook

Upon evaluating the accuracy and stability of the Backward Euler method and Explicit Asymptotic approximation, one method showed significant reliability at all tested time steps. The Backward Euler method provided very accurate numerical solutions during every time step, while the Explicit Asymptotic approximation provided decreased accuracy at larger time steps. Both approximations, however, consistently produced stable solutions, with numerical solutions converging to the analytic solutions. When comparing the methods for error, there was an appreciable difference. The Backward Euler method provided a period of increasing error prior to reaching equilibrium, followed by a period of exceptionally decreasing error. The Explicit Asymptotic approximation also showed a period of increasing error before reaching equilibrium. Rather than showing a period of decrease following equilibrium, however, the Explicit Asymptotic approximation solutions experienced a period of slight increase or decrease before becoming constant. For a simulation using a time step of 10^{-9} seconds, the least amount of final error from the Explicit Asymptotic approximation was four magnitudes greater than the highest amount of error produced from the Backward Euler method. The Explicit Asymptotic approximation was also shown to add or lose neutrino particles during simulations, an issue not shared by the Backward Euler method. This issue likely increased the error obtained during Explicit Asymptotic approximations. Though it produced higher error for simulations using a constant time step, the Explicit Asymptotic approximation was shown to produce numeric results highly similar to Backward Euler method results when an adaptive time step was implemented. Considering the accuracy and stability qualities of both methods, the Backward Euler method consistently provided more reliable numerical results. Depending on the type of simulation, however, the Backward Euler method may demand much greater computation work and time. This cost is not always appropriate or manageable, which is why the Explicit Asymptotic approximation may be the suitable approximation method choice. When performing a simulation, the desired accuracy and computational cost should be determined before selecting an approximation method. When applying an appropriate time step condition and utilizing their qualities efficiently, both the Backward Euler method and the Explicit Asymptotic approximation perform effectively and reliably as numerical approximation techniques. For future simulations of the neutrino transport mechanism, implementing a combination of the Forward Euler technique and Explicit Asymptotic approximation is expected to save computation resources without sacrificing extra accuracy or stability. This method, known as the Hybrid method, applies the Forward Euler method at small time steps away from equilibrium and the Explicit Asymptotic approximation at all time steps close to equilibrium. This technique utilizes the accuracy,stability, and computational efficiency of both methods to compute better approximations.



	Bin 1	Bin 10	Bin 15	Bin 25	Bin 35
10^{-9}	0.1744	0.0551	0.0215	0.0052	3.7656e-05
10^{-8}	1.4057	0.4443	0.1727	0.0452	2.1929e-04
10^{-7}	4.7797	1.5106	0.5868	0.1294	2.3145e-04
10^{-6}	6.2892	1.9876	0.7721	0.0913	2.5480e-04
10^{-5}	6.3406	2.0380	0.7939	0.0840	2.5664e-04

Figure 27: Explicit Asymptotic approximation (EA): Maximum absolute error of each energy bin group at a various time step conditions. The energy bin group (35) consistently contains the least error throughout every time step, while energy bin group (1) contains much higher error.

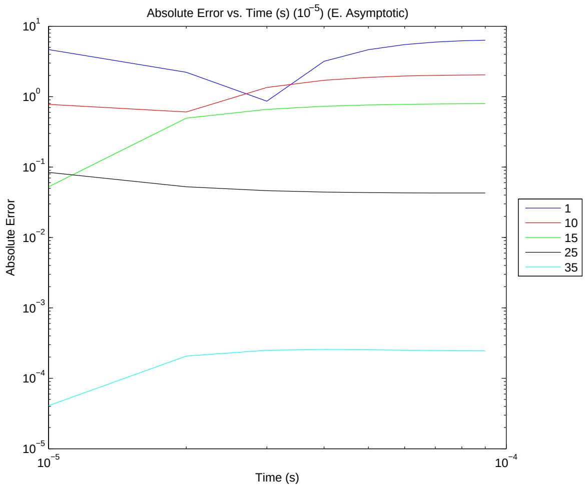


Figure 28: Explicit Asymptotic approximation (EA): Absolute error vs. Time at time step of 10^{-5}. Using a large time step, the absolute error remains fairly consistent for every energy bin group.

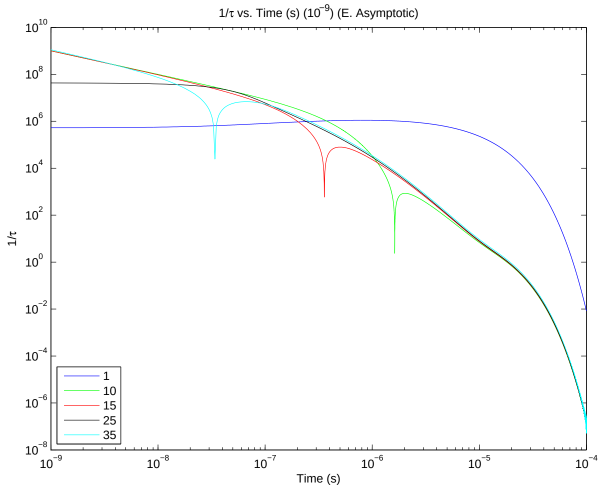


Figure 29: Explicit Asymptotic approximation (EA): Characteristic timescale using $1/\tau$ vs. Time at time step of 10^{-9}. The evolution of F_i^+ and F_i^- shows that a difference between the two of 10^{-5} occurs at an approximate time of 10^{-4} for all energy bin groups other than (1). At this time, energy bin group (1) shows a greater difference of approximately 10^1.

