Comparsion of Two Methods for Modeling Electron-Radiation Interactions

Jennifer B. Webster
UG Student at Los Alamos National Laboratory, CCS-2
University of California - San Diego, Applied Math

Undergraduate Honors Thesis, Sumbitted June 2008

June 2, 2008

Advisors: Michael J. Holst, UCSD Mathematics Department
Daniel Tartakovsky, UCSD Mechanical and Aerospace Engineering Department
Marv Alme, Los Alamos National Laboratory, X-3
Robert Webster, Los Alamos National Laboratory, X-3
Ben Bergen, Los Alamos National Laboratory, CCS-2

Contents

List of Figures

List of Tables

1 Introduction

At high temperatures, radiation-electron interactions are dominated by a process called Compton Scattering. This type of interaction is important for high energy physics problems, such as modeling stars, and is commonly modeled using a Fokker-Planck. There are many different ways to solve a Fokker-Planck equation, including an analysis using Green's Theorem as well as a variety of finite differencing schemes.

This paper will examine two of the finite differencing schemes. The first is a method published in 1969 by Chang and Cooper of Lawrence Livermore National Laboratory and the second is described in a 1984 Paper by Larsen et Al [4] [7]. First, we will look into how the Fokker-Planck equation compares to the differential equation describing a full system of radiation transport, called the Kompaneets Equation, and what the process of Compton Scattering entails. The following two sections will describe each method and provide some of the details of problems and pitfalls encountered in this implementation. Then, the next section gives a brief overview of the code architecture and some programming challenges encountered. The fifth section will discuss several test problems and comparison of the two methods and the last is a section giving the some general conclusions of this project and ideas for future research.

2 Background

Models of Radiation transport problems involve predicting several different types of interactions between radiation and matter that occur simultaneously. In a high energy system, the interactions of the ions, electrons and photons exhibit characteristics of both classic and relativistic physics. This paper considers a system in which the ion-ion, ion-electron and electron-electron interactions are negligible, as in a Maxwellian gas of free electrons, and ignores relativistic effects.

Current models of radiation transport have used different approximations of the Kompaneets Equation, also called the transport equation, to model this situation. This equation is given as

$$\frac{\partial \bar{n}}{\partial t} = \frac{n_e \sigma_T c}{m_e c^2} \frac{1}{\epsilon^2} \frac{\partial}{\partial \epsilon} \left[\epsilon^4 \left(\bar{n} + \bar{n}^2 + k T_e \frac{\partial \bar{n}}{\partial \epsilon} \right) \right] + \nabla \cdot \left(\frac{c}{3 n_e \sigma_T} \nabla \bar{n} \right) + j \tag{2.1}$$

where the variables are as follows: $\bar{n}\left(\epsilon, \mathbf{r}, t\right)$ is the photon occupation number, which can also be seen as a distribution function of the photons at different energy levels, ϵ, at specific points in space, $\mathbf{r}$, at time t. Other constants in this equation are the speed of light, c, Boltzmann's constant, k, and the rest energy of an electron, $m_e c^2$. Specific to a given problem are the electron temperature, T_e, the number of electrons in the system, n_e, and the Thomson cross-section, σ_T [6] [1].

The right hand side of this equation can be described as representing the three main phenomenon that the photons in this system can experience. The first term represents the scattering effect known as Compton Scattering, which will be examined in thie paper. The second term accounts for spatial diffusion, and the last is an independent source term that either removes or introduces photons to our environment. Unfortunately, since all of these processes are occurring at simultaneously, solving for the full transport solution is challenging. "Opperator splitting" methods are commonly used to break this problem into simpler problems and then recombine these answers to obtain the actual solution. Two such methods are analyzed in this paper. These methods deal with a system in which there is an infinite medium in which the energy distribution is the same at every point. Also, assume that the number of photons in our system is conserved, but that the total energy is not. For the Kompaneets equation, this means that the second and third terms do not contribute anything to the system, giving an approximation in the form of a Fokker-Planck equation:

$$\frac{\partial \bar{n}}{\partial t} = \frac{n_e \sigma_T c}{m_e c^2} \frac{1}{\epsilon^2} \frac{\partial}{\partial \epsilon} \left[\epsilon^4 \left(\bar{n} + \bar{n}^2 + k T_e \frac{\partial \bar{n}}{\partial \epsilon} \right) \right] \tag{2.2}$$

where $t \in [0, t_0]$ and $x \in [0, \infty)$. This describes the process known as either Compton or inverse Compton scattering depending on how a scientist defines the problem. These are interactions between photons and electrons describe situations where photons collide with electrons and as a result produce inelastic collisions, so there is a transfer of energy between photon and electron as well as a change in direction and velocity of the members of the interactions (see Fig 1). Compton Scattering refers to collisions where energy is transfered from the photons to the electrons and inverse Compton refers to those where it is transfered from the electrons to the photons known as Radiation Cooling and Heating, respectively ([11], 195-223).

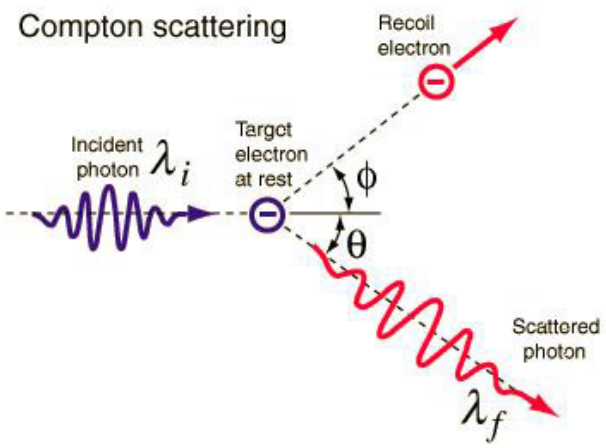


Figure 1: Example of Compton Scattering - [9]

3 The Chang and Cooper Method

3.1 The Discretizations in the Abstract

This algorithm comes from one of the earliest and most widely cited papers on the subject of Compton Scattering, so it provides a useful first study of methodology in solving this problem. The analysis and development of the general algorithm discussed in this paper is written for a linear differential equation, rather than the Fokker-Planck Equation, which is non-linear. The general form of the linear problem is given as:

$$\frac{\partial u}{\partial t} = \frac{1}{A(x)} \frac{\partial}{\partial x} \left[B(x,t)u + C(x,t) \frac{\partial u}{\partial x} \right] \tag{3.1}$$

Chang and Cooper do all of their analytical work with this linear method and it is only in the very last section of their paper where they discuss a single nonlinear application, the Compton Scattering problem. They do not make any statements about the error of this method in either the linear form or when it is applied to the non-linear Fokker-Planck Equation. Chang and Cooper do provide some convergence criterion for this method. First, this method is only defined on the set of time values between 0.0 a characteristic equilibrium time, t_{Eq}. The user must calculate this time before setting the final time in the input deck to ensure that it meets this criteria. If the final time entered is much longer than t_{Eq}, it is possible that the problem will advance beyond the equillibrium solution and eventually give a photon distribution that is not descriptive of the actual physical situation. This could be demonstrated by the production of a distribution with negative numbers of photons present at a given energy or one with enormous values at random intervals. Therefore, an approximation for the characteristic time of decay, T_{Eq}, is given by the following equation

$$t_{Eq} = \left(\frac{l^2}{\lambda_p c} \right) \tag{3.2}$$

where λ_p is the mean free path or average distance a photon can travel before collision, l is the size of the sampling range and c is the speed of light. An additional convergence criteria is required by this method in setting up the maximum step size for the energy mesh. This is that the maximum distance between between two sampling points in the energy mesh, Δk_j, must satisfy the following equation:

$$\Delta k_j \ll \frac{1}{2} \frac{C^n_{j+1/2}}{B^n_{j+1/2}} = \frac{\theta_e}{1 + f^n_{j+1/2}} \tag{3.3}$$

When Chang and Cooper applied their abstracted method to the Fokker-Planck equation given in the form:

$$\frac{\partial f}{\partial t} = \frac{\sigma_0 \rho_e}{k^2} \frac{\partial}{\partial k} \left[k^4 \left[\theta_e(t) \frac{\partial f}{\partial k} + f + f^2 \right] \right] \tag{3.4}$$

where $f = ((hc)^3/2) f_p(k,t)$, $f_p(k,t)$ is the non-equilibrium photon distribution, k is the energy of the photons in keV, $\theta_e(t)$ is the electron temperature at time t, and ρ_e is the electron density. To gain the correct scaling from initial energy distirbutions, scaling functions were used in this implementation (see Sec. (5.3)). They have discretized it as

$$\frac{f_j^{n+1} - f_j^n}{\Delta t} = \frac{\alpha}{k_j^2} \frac{1}{\Delta k_j} \left[k_{j+1/2}^4 F_{j+1/2}^{n+1} - k_{j-1/2}^4 F_{j-1/2}^{n+1} \right] \tag{3.5}$$

where the following relations are defined:

$$\begin{aligned}
\alpha &= \sigma_T n_e c && (3.6)\\
\Delta k_{j+1/2} &= k_{j+1} - k_j && (3.7)\\
\Delta k_j &= \frac{\Delta k_{j+1/2} - \Delta k_{j-1/2}}{2} && (3.8)\\
F_{j+1/2}^{n+1} &= \frac{\theta_{j+1}^{n+1/2}}{\Delta k_{j+1/2}} \left(f_{j+1}^{n+1} - f_j^{n+1} \right) + (1+f)_{j+1/2}^n f_{j+1/2}^{n+1} && (3.9)\\
f_{j+1/2}^n &= (1-\delta_j^{n+1}) f_{j+1}^n + \delta_j^{n+1} f_j^n && (3.10)\\
f_{j+1/2}^{n+1} &= (1-\delta_{j+1}^{n+1}) f_{j+1}^{n+1} + \delta_j^{n+1} f_j^{n+1} && (3.11)
\end{aligned}$$

The δ_j^{n+1} terms are found by solving the following quadratic equation

$$\frac{\theta_e^{n+1/2}}{\Delta k_{j+1/2}} \left(f_{j+1}^e - f_j^e \right) + \left[1 + (1-\delta_j^{n+1}) f_{j+1}^e + \delta_j^{n+1} f_j^e \right] \left[(1-\delta_j^{n+1}) f_{j+1}^e + \delta_j^{n+1} f_j^e \right] = 0 \tag{3.12}$$

where $f_j^e = [C \cdot exp[k_j/\theta_e(t)] - 1]^{-1}$. In addition, the value for any δ_j^{n+1} term will be between 0 and $1/2$ and the constant C is the Fugacity, which is defined by means of an infinite sum (see Sec. (3.3) below).

