STUDYING EFFECTS OF NON-EQUILIBRIUM RADIATIVE TRANSFER VIA HPC

A Dissertation

by

DANIEL ALPHIN HOLLADAY

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Ryan McClarren
Committee Members,	Marvin Adams
	Nancy Amato
	Jim Morel
Head of Department,	Yassin Hassan

May 2018

Major Subject: Nuclear Engineering

Copyright 2018 Daniel Alphin Holladay

ABSTRACT

A goal across many physics and engineering disciplines is to accurately simulate physical systems with radiation-hydrodynamics codes in which the assumption of local thermodynamic equilibrium (LTE) is predicted to be violated, such as laser driven hohlraums, stellar coronae, and supernova ejecta. In such regimes, the computational costs to capture relevant physics have been observed to be 3 orders of magnitude higher over the LTE problem. This thesis discusses innovations in non-LTE (NLTE) calculationsby leveraging high performance computing (HPC), load balancing, and algorithmic improvements to solve problems that were insoluble with existing techniques. We also present results demonstrating where NLTE is most important in simulation results. In particular, this capability was applied to 3 demonstration problems: a supernova (SN), a 1D problem meant to capture dynamics present in inertial confinement fusion (ICF) hohlraums, and a modified version of the blastwave diagnostic experiment performed at Sandia National Laboratory's (SNL) Z machine.

SN modeling was accomplished with a two stage process, first a radiation hydrodynamics (RH) simulation to model explosive dynamics followed by an offline post-processing stage for calculating quantities analogous to those measured by observers. The use of NLTE physics enabled by this work were important in both stages of this modeling, ultimately resulting qualitatively different results between LTE and NLTE.

The 1D hohlraum problem analysis indicates that the quantity of interest (QOI), the analogue of laser entrance hole (LEH) closing time, was mildly sensitive to the use of NLTE physics, but relatively insensitive to atomic model complexity, indicating some detail is needed in the treatment of auto-ionizing states. For non-integrated

quantities like the radiation spectral energy density, some modest differences were observed.

The Z machine blastwave diagnostic resulted in comparisons between LTE and NLTE, yielding some differences associated with the early dynamics of simulation, particularly with the hohlraum. More detailed comparisons corroborate the importance of NLTE modeling near hohlraum walls, but also support the idea that LTE is a reasonable assumption at late times, demonstrating that work done by experimental designers to avoid NLTE effects was successful.

ACKNOWLEDGEMENTS

I would like to thank my committee chair, Dr. McClarren, and my committee members, Dr. Adams, Dr. Amato, Dr. Fontes, and Dr. Morel for their mentorship during this project.

I would like to acknowledge my mentors and colleagues who have contributed large portions of their time to make this work possible: Wesley Even, Christopher Fontes, John Wöhlbier, Samuel Gutiérrez, and Gabriel Rockefeller. John and Dr. McClarren spent time to create the collaborations and find funding necessary for this work. The collaboration with Chis was a direct result of their work.

NOMENCLATURE

HEDP	High Energy Density Physics
RH	Radiation Hydrodynamics
LTE	Local Thermodynamic Equilibrium
AI	Auto-Ionization
ATS	Advanced Technology System
CPU	Central Processing Unit
CR	Collisional Radiative
DAG	Directed Acyclic Graph
DCA	Detailed Configuration Accounting
DOE	Department of Energy
EIE	Electron-Impact Exciation
EII	Electron-Impact Ionization
GPU	Graphics Processing Unit
HEDP	High Energy Density Physics
HPC	High Performance Computing
ICF	Iniertial Confinment Fusion
KNL	Knights Landing
LANL	Los Alamos National Laboratory
LEH	Laser Entrance Hole
LLNL	Lawerence Livermore National Laboratory
LTE	Local Thermodynamic Equilibrium
NIF	National Ignition Facility
NLTE	Non-Local Thermodynamic Equilibrium
ODE	Ordinary Differential Equation

PI	Photo-Ionization
QOI	Quanity of Interest
rDCA	Reduced Detailed Configuration Accouting
RH	Radiation Hydrodynamics
RHS	Right Hand Side
SN(e)	Supernova(e)
SNL	Sandia National Laboratory
STP	Standard Temperature and Pressure
TRT	Thermal Radiation Transport
VPU	Vector Pocessing Unit

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supervised by a dissertation committee consisting of Professor McClarren as the advisor, Professors Adams and Morel of the Department of Nuclear Engineering, Professor Amato of the Department of Computer Science and Engineering, and Dr. Fontes of the Los Alamos National Laboratory.

The problem setup for the 1D hohlraum simulations in chapter 6 was devised by Wesley Even. The input geometry of the blastwave simulation in chapter 6 was provided by Wesley Even. The atomic model data for all simulations were provided by Christopher Fontes.

All other work conducted for the dissertation was completed by the student. Much of the simulations and analysis was conducted on resources made available through the Los Alamos National Laboratory.