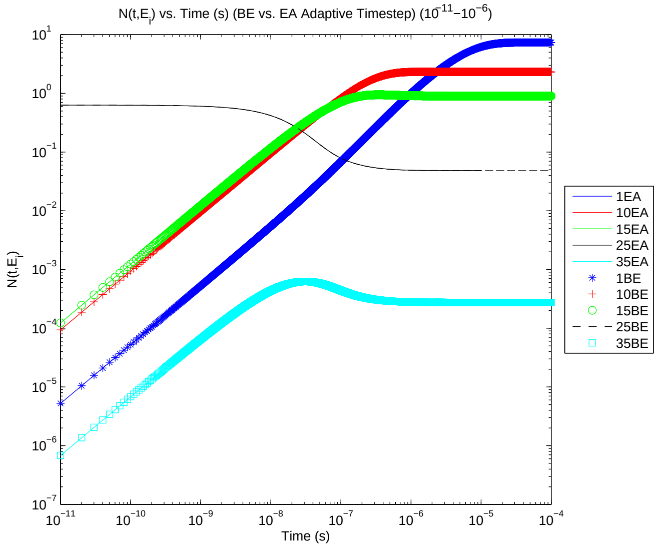


Figure 30: Explicit Asymptotic approximation (EA) vs. Backward Euler method (BE): Numerical solutions of both methods showing Neutrino density vs. Time at a varying time step of 10^{-11} to 10^{-6} seconds.

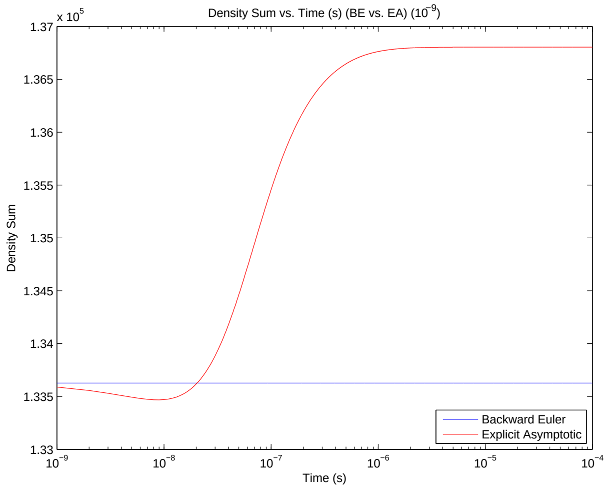


Figure 31: Explicit Asymptotic approximation (EA) vs. Backward Euler method (BE): Total amount of neutrino particles vs. Time for both methods at time step of 10^{-9} seconds.

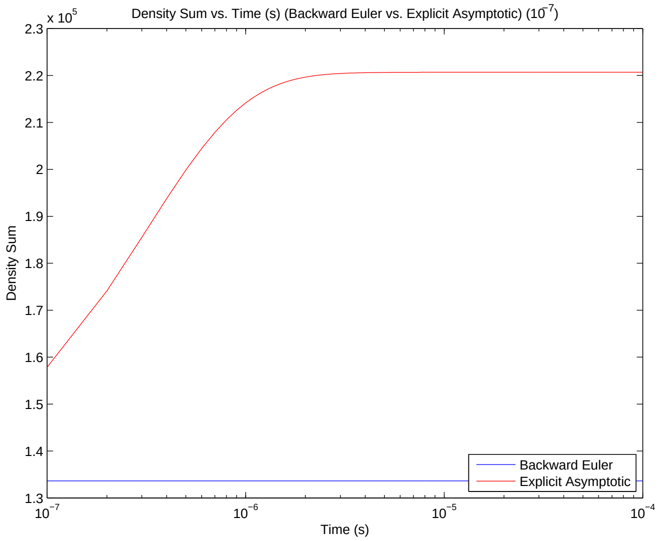


Figure 32: Explicit Asymptotic approximation (EA) vs. Backward Euler method (BE): Total amount of neutrino particles vs. Time for both methods at time step of 10^{-7} seconds.

Acknowledgment

I would like to express profound gratitude to DR. EIRIK ENDEVE, who instructed and guided me on an immensely interesting subject I may have never discovered otherwise. His elaborate explanations on the material made meetings engaging and encouraged me on the upcoming work week. His generous patience during my numerous moments of misunderstanding was highly appreciated as well. Mostly, thank you for letting me come on the team and contribute in this project.

I would also like to sincerely thank DR. MIKE GUIDRY, for allowing me to come on the project and contribute to the great work being completed. His role as my faculty adviser during this thesis project couldn't have been filled by a more knowledgeable or helpful individual. His masterful knowledge on document formatting and editing was tremendously helpful. Finally, for assisting Eirik in helping me attempt to learn this interesting but confusing material, thank you.

Lastly, I must express my deepest appreciation to my FAMILY and GIRLFRIEND, for understanding and allowing my long work hours. Without your support, I wouldn't remotely have the thesis I have today. Thank you for accepting this daunting commitment as the terrific opportunity it was.

REFERENCES

[1] M. YANO, J. D. PENN, G. KONIDARIS, AND A. T. PATERA, *Math, Numerics, and Programming (for Mechanical Engineers)* (The Authors, n.d.), 2nd ed.

[2] D. A. JONES (unpublished).

[3] S. AMEN, P. BILOKON, A. CODD, M. FOFARIA, T. SHAH, AND J. CASH, *Numerical Solutions of Differential Equations* (Academic, London, 2004).

[4] T. BUI, B.S. Honors thesis, University of Connecticut, 2010.

[5] E. ENDEVE AND M. GUIDRY, *Explicit Integration of Neutrino-Electron Scattering in Dense Nuclear Matter*, (unpublished Report),(2016), pp. 1-7.

[6] M. GUIDRY, *Algebraic stabilization of explicit numerical integration for extremely stiff reaction networks*, Journal of Computational Physics, 231 (2012), pp. 1-5.