3.2 This Implementation of the Procedure

This is a semi-implicit method - meaning that data from the previous time step is used in calculating the next iteration of the distribution. This method also uses constants derived from the assumed equilibrium (Planckian) distribution that would occur with the number of photons given in the system, δ_j^{n+1}. It is solved in this instance by creating a tridiagonal matrix containing the coefficients to be multiplied by the photon distribution function that we are sampling at the next time we wish to know something about, f_j^{n+1}.

This discretization is implemented in the stationary class called *ChangCooper*. This class contains three internal, or private, functions and one function called upon by other programs to implement all the others. When this method is initiated, the FP_init function checks to make sure that all of the vectors used to implement the method are the correct size and retrieves all of the constants that are needed in the process. Then it calls the functions which calculate the Fugacity. This involves calculating a dimensionless ratio of the actual number of photons to the number that would be present in a perfect black body at the current electron temperature and then using a polynomial approximation of the infinite sum that defines C^{-1}. The polynomial allows for a quicker calculation of the value (see Sec. 3.3 below).

The next step in the process is completed by the "nu_coeff_update" function, which is responsible for calculating the values of the δ_j weighting coefficients, the centered and boundary values of Δk and other standard relations necessary to the problem. The Δk values are computed by simply using the definitions described above in Eqns. (3.7) and (3.8). Also computed at this point is the vector of Diffusion constants

$$K_{j+1/2} = \frac{\Delta t \, \sigma_T \, \rho_e \, c}{m_e \, k_{j+1/2}^2 \, \Delta k_{j+1/2}} \tag{3.13}$$

This vector helps to keep the lines of codes used in itializing the tridiagonal matrix clean. It corresponds to the $(\alpha)(k_j^2 \Delta k_j)^{-1}$ portion of Eq. (3.5). The computation of the weighting coefficients, δ_j, is achieved by solving the quadratic equation:

$$(f_{i+1}^e - f_i^e)^2 \delta_j^2 + \left[-(f_{i+1}^e - f_i^e) - 2*(f^e)_{i+1}^2 + 2*(f^e)_{i+1}*(f^e)_i \right] \delta_j + \frac{\theta_e}{\Delta k_{i+1}} * (f_{i+1}^e - f_i^e) + f_{i+1}^e (1 + f_{i+1}^e) = 0 \tag{3.14}$$

This is done by first, calculating the values of the quasi-equilibrium solution f^e. Then, the standard quadratic formula gives:

$$\begin{aligned}
\delta_j &= \frac{-b \pm \sqrt{b^2 - 4ac}}{2} \\
a &= (f^e_{i+1} - f^e_i)^2 \\
b &= \left[-(f^e_{i+1} - f^e_i) - 2 * (f^e)^2_{i+1} + 2 * (f^e)_{i+1} * (f^e)_i\right] \\
c &= \frac{\theta_e}{\Delta k_{i+1}} * (f^e_{i+1} - f^e_i) + f^e_{i+1}(1 + f^e_{i+1})
\end{aligned} \tag{3.15}$$

These values are then checked for two requirements. First, the discriminant, $b^2 - 4ac$, is checked to make sure that it is positive and give a defined square root. If not, the program prints an error message to the screen and exits the program because no further useful information can be derived by continuing. The second thing the program checks is whether the value stored in the δ_j vector is between 0 and 0.5. This is done by first trying the positive square root of the determinant and then, if that value does not give something in the correct range, it uses the negative value. Once these weighting factors are found, the "half point values," $f^n_{j+1/2}$, are calculated according to Eq. (3.10) and are stored in the vector p to calculate the vectors listed in Eq. (3.16).

After all of these auxiliary values have been computed, they are passed back to the FP_init function, which uses the values to create three vectors corresponding to the main three diagonals in the matrix. Through substituting the definitions in Eq. (3.6) in Eq. (3.5), the following formulas can be used to define the coefficients needed to solve the linear system for time advancing the vector:

$$\begin{aligned}
L_j &= -K_{j+1/2} * \left(-k^4_j * \left(\frac{-\theta_e}{\Delta k_j} + p_j * \delta_j\right)\right) \\
D_j &= 1.0 - K_{j+1/2} * \left(k^4_{j+1} * \left(\frac{-\theta_e}{\Delta k_{j+1}} + p_{j+1} * \delta_{j+1}\right) - k^4_j * \left(\frac{-\theta_e}{\Delta k_j} + p_j * (1.0 - \delta_j)\right)\right) \\
U_j &= -K_{j+1/2} * k^4_{j+1} * \left(\frac{-\theta_e}{\Delta k_{j+1}} + p_{j+1} * (1.0 - \delta_{j+1})\right)
\end{aligned} \tag{3.16}$$

where the vector U contains the entries stored immediately to the right of the main diagonal, D stores entries of the main diagonal, and L are the values to the left of the main diagonal. Physically, the use of a Tridiagonal Matrix implies that the number of photons at a given sampling point is effected by the number of photons at the sampling points on either side of it. The boundary condition requiring that there is no net flow of photons at either the minimum or the maximum sampling point does change these vectors slightly in that L_0 and U_{m-1} values are set to zero and some small changes are made to the value of the main diagonal at the first and last p;oints to account for this.

The Tridiagonal Matrix Solver then takes these three vectors and solves the standard equation $Ax = b$, where A is the $m \times m$ matrix

$$A = \begin{pmatrix}
D_0 & U_0 & 0 & \cdots & 0 \\
L_1 & D_1 & U_1 & & \vdots \\
0 & \ddots & \ddots & \ddots & 0 \\
\vdots & & L_{m-2} & D_{m-2} & U_{m-2} \\
0 & \cdots & 0 & L_{m-1} & D_{m-1}
\end{pmatrix} \tag{3.17}$$

and b is the vector given by the photon distribution function at the currently known time step, f^n_j, and x is the photon distribution at the next desired step, f^{n+1}_j. As in the rest of the problem, f^n_0 and f^{n+1}_0 are the values at the minimum sampling point in the mesh. The tridiagonal matrix is then solved by Gaussian Elimination without partial pivoting. The interface between this Matrix Solving algorithm and the main portion of the update program has been standardized to allow for future changes to this method by only requiring that the three vectors containing the coefficients and the b vector be passed to the program.

3.3 Challenges - Finding the Fugacity

3.3.1 Analysis and Algebraic Manipulation

In this method, the quasi-equilibrium solution f^e must be computed at every sampling time in order to calculate a weighting function needed to determine how much energy is moved between neighboring bins. This is given by

$$f^e(k,t) = \frac{1}{Ce^{k/\theta_e(t)} - 1} \tag{3.18}$$

where C is the fugacity to be determined for each time step, k is the energy of the photon in keV and $\theta_e(t)$ is the electron temperature at time t in keV. Fugacity is an adjusted form of pressure that is related to the chemical potential of the system. It describes the likelihood of the to change. When $C = 1.0$, this fugacity means that the system is in the Planckian equilibrium solution and, therefore, will not change. The fugacity of the system is related to the total number of photons, N, and the electron temperature, θ_e. The total number of photons in the system is expressed in Chang and Cooper [4, 14] as:

$$N(t) = 4\pi \int_0^\infty k^2 f(k,t) dk \tag{3.19}$$

The distribution, $f(k,t)$, in Chang and Cooper's paper was non-dimensional, so to correct the dimension Eq. (3.19) is multiplied by a factor of $\frac{2}{(hc)^3}$ (see Eq. (5.4)), giving the equation:

$$N(t) = \frac{8\pi}{(hc)^3} \int_0^\infty k^2 f(k,t) dk \tag{3.20}$$

where h is Planck's constant in keV·s, c is the speed of light in cm/s, k is the energy of the photon in keV (sometimes written as $h\nu$, with ν being the frequency of the photon) and $f(k,t)$ is the photon distribution function. Since the number of photons in the system is conserved, the number of photons given by the proposed equilibrium solution must be equal to this value. Substituting Eq. (3.18) into Eq. (3.20) then requires:

$$N(t) = \frac{8\pi}{(hc)^3} \int_0^\infty \frac{k^2}{Ce^{\frac{k}{\theta_e(t)}} - 1} dk \tag{3.21}$$

Performing a change of variables where $\alpha \equiv \frac{k}{\theta_e(t)}$ gives:

$$N(t) = \frac{8\pi\theta_e^3(t)}{(hc)^3} \int_0^\infty \frac{\alpha^2}{Ce^\alpha - 1} d\alpha \tag{3.22}$$

Since

$$\frac{1}{Ce^\alpha - 1} = C^{-1}e^{-\alpha} * \frac{1}{1 - Ce^\alpha} \tag{3.23}$$

a Power Series Expansion can be performed to give:

$$\frac{1}{Ce^\alpha - 1} = \frac{e^{-\alpha}}{C}\left[1 + \frac{e^{-\alpha}}{C} + \left(\frac{e^{-\alpha}}{C}\right)^2 + ...\right] \tag{3.24}$$

$$= \sum_{n=1}^{\infty} \left(\frac{e^{-\alpha}}{C}\right)^n \tag{3.25}$$

Then substituting Eq. (3.25) back into Eq. (3.22), the number of photons can be expressed as:

$$N(t) = \frac{8\pi\theta_e^3(t)}{(hc)^3} \int_0^\infty \alpha^2 \sum_{n=1}^{\infty} \left(\frac{e^{-\alpha}}{C}\right)^n d\alpha \tag{3.26}$$

$$= \frac{8\pi\theta_e^3(t)}{(hc)^3} \sum_{n=1}^{\infty} C^{-n} \int_0^\infty \alpha^2 e^{-n\alpha} d\alpha \tag{3.27}$$

Performing integration by parts on $\int_0^\infty \alpha^2 e^{-n\alpha} d\alpha$ treating n as a constant shows:

$$\int_0^\infty \alpha^2 e^{-n\alpha} d\alpha = \frac{-e^{-n\alpha}}{C^n n^3} \left[\alpha^2 n^2 + 2n\alpha + 2\right] \Big|_0^\infty \tag{3.28}$$

$$= \frac{2}{C^n n^3} \tag{3.29}$$

Performing the substitution given in Eq. (3.29) in Eq. (3.27) gives:

$$N(t) = \frac{8\pi}{(hc)^3} \sum_{n=1}^{\infty} \frac{2C^{-n}}{n^3} \tag{3.30}$$

$$= \frac{16\pi}{(hc)^3} \sum_{n=1}^{\infty} \frac{C^{-n}}{n^3} \tag{3.31}$$

From basic analysis on $\sum_{n=1}^{\infty} \frac{C^{-n}}{n^3}$, $C \geq 1$ is required to guarantee convergence of the sum.

3.3.2 Early Methods used to Compute the Fugactiy

In the Chang and Cooper algorithm, N is determined by integrating the distribution curve f over all energies of photons, ν. While this value is currently conserved by the program, it is possible that the electron temperature, θ_e, could change with time, causing the fugacity to be recomputed at each step. Several methods to compute this value are discussed below.

Newton Iteration In the initial phase of calculating fugacity, Newton iteration was used, where

$$C_i = \left(\frac{N(hc)^3}{16\pi\theta_e^3} - \sum_{n=2}^{\infty} \frac{C_{i-1}^{-n}}{n^3} \right)^{-1} \tag{3.32}$$

The infinite sum was approximated by summing the first 100 iterations of the sum using $C_0 = 1$ as the initial guess. This is computationally expensive, which quickly becomes a problem with small time steps between calculations since it must be recalculated for every sample time t.

Brent's Method The next method tried was Brent's Method for root finding. This method uses a combination of bisection, secant method, and quadratic interpolation to find the root C of the equation

$$g(C) = N - \frac{16.0\pi\theta_e^3}{(hc)^3} \sum_{n=1}^{\infty} \frac{C^{-n}}{n^3} \tag{3.33}$$

Again, the infinite sum was approximated by the first 100 terms. This method proved quicker and computationally cheaper than the Newton iterates previously used, but for the size of the computations eventually desired, this method (especially the summation term) is still too expensive computationally.

3.3.3 The Ratio Solution

Given the number of photons in the system, N, and the electron temperature at the time t, $\theta_e(t)$, a dimensionless ratio, $\widetilde{N}$, can be found by defining

$$\widetilde{N} \equiv \frac{N}{\frac{16\pi\theta_e^3}{(hc)^3} \sum_{n=1}^{\infty} \frac{1}{n^3}} \tag{3.34}$$

This gives us a value for $\widetilde{N}$ that is between 0 and 1 for all possible values N and $\theta_e(t)$. Computationally, this is very inexpensive because $\sum_{n=1}^{\infty} \frac{1}{n^3} = \zeta(3)$, which is Apéry's constant and has been previously calculated beyond machine precision, can be set as a constant in our calculations, hereafter denoted by A. Also, $P \equiv C^{-1}$. Since C is of the range [1, ∞), this means that P has a value in [0, 1], where $P = 1$ gives the Black Body spectrum.

A one time application of Brent's Method to the ratio of $\frac{P}{\widetilde{N}}$ gives the values of $\widetilde{N}$ as a function of values of P. Given these values, several polynomial approximations to this curve were attempted. First, a Chebyshev polynomial was proposed as a way to approximated the solution, but this was very inaccurate at values close to $P = 1$, the Black Body Spectrum, which is an important checkpoint for Compton Scattering problems. In order to decrease the error close to this value, a Hermite Interpolating Polynomial of the following form was found to be more accurate at values at near 1.0:

$$\begin{aligned}\frac{P}{\widetilde{N}} =&(1.0-\widetilde{N})*(A-1.0)+1.0+(0.0+(\widetilde{N})*(2.08679886936000009e-02+\\ &(\widetilde{N}-\frac{1}{18})*(-1.09082464452000087e-02+\\ &(\widetilde{N}-\frac{1}{9})*(-5.46383614319995853e-03+\\ &(\widetilde{N}-\frac{1}{6})*(-3.47464815480006631e-03+\\ &(\widetilde{N}-\frac{1}{3})*(-3.29539334502862435e-03+\\ &(\widetilde{N}-\frac{1}{2})*(-2.79493951062817318e-03+\\ &(\widetilde{N}-\frac{2}{3})*(-9.95466439332460401e-03+\\ &(\widetilde{N}-\frac{5}{6})*(-2.00898111631381202e-02+\\ &(\widetilde{N}-\frac{21}{24})*(-1.04187874273091952e-01+\\ &(\widetilde{N}-\frac{11}{12})*(-6.54801131638733436e-01+\\ &(\widetilde{N}-\frac{23}{24})*(-1.21173685911828830e+01)))))))))))\end{aligned} \tag{3.35}$$

Using this polynomial, this polynomial fit the curve produced by the Brent's method, a direct (and computationally expensive) calculation, with extremely restrictive tolerances on finding the roots to within 10^{-5} at all points along the curve. By using the dimensionless ratios of $\widetilde{N}$ and $\frac{P}{\widetilde{N}}$, the fugacity can now be found by evaluating Eq. (3.35) at $\widetilde{N}$ with a cost of approximately 35 mathematical operations, which is far cheaper that evaluating the sum (3.25) each time we need to calculate C. Using this polynomial to find values of C, Eq. (3.18) can be evaluated with much less preparation time, which is important as more complex problems are evaluated.

3.3.4 Graphical Analysis

The two graphs below are the result of running the code built to implement both the polynomial approximation and Brent's method. Figure 2(a) one shows a plot of our dimensionless ratio $\frac{P}{\widetilde{N}}$ versus the values for P calculated by the polynomial. Figure 2(b) shows the error between the calculated values, which is always within 2.0×10^{-5} or 0.002% of the desired value produced through Brent's Method.

4 The Larsen Method

4.1 The Discretization in the Abstract

The paper by Larsen, et Al. is different from that of Chang and Cooper mentioned above (3) in that it treats a version of the Fokker-Planck equation given as

$$\frac{\partial u(x,t)}{\partial t} = \frac{1}{A(x)}\frac{\partial}{\partial x}\left[B(x,t,u)u(x,t) + C(x,t,u)\frac{\partial u(x,t)}{\partial x}\right] \tag{4.1}$$

which has the potential to be a nonlinear function of $u(x,t)$, the photon distribution function in a Compton Scattering Problem in the course of their analysis. Larsen and his colleagues rewrite Eq. (4.1) in two different ways with corresponding discretizations. One of them produces a similar discretization to Chang and Cooper's method, as well as a discretization known as Young's method. The other discretization is given through manipulation of Eq. (4.1) in the following form:

$$\frac{\partial u(x,t)}{\partial t} = \frac{1}{A(x)}\frac{\partial}{\partial x}\left[F\frac{\partial}{\partial(e^{\rho})}\left[e^{\rho}Gu\right]\right] \tag{4.2}$$

They have discretized this Equation with a method called the modified Young's Method, which is given as follows:

$$\frac{u_j^{n+1}-u_j^n}{t^{n+1}-t^n} = \frac{1}{A_j}\left[\frac{S_{j+1/2}^{n+1}-S_{j-1/2}^{n+1}}{x_{j+1/2}-x_{j-1/2}}\right] \tag{4.3}$$

where

$$\begin{aligned} S_{j+1/2}^{n+1} &= 0, \qquad j=0,J \\ &= F(x_{j+1/2},t^{n+1},u_{j+1/2}^{*})\times \\ &\qquad \left[\frac{e^{\rho(x_{j+1},t^{n+1})}G(x_{j+1},t^{n+1},u_{j+1}^{*})u_{j+1}^{n+1}-e^{\rho(x_j,t^{n+1})}G(x_j,t^{n+1},u_j^{*})u_j^{n+1}}{e^{\rho(x_{j+1},t^{n+1})}-e^{\rho(x_j,t^{n+1})}}\right], \\ &\qquad j=1,...,J-1 \end{aligned} \tag{4.4}$$

The following intermediate steps are necessary to define the problem specific to radiation transport, as the above discretization and the first half of the paper are written in dealing with the generic problem given in Eq. (4.1).

$$\begin{aligned} A_j^{-1} &= \sigma_s x_j \\ F(x_{j+1/2},u_{j+1/2}^{*}) &= \left\{x_{j+1/2}^2+\frac{\gamma}{2}\left[\frac{u_{j+1}^{*}}{x_{j+1}}+\frac{u_j^{*}}{x_j}\right]\right\}^2 \\ \rho(x_j) &= \frac{x_j}{\theta_L} \\ G(x_j,u_j^{*}) &= \left(x_j^3+\gamma u_j^{*}\right)^{-1} \end{aligned} \tag{4.5}$$

4.2 This Implementation of the Procedure

This method posed some interesting challenges not present in the first method simply because it was implemented second, which means that it must work within the framework established by the implementation of the Chang and Cooper method in order to have any meaningful comparisons. Therefore, the first step is to make sure that the scaling between the two methods is correct. Fortunately, it is relatively easy to perform such a check as Larsen provided Chang and Cooper's method in their paper for comparison in terms of their definitions of the sampling space:

$$\frac{u_j^{n+1}-u_j^n}{t^{n+1}-t^n} = \sigma_s x_j\left[\frac{T_{j+1/2}^{n+1}-T_{j-1/2}^{n+1}}{x_{j+1/2}-x_{j-1/2}}\right] \tag{4.6}$$

$$\begin{aligned} T_{j+1/2}^{n+1} &= 0, \qquad j=0,J \\ &= x_{j+1/2}^4\{\theta(x_{j+1}-x_j)^{-1}\left[\frac{u_{j+1}^{n+1}}{x_{j+1}^3}-\frac{u_j^{n+1}}{x_j^3}\right] \\ &\qquad +\left[1+\gamma\left((1-\delta_j^{*})\frac{u_{j+1}^{*}}{x_{j+1}^3}+\delta_j^{*}\frac{u_j^{*}}{x_j^3}\right)\right]\cdot\left[(1-\delta_j^{*})\frac{u_{j+1}^{n+1}}{x_{j+1}^3}+\delta_j^{*}\frac{u_j^{n+1}}{x_j^3}\right]\}, \\ &\qquad j=1,...,J-1 \end{aligned} \tag{4.7}$$

Through term by term comparison of Eq. (3.5) and the supplementary definitions with the following relations [[7], 373], the following conclusions are reached:

1. x and θ in these equations have been changed to dimensionless units that are ratios of the dimensioned values and the rest energy of an electron, m_e. So the following supplimentary definitions are necessary:

   (a) $x = k/m_e$

   (b) $\theta = \theta_e/m_e$

2. The photon distribution functions of the two methods have also been scaled and are represented by the linear transformation $u_j^* = f_j^* x_j^3$, so this conversion is accomplished by a one of the global conversion functions, discussed in Sec. (5.3).

3. The Thomson scattering cross section in Larsen's paper, σ_s, is equivalent to the Thomson cross section of Chang and Cooper's paper with the additional factor of the electron number density, ρ_e. So we make the following replacement: $\sigma_s = \sigma_T \rho_e$

4. γ is a constant that is found using the fact that the Planckian Distribution can be written as

$$\begin{aligned} B(x,t) &= \frac{\gamma^{-1} x^3}{e^{x/\theta} - 1} \\ &= \frac{\gamma^{-1} \nu^3}{m_e^3 [e^{\nu/\theta_e} - 1]} \\ &= \frac{8\pi\nu^3}{(c\,h)^3 [e^{\nu/\theta_e} - 1]} \end{aligned} \tag{4.8}$$

   where the last line shows the Planckian Distribution in energy that has become standardized in introductory texts on radiation transport. Therefore, γ can be defined as:

$$\gamma = \left(\frac{c\,h}{m_e}\right)^3 [8\pi]^{-1} \tag{4.9}$$

Taking into account these scaling factors, the following semi-implicit discretization is actually implemented in the program:

$$\begin{aligned} A_j^{-1} &= c\,\sigma_T \rho_e x_j \\ F(x_{j+1/2}, u_{j+1/2}^n) &= \left\{ x_{j+1/2}^2 + \frac{\gamma}{2}\left[\frac{u_{j+1}^n}{x_{j+1}} + \frac{u_j^n}{x_j}\right] \right\}^2 \\ \rho(x_j) &= e^{x_j\ \theta^{-1}} \\ G(x_j, u_j^n) &= \left(x_j^3 + \gamma u_j^n\right)^{-1} \end{aligned} \tag{4.10}$$

These intermediate values are then used to set up the three main diagonals in the following manner

$$\begin{aligned} k_j &= \frac{\Delta t A_j^{-1}}{\Delta x_{j+1/2}} \\ U_j &= k\,F_{j+1/2} \frac{\rho(x_{j+1}) G_{j+1}}{\rho(x_{j+1}) - \rho(x_j)} \\ D_j &= -1.0 - k\,[\rho(x_j)\,G_j] \left[\frac{F_{j+1/2}}{\rho(x_{j+1}) - \rho(x_j)} + \frac{F_{j-1/2}}{\rho(x_j) - \rho(x_{j-1})}\right] \\ L_j &= k\,F_{j-1/2} \frac{\rho(x_{j-1}) G_{j-1}}{\rho(x_j) - \rho(x_{j-1})} \\ b_j &= -f_j \end{aligned} \tag{4.11}$$

It should be noted that this discretization solves for $\partial u/\partial t$, so it must be rescaled to energy and number distribution for comparison with Chang and Cooper's method through the global scaling functions (Sec. (5.3).

5 Code Architecture

5.1 General Structure

This code is written in C++ and can be most easily classified as a modular-style, levelized program. It is contained in 18 files - the Makefile, 9 header files and 8 implementation files. There are stationary classes containing the processes (referenced by the numbers in Fig (3) that import and export the data (1 and 2), update the solution by one time step (3), and perform various types of analysis on the problem (4). There is also one movable object, the *Problem* class, which holds problem specific data in four subclasses - Setup Parameters, Control Variables, Problem Constants and 'state.' The next several sections will discuss these features in more detail.

5.1.1 Defining the Problem within the Code

Many variables are necessary to define this problem, as can be seen from the descriptions of the discretizations below (Sec. 3 and Sec. 4). Initially, all of these variables existed in one movable class called "state." As the problem became more and more complex, this single object became inefficient and hard to maintain. Therefore, this state was split into the four subclasses mentioned above. The *state* class is the only one of these classes that is changed over the course of the problem, but it is affected by the other subclasses during initial setup and during the time updates. In Figure 3, it is shown how the various stationary modules interact with one another and shows how the structure of the "Problem" is passed between them.

The *setup* class is the resting place for those variables that are really only used once in the problem because after the initial setup all of this information is included in other places. For example, in order to setup the initial sampling mesh, whether it is in energy or space, the program requires the user to specify a minimum and maximum sampling point, the number of places to sample at and the rule for how to set up such a grid. Once a function has been executed to do this, all of this information is recorded in the vector of the mesh as the first and last entries of the vector, the size of the vector, and the spacing between the rest of the sampling points, respectively. It does not make sense, computationally, to pass these variables around in duplicate. Therefore, they have been separated from the main portion of the problem.

The *Control* and *Problem Constants* classes were separated for ease in maintainenece. They contain variables that a user must define initially, but are not changed for the rest of the problem. The *Control* class holds user commands about how to update the problem - method used to advance or analyze the problem, boundary conditions desired, etc. The *Problem Constants* class contains data needed to advance the problem, but that is not a changing part of the problem, such as the Thompson Scattering cross section, absorption coefficients, and so forth.

Another portion of the *Problem* files is the definition of a namespace for world constants. These constants included the Planck, Stefan-Boltzmann, and Apery constants as well as the speed of light, pi and the rest energy of an electron. With such a large program and these values needed so often, it made more sense to define them once and avoid the problems created when these values were defined in different units/scales in different parts of the problem. For example, declaring the speed of light in meters versus centimeters per seconds would cause significant problems in the scaling of the problem. Since the problem class, and therefore header file, is accessed by all of the other parts of the program, it made the most sense to put this definition here.

5.1.2 The Stationary Modules

The stationary modules, as mentioned above, are pre-defined sets of instructions that a user can choose to help analyze the problem of radiation hydrodynamics. Each of the three is a separate entity which interacts with the appropriate portions of the *Problem* module to that end. They are all governed by the main.cc file, which calls upon the various functions to time advance the problem.

The Update Module This is the module that the rest of the paper is concerned with, so it will only be described briefly here. This is contained in 5 different files - two to define the class, the functions that enforce boundary conditions and the time step limit known as a Courant check (5.1.2), and the matrix solving package; two to define the Larsen and Chang-Cooper Methods; and one to describe the ways of solving the Diffusion Equation.

The Boundary Conditions can be set to either Dirichlet or Neumann conditions. The Dirichlet boundary conditions require that the distribution function of the photons is precisely equal to some predefined boundary value, like zero, at either endpoint. Neumann conditions required the Flux, or first derivative, through the boundary to be zero, so it is approximated by requiring that the first two and last two entries are the same value. The functions defining values

along the boundary can fluctuate as the time advances, so they are updated with each time step before preforming an update on our distribution function. Currently, this choice of boundary conditions is only implemented for the Diffusion solver, not the Fokker-Planck equation. Both of the algorithms discussed in this paper implement their own boundary conditions in that they require their "Flux" at either boundary to be zero, which is hard wired into the code.

The Courant Condition, also known as the CFL Condition, is a control function instigated to take care of a numerical instability that arises when using an explicit scheme to time advance the problem, which was first described in a paper by Courant, Friedrichs and Lewy in 1928. This time control is currently only used in the program, when the user is solving the simple diffusion problem, and not the Fokker-Planck Equation. Numerical stability is more likely, but not guaranteed because there are some unusual cases in which the solution is not possible, when the following equation is satisfied:

$$\Delta t < \frac{3 * (\Delta x)^2}{2 * c} \tag{5.1}$$

where Δt is the distance between the current time and the next sampling time, Δx is the smallest distance between points in the sampling mesh and c is the speed of light in cm per sec. When the time step is very large and/or the distance between sample mesh points is very small, advancing the solution function can exhibit some very odd characteristics including returning a negative value for the photon distribution function (see Fig. 5.1.2). In the problem below in Figure 5.1.2, we are solving a simple diffusion approximation given by the following equation

$$\frac{\partial f}{\partial t} = \nabla \cdot \left(\frac{c}{3 n_e \sigma_T} \nabla f \right) \tag{5.2}$$

with a starting curve of $y = 1 - x^2$ where $x \in [-1, 1]$ instead of the Fokker-Planck equation for the simplicity of calculation and illustration, but the same process holds true with the Fokker-Planck equation as well. In Figure 4(a), the initial time step is checked against the condition and if it is larger than the right hand side of Eq. 5.1, then the program resets the time step to be 0.8 of this value. As can be seen in a comparison of Figures 4(a) and 4(b), when the CFL condition is in effect, the problem shows a nice diffusion to zero energy, but without this condition, the answer given becomes non-sensical, ranging from an enormous negative number to an enormous positive number at either end of the sample space. This phenomenon is observable even after one attempt at time advancement, as is illustrated by Figure 4(c).

The Parser The parser is solely for reading input into this program and storing it in the appropriate variables. It contains four different maps or dictionaries - one for integers, one for doubles, one for character strings and one for functions. This means that a user can define any variable they introduce into the program with a generic register function and the program is smart enough to recognize the type. Then when the variable is encountered in the input file, the parser will check all of the available maps and store the input value as the appropriate type without having the user define a second level parsing function to convert it from a string. This makes the program much neater and more maintainable in the long run. In addition to this variable recognition, the *Parser* class, which is a static object, is initiated once and then every other time the program requires the parser, an instance of this class is created, which has access to all of the data previously stored in the parser without requiring that the class is passed between the functions. This is known as a Singleton Pattern. The *Matrixsolver* class described in the Update package also works this way. This class also holds the parsing abilities to restart the problem from a previous output file, which is important as problems get larger.

The Analysis Module This class is a suite of functions designed to evaluate the performance of this code. Some of the analysis functions that this class performs are:

1. *AnalyzeStep* to help check a single step for index errors
2. *CalculateError* that calculates the error from the diffusion at every point
3. *SinglePointAnalyze* that checks the rate of decay for a sine function against the appropriate exponential decay as well as the log of the data at a point
4. *TrapCalc* that calculates the area under the curve with the trapezoidal rule for plotting as well as the log of that value.

5. *Tracer* function which allows the user to track the movement and evolution of the error at a certain point in the spatial mesh when the program is run multiple times using different time steps. This should produce a straight line for both the implicit and explicit methods and a parabolic curve using the time centered method indicating linear convergence for the first two and quadratic convergence for the second.

The second major function of this class is the ability to print the output for user analysis. This *printout* has three sub-functions. First, it prints the f and x vectors to a file for later graphing. It also calls the Analyze function to printout the appropriate error analysis for the current state. Lastly, it creates a gnuplot command file which is called by the main program and plots 10 diffusion curves and/or error curves on one graph for ease in analysis. This printout function is called by the main program and is run at ten roughly equally spaced intervals between t0 and t1 (usually this input is read as 0.0 and the characteristic time).

5.2 Challenges of Dealing with a Large Program

In splitting the problem into so many modules and having it show so many different features, we had to be very concerned with levelization, that is making sure that we did not have files that reference each other, creating a recursive definition that causes an error in compiling. As the variables needed to define the problem grew in number, it was necessary to split the problem into four subclasses and this changed how the variables could interact with one another. In levelizing the program, some of the function calls needed to be rearranged to ensure that variables defining the "state" of the problem at any particular instance could be set up using rules stored in the control class as well as initial parameters from the Setup Problem Constants classes. Mostly, this requirement was satisfied by moving functions requiring information from more than one subcategory to the Problem class level (ie, one level higher than all of the other classes involved in the function). For example, to set up the initial energy mesh, the code needs to handle the rule for setting up the mesh (whether it is logarithmically scaled or linearly), the minimum and maximum energies the problem involves and the number of sampling points desired in addition to the vector which will contain the mesh for calculations. This means that a function to set this up must interact with three different subclasses of the Problem class. Therefore, the function must be written out in the Problem class, which is higher up in the hierarchy.

5.3 Changing the Space of the Problems

Another challenge for this project is that of defining the space in which work is done. Because this field does not have a standard termonolgy associated with it, people trained in different disciplines use language that they are most comfortable with to analyze this problem. Therefore, there each paper differs in defining the space of the problem. A standard equation is defined for the distribution of the energy as a function of the energy or frequency of the photons, $E(\nu, t)$, so it is logical to initialize the problem in this space. Once given this energy distribution, it is easy to convert to the number of photons, $N(\nu, t)$, from this by the following relation

$$N(\nu, t) = \nu^{-1} E(\nu, t) \tag{5.3}$$

where ν is the energy of the photon and t refers to the time at which the sample is analyzed. From there, we have the relation between the number distribution of the photons to the occupation number used in Chang and Cooper's paper (see Sec. (3):

$$f(\nu, t) = \frac{(h\ c)^3}{8.0\pi\nu^2} N(\nu, t) \tag{5.4}$$

Chang and Cooper's function, which they have time evolved uses the occupation number of the photons to time advance their solution and this number, $f(k, t)$, is very similar to a probability distribution in that all of the values are less than 1.0. Larsen and his compatriots, on the other hand, have chosen to time advance the energy distribution of their photons, but they again, do not use the same value for energy as was implied by the paper of Chang and Cooper. According to Chang and Cooper's paper and the scaling we have done above, the Planckian distribution function for the Energy, $E(\nu, t)$ is given as

$$B(\nu, t) = \frac{8\pi\nu^3}{(c\ h)^3[e^{\nu/\theta_e} - 1]} \tag{5.5}$$

Larsen's paper is related to this energy distribution and to Chang and Cooper's occupation number by:

$$u(\nu,t) = \nu^3 f(\nu,t) \tag{5.6}$$

$$= \left(\frac{c\,h}{m_e}\right)^3 \frac{1}{8\pi} E(\nu,t) \tag{5.7}$$

More about the specific manner in which these relations were found can be seen in Sec. (4.2).

6 Test Problems and Comparison

6.1 Problem 1: Stability Check with Equal Electron and Radiation Temperatures

The first set of tests run to compare both methods was given an initial Planck Distribution(Eq. 5.5) of the photons where the radiation and electron temperatures were equal. Physically, this situation is already in equilibrium, which means that the total number of photons and total energy of the system should have very little fluctuation. Some small fluctuation is expected because even though the system is in equilibrium, collisions still occur. Mathematically, this means that the derivative with respect to time of our function should be near zero.

The energy sampling mesh ranges from 0 to 500 keV with 1000 groups at evenly spaced intervals over a time interval of 1 shake with a time step of 5.0e-13 seconds. The electron temperature is set to 5.0 keV as is the radiation temperature for the initial distribution of the photons. The electron density, ρ_e, was set to $1.2e24$ and the Thomson scattering cross-section, σ_T, which is set to 0.67 barns.

To check how well the equilibrium solution is held by both methods, the change of the total energy and total number of photons is analyzed over time. As stated above, there should be very little fluctuation, since both the electron and radiation temperatures are constant and they start at the same value. To get these total values, a composite Trapezoidal rule was used to calculate the area under the properly scaled distribution curves with an equal step size of 0.5keV. As can be shown from Figures (5(a)) and (5(b)), this prediction holds fairly true. In Chang and Cooper's method, the number of photons changes by approximately 0.5% and the total energy changes by 8.0%. The number of photons changes by less than 0.4% in Larsen's method and the total energy by 10.6%. The change in the number of photons for both methods is relatively small and can be explained by the fact that we a distribution over an infinite space is being represented with a finite number of sampling points that only cover a small portion of this range. This means that the photon distribution will produce results that claim there are less photons at a given energy each time as it tries to spread the distribution out correctly and we can see this in the graph. Also, when tests were run using the same size energy step between bins over a larger range of values, the change in the solution decreases for both methods. This problem of representing the infinite distributions with finite points can be seen in the change in energy since the program requires that the equilibrium temperatures of both the electrons and radiation remain constant. The problem with this is that as the number of photons in the problem changes, the temperature at which the actual equilibrium solution can be maintained lowers slightly. Since the radiation and electron temperatures are the same throughout, the problem gradually starts to behave as a Heating Problem as time progresses. Therefore, it is reasonable to state that both the Chang and Cooper and the Larsen methods are relatively consistant and can hold an equilibrium solution.

6.2 Convergence Studies in Time and Space

These next tests were performed using a Compton Heating problem in which the initial radiation temperature was much lower than the initial electron temperature. In this case, $T_r = 5.0$ and $T_e = 37.0$. To better analyze this problem, the radiation temperature, T_r, was recorded as it changed over time along with the total number of photons, N_r, and the total energy of the radiation, E_r. It should be mentioned that the radiation temperature is not an actual temperature that can be measured and recorded, rather it is based on the Stefan-Boltzmann Law which relates total energy radiated from a black body to the fourth power of the Temperature of the black body. Therefore, T_r is calculated with the following relationship:

$$T_r = \left(\frac{c \cdot E_r}{4\sigma}\right)^{1/4} \tag{6.1}$$

where c is the speed of light in cm per sec, E is the energy of the system in keV, and σ is the Stefan-Boltzmann constant properly adjusted for units (ie it is given in keV per keV^4 rather than ergs per Kelvin to the fourth). In the case of a Compton Heating Problem, this value will rise as the photons interact with the electrons.

The other initial parameters set in this test are as follows:

1. The energy sampling mesh ν_i ranged from 0 to 1000 keV with an even spacing between sampling points
2. Tests were run for 1.0 shakes (1.0e-8 seconds) with equal step size between time advancements of the differential equation during each run
3. An initial Planckian distribution was set up in the energy space, but was converted to the appropriate spaces for time advancement by the program

The first set of tests were done to compare the reaction of Chang and Cooper's method versus Larsen's method when the size of the time step is held fixed and the number of points at which the distribution function is sampled is varied from 100 to 1000 sampling points or bins (see Table (1)). Fig. (6.2) shows that as the number of bins is increased, an asymptotic solution is approached by both methods. It is interesting to note that Larsen's method consistently predicts a lower Temperature at any given time and bin combination than does the method from Chang and Cooper. One explanation for this could be that Chang and Cooper's method uses the quasi-equilibrium solution f^e to calculate each new distribution and assumes that the given situation will evolve into this solution eventually, while Larsen's method does not use these values and, therefore, can only approximate what the final state of the solution should be. Also, both methods show that the radiation temperature drops during the first time step when 100 sampling points are used. This can be explained by the manner in which the total energy is calculated, and thus the radiation temperature, of our system. The most drastic change in these distributions function is usually between 0 and 100 keV by nature of the Planckian Distribution, but when only 100 sampling points are used to span the entire function from 0 to 1000 keV, only 10 sampling points fall in this range. Therefore, the area under this portion of the curve is not very accurate and this will influence the total number of photons and energy in the system greatly.

The next convergence tests hold the number of bins constant at 500 while varying the time between iterations of the algorithms. In this test, the step size ranges from 5.0e-10 to 5.0e-13 seconds(see Fig (7)). As one can see, both methods show an approach to an equilibrium solution asymptotically, although Larsen's method again appears to stabilize at a slightly lower temperature than that projected by Chang and Cooper's method, which is probably still a factor of the predicted equilibrium solution used in Chang and Cooper's method (see above).

6.3 Cost Comparison

Another factor used to analyze the effectiveness of these algorithms is the time cost of each program. To do so, the average time it takes to compute each bin at each step of the program, τ, can be computed by:

$$\tau = \frac{\text{total time}}{(\#\text{ time steps})(\#\text{ groups})} \tag{6.2}$$

To do this, the algorithms were timed using different numbers of sampling points with various size time steps (see Table (1)). The number of steps is calculated based on how many time steps are necessary to get from 0 to $1.0e-8$ shakes and the number of groups is spread out over 1000 keV, as in the tests above (6.2).

Table 1: Variables used in Creating Data for Cost Comparison

Possible # of groups	Possible # of time steps	Corresponding Δt
100	2001	5.0e-12
200	10002	1.0e-12
500	20001	5.0e-13
1000	100002	1.0e-13
	200002	5.0e-14

Once these runs were complete and τ calculated for each one, this value was plotted versus the amount of work done to produce the following plot (Fig. 8(a)). As one can see, the lines created by these values are fairly flat for both Chang and Cooper and Larsen's methods, which implies that they are an accurate representation of the average time per step per group of each method. The one measurement that is farthest from the line created by the rest in both methods is the one that was created using 100 groups and a time step of 5.0e-12. This is probably because the run

time for the entire problem is the shortest, but the same amount of time is necessary to read in the input deck and store the problem constants as the other runs. Larsen's Method is approximately 1.6 times faster than Chang and Cooper's, especially at higher refinements of Δt and Δx (Fig. 8(b)).

7 Conclusions

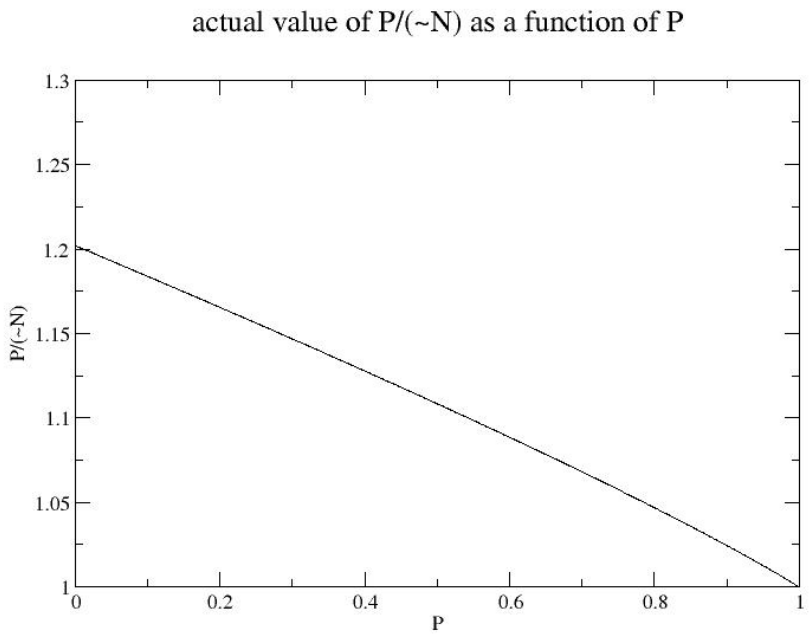
As the above data shows, Larsen's method appears to be a better overall approach to solving the problem of modeling Compton Scattering. It produces results that are within 1% of those produced by the Chang and Cooper Method in the problems where stability of the equillibrium solution is tested (6.1). Larsen's method is much quicker than the Chang and Cooper's method at the same refinement of time and space sampling points, although it is a bit slower to approach the stable value than Chang and Cooper's. Another advantage of using Larsen's method is that it does not place a restriction on the time range of the problem. Chang and Cooper's method, on the other hand, requires that the user calculate the equillibrium time before running the program to insure that they do not run longer than this time and get non-sensical information. Larsen's method also ensures that the distribution that it produces is always positive, which is useful to know, especially in large simulations. In dealing with problems of radiation transport, several factors showed themselves to be of great importance. The conversions between the different spaces - Larsen's scaled energy, standard energy, number density and occupation number - proved crucial to understanding the physical process and obtaining usable results from the program. This problem influenced the structure of the code as much as the actual methods described by Chang and Cooper and Larsen. Hopefully, this structure also leaves enough room for growth if the program is eventually attached to a hydrodynamics code which would cause the sampling mesh to change over the course of the problem, where it is fixed right now. The documentation and "in code" comments were also very valuable as the code became larger and more layered.

7.1 Future Research

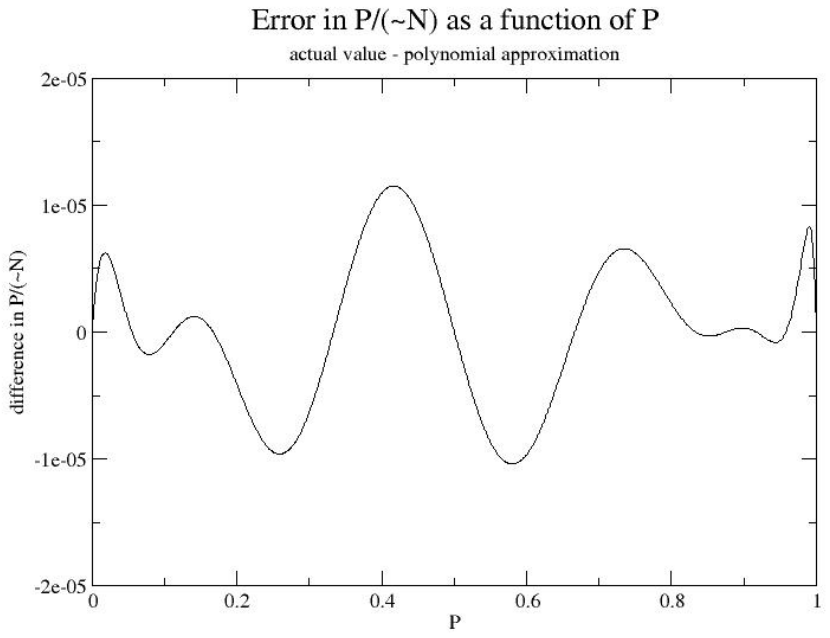
To further understanding of this problem, there are several things which could be investigated. First, it would be beneficial to implement a fully implicit version of the Modified Young's Method described in Larsen's Paper, the semi-implicit version of which has been studied in this paper (4). It requires substantially more time on algebraic preparation than was feasible as it has non-linear functions of the photon distribution function at the future time and therefore can not be solved by simple Gaussian Elimination. The Matrix Solver package should also be improved as a matter of course to use partial pivoting because that may account for part of the stability problems as well as situations were a negative value for the photon distribution was returned. Another area of interest for future research is how to implement a new operator splitting method that solves for diffusion in space and diffusion in energy simultaneously. This requires solving the two dependent, implicit differential equations, the Heat equation and the Fokker-Planck equation, which has been shown to be difficult in much of the literature on radiation transport. This could be done by storing the photon distribution in a matrix, where the columns represent the photon distribution at the different spatial sampling points, and preforming matrix multiplication to solve for the future time step. It also might be possible to correct for the discrepancy in the true solution and our approximation - solving for each term of the transport equation separately - by using a very small sample and running it through a code using stochastic or Monte Carlo methods as a kind of predictor-corrector idea. A more in depth research study that deals with similar types of interactions in other scientific fields, such as Plasma Physics, could show established algorithms where a mixture of stochastic and deterministic methods are used. It might be possible to adapt these methods to fit these Electron-Radiation interactions.

8 Acknowledgments

I would like to thank my mentors both at UCSD and LANL: Marv Alme and Robert Webster for suggesting and sponsoring this project, Daniel Tartakovsky and Michael Holst for providing advice on the paper and implementation of this project, and Ben Bergen for helping me learn to better utilize features of C++ in building this code.



(a) The ratio of $\frac{P}{\widetilde{N}}$ calculated by polynomial



(b) Error Plot: Ratio difference in calculation between infinite sum and polynomial

Figure 2: Graphical Analysis of the Fugacity Function - 3.3

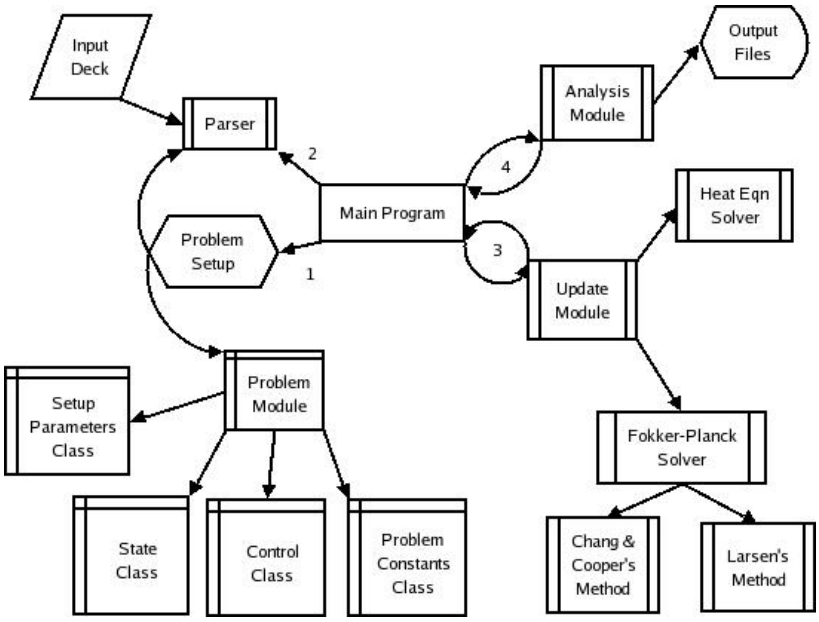
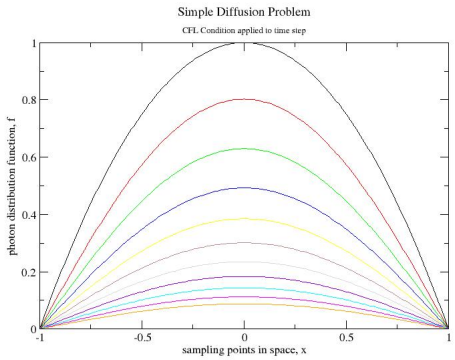
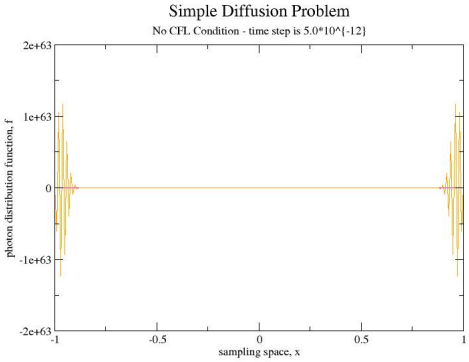


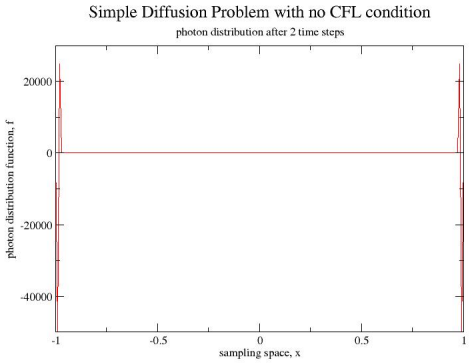
Figure 3: Pictoral Representation of the Code Structure



(a) CFL Condition Applied - time step is 4.002802e-15

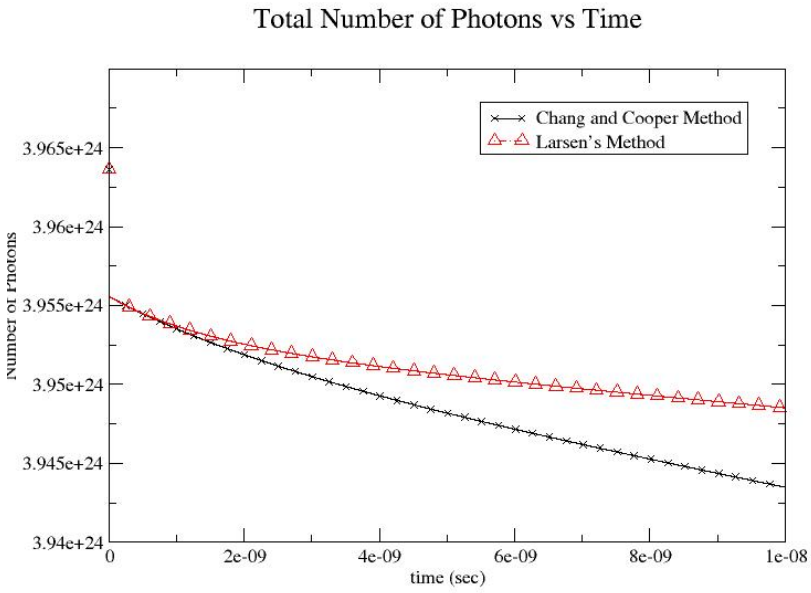


(b) No CFL Condition - time step is 5.0e-12

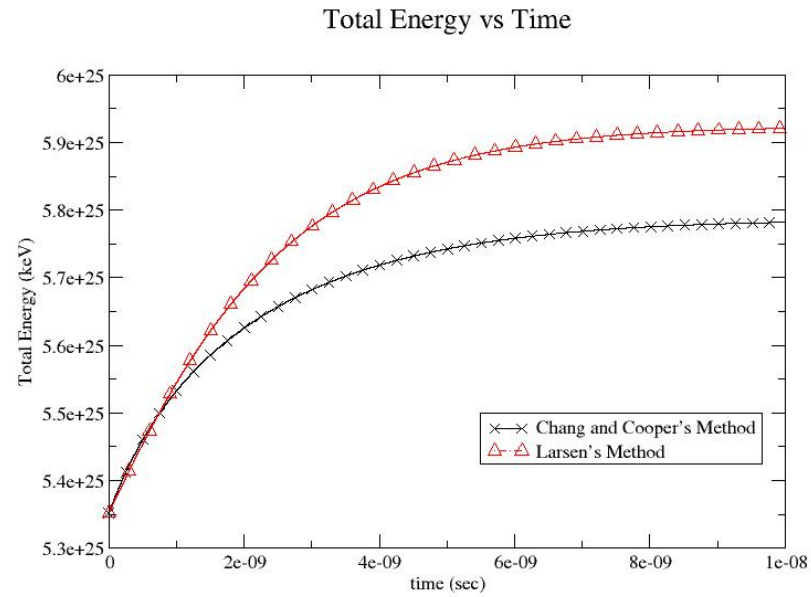


(c) No CFL Condition - function after only 2 iterations

Figure 4: Diffusion Approximation with and without CFL Condition



(a) Total Number of Photons over time



(b) Total Energy over time

Figure 5: Maintenance of Equilibrium

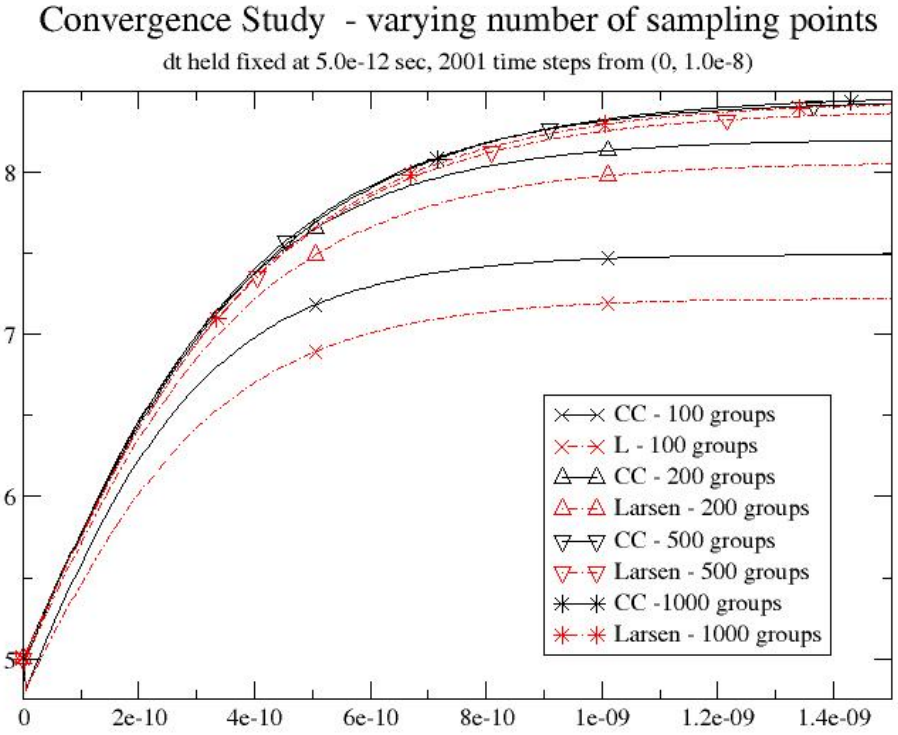
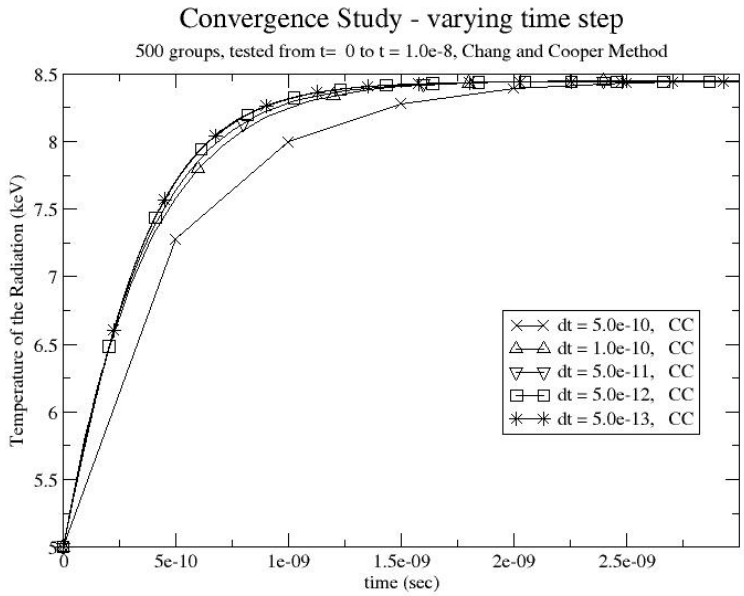
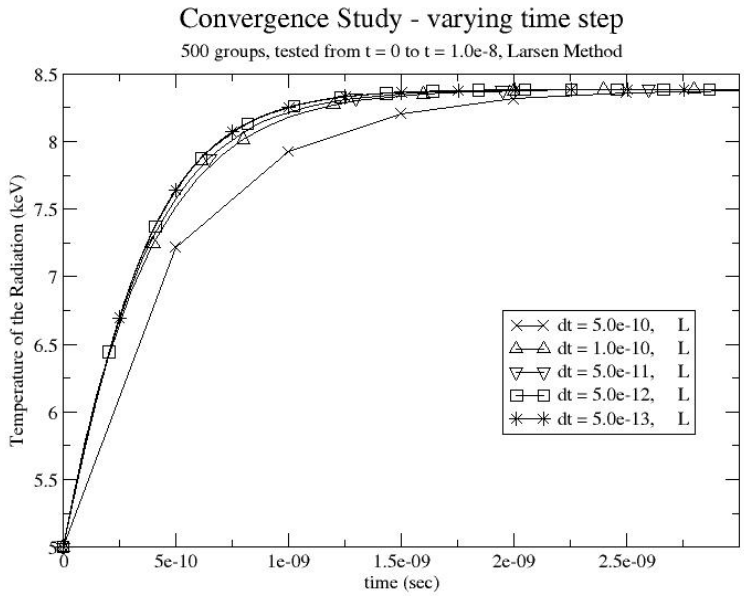


Figure 6: Convergence test with varying number of energy bins

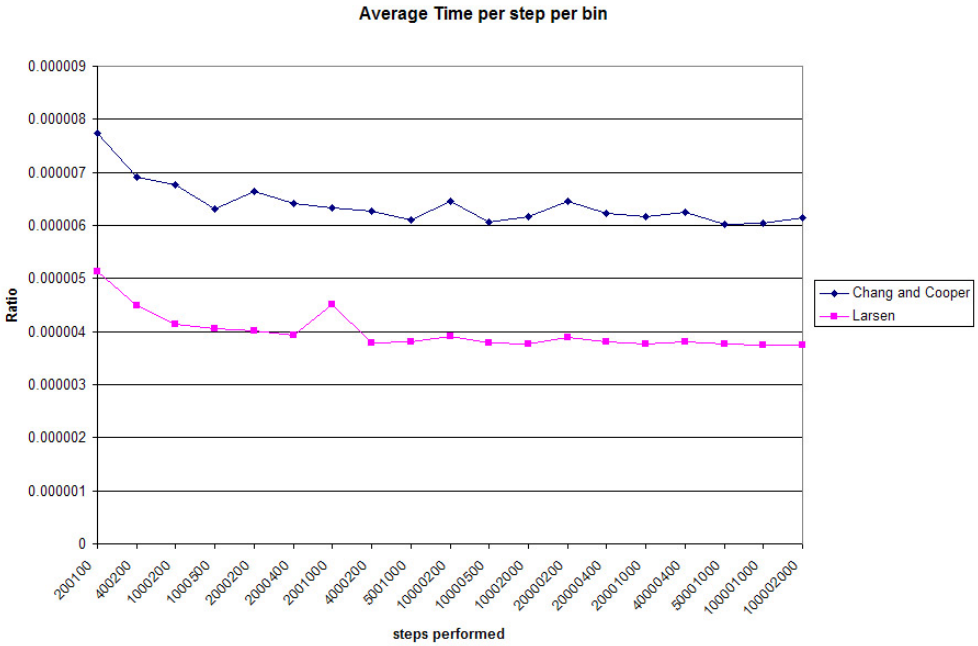


(a) Chang and Cooper's Method

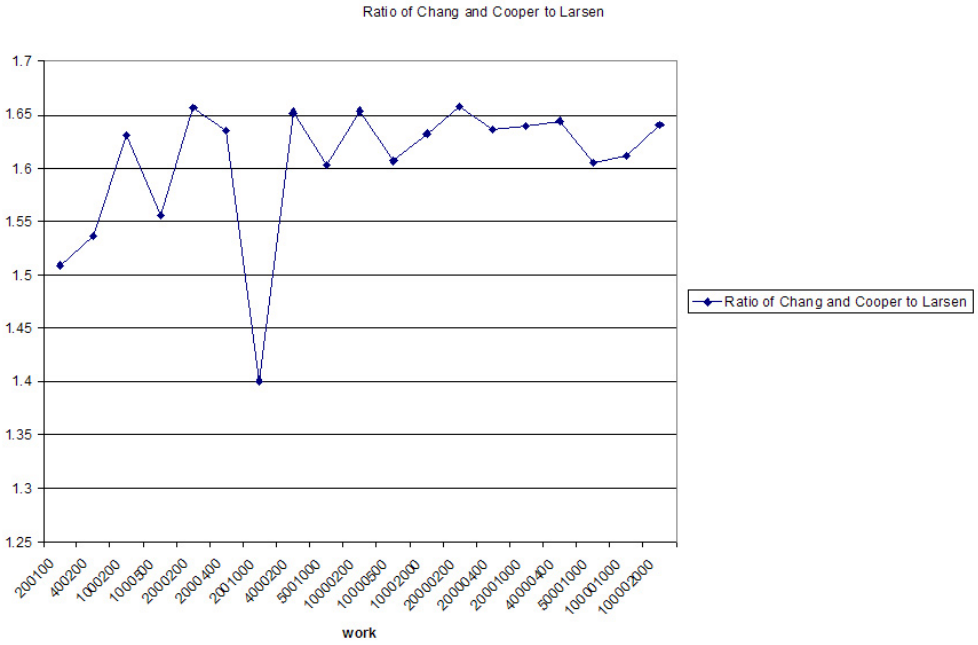


(b) Larsen's Method

Figure 7: Convergence test with varying time step



(a) τ versus the amount of work



(b) Ratio of $\tau_C C$ to τ_L

Figure 8: Cost Analysis of the two methods

References

[1] Peter A. Becker. Exact solution for the green's function describing time-dependent thermal comptonization. *Mon. Not. R. Astron. Soc.*, 343:215 – 240, 2003.

[2] Richard L. Bowers and James R. Wilson. *Numerical Modeling in Applied Physics and Astrophysics*. Jones and Bartlett, 1991.

[3] John Castor. *Radiation Hydrodynamics*. Camridge University Press, 2004.

[4] J. S. Chang and G. Cooper. A practical difference scheme for fokker-planck equations. *Journal of Computational Physics*, 6(1):1 – 16, 1970.

[5] A. R. Fraser. The fundamental equations of radiation hydrodynamics. AWRE 0-82/65, Atomic Weapons Research Establishment, January 1966.

[6] A. S. Kompaneets. The establishment of thermal equilibrium between quanta and electrons. *Soviet Physics JETP*, 4(5):730 – 737, June 1957.

[7] E.W. Larsen, C.D. Levermore, G.C. Pomraning, and J.G. Sanderson. Discretization methods for one-dimensional fokker-planck operators. *Journal of Computational Physics*, 61:359 – 390, 1985.

[8] Dimitri Mihalas and Barbara Weibel Mihalas. *Foundations of Radiation Hydrodynamics*. Oxford University Press, 1984.

[9] C.R. Nave. Hyperphysics: Quantum physics: Compton scattering. website: `http://hyperphysics.phy-astr.gsu.edu/Hbase/quantum/compton.html#c1`, 2005.

[10] G. C. Pomraning. *The Equations of Radiation Hydrodynamics*. Pergamon Press, 1973.

[11] George B. Rybicki and Alan P. Lightman. *Radiative Processes in Astrophysics*. John Wiley and Sons, Inc., 1979.

[12] Donald G. Shirk. A practical review of the kompaneets equation and its application to compton scattering. LAUR 14297, Los Alamos National Laboratory, May 2006.