Funding Sources

Graduate study was funded under the NRC Fellowship during the 2012-2013 academic year and by the Los Alamos National Laboratory for the following years.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

The field of high energy density physics (HEDP) attempts to model physical systems in which pressures exceed 1Mbar [4, Chapter 1]. Systems of interest include astrophysical events such as supernovae (SNe) as well as inertial confinement fusion (ICF). At these conditions of high temperature and pressure, due to the non-linear dependence of radiation energy on temperature, which scales as temperature to the fourth power, the radiation field can carry a non-negligible amount of the total energy and momentum in the system. When this occurs, one must include radiation energy and momentum transport in hydrodynamic simulations in order to capture relevant physics.

1.1 Radiative Transfer

Accurate modeling of physical systems of interest require knowledge of the photon intensity, $I\left(\mathbf{r},\nu,\mathbf{\Omega},t\right)$, governed by the radiation transport equation given by

$$\frac{1}{c}\frac{\partial I\left(\mathbf{r},\nu,\mathbf{\Omega},t\right)}{\partial t}+\nabla\cdot\left(\mathbf{\Omega}I\left(\mathbf{r},\nu,\mathbf{\Omega},t\right)\right)+\sigma_t\left(\mathbf{r},\nu,t\right)I\left(\mathbf{r},\nu,\mathbf{\Omega},t\right)=\varepsilon\left(\mathbf{r},\nu,\mathbf{\Omega},t\right)+ \\ \int\limits_0^\infty \mathrm{d}\nu'\int\limits_{4\pi}\mathrm{d}\Omega'\left[\sigma_s\left(\mathbf{r},\nu'\to\nu,\mathbf{\Omega}'\cdot\mathbf{\Omega},t\right)I\left(\mathbf{r},\nu',\mathbf{\Omega}',t\right)\right]+Q\left(\mathbf{r},\nu,\mathbf{\Omega},t\right), \tag{1.1}$$

where c is the speed of light, $\mathbf{r}$ is a position vector, ν is the photon frequency, $\mathbf{\Omega}$ is a direction unit vector, t is the time variable, $\sigma_t\left(\mathbf{r},\nu,t\right)$ is the macroscopic total photon interaction cross section, $\varepsilon\left(\mathbf{r},\nu,\mathbf{\Omega},t\right)$ is the total emissivity, $\sigma_s\left(\mathbf{r},\nu'\to\nu,\mathbf{\Omega}'\cdot\mathbf{\Omega},t\right)$ is the macroscopic double differential scattering cross section, and $Q\left(\mathbf{r},\nu,\mathbf{\Omega},t\right)$ is an extraneous source. The field of collisional radiative modeling is dedicated to producing macroscopic cross sections and emissivities with sufficient fidelity and accuracy as to provide useful results for typical HEDP regimes. Under the assumptions of collisional radiative modeling, in which photons, free electrons, and ions are the primary interacting species, the emissivities and macroscopic cross sections are a function of the ion species, radiation and free electron energy distributions, free electron density, and (weakly) on the ion energy distribution. Under LTE conditions, the emissivity must be locally balanced by absorption in which the local radiation energy distribution has relaxed to a Planckian. The emissivity in LTE is given by

$$\varepsilon_{LTE}\left(\mathbf{r},\nu,\mathbf{\Omega},t\right) = \sigma_a\left(\mathbf{r},\nu,t\right) B\left(\nu,T\right), \tag{1.2}$$

where σ_a is the macroscopic cross section due to absorptions, $B\left(\nu,T\right)$ is the Planck distribution, and T is the temperature. The Planck distribution is given by

$$B\left(\nu,T\right) = \frac{2h\nu^3}{c^2}\frac{1}{\exp\left(\frac{h\nu}{k_BT}\right)-1}, \tag{1.3}$$

where h is Planck's constant and k_B is the Boltzmann constant. In LTE, a temperature is sufficient to describe the photon, electron, and ion energy distributions, and the emissivity is related to the macroscopic absorption cross section and does not need to be calculated separately. In TRT/RH codes that use the assumption of LTE, only the macroscopic cross sections are required to calculate all of the collisional radiative quantities. These macroscopic cross sections in some communities are referred to as the opacity. At the very least, these two quantities are related to each other and will now be discussed in further detail.

1.1.1 Opacities

A great number of resources have been allocated in the past to improving the computational methods used for solving transport equations. Included in that list are the highly coupled and non-linear equations of thermal radiative transfer. These methods have focused on decreasing the overall cost and improving robustness of algorithms such that the equations can be solved on very fine spatial grids, accurate angular quadratures, and novel energy discretization techniques. While this task is necessary for accurately modeling physical systems of interest, it is not solely sufficient. An additional requirement is the use of accurate physical quantities. One such quantity is the opacity, a quantity that describes how opaque a material is. It is related to the probability of a photon interacting with a given material, and thus related to the photon mean free path. The word 'opacity' is used in different ways among various communities. The HEDP community often refers to an opacity as a linear attenuation coefficient quantity, while in the atomic physics and other communities, it is defined as a mass attenuation coefficient. These differences can cause a non-insignificant amount of confusion amongst researches from different fields. This work will use the mass attenuation coefficient version. The two are relatable by the following relationship

$$\sigma\left(\nu\right) = \rho\kappa_{\nu}, \tag{1.4}$$

where $\sigma\left(\nu\right)$ is the macroscopic cross section or linear attenuation coefficient, which has dimensions of length^{-1}, κ_{ν} is the mass attenuation coefficient, which has dimensions of length$^2\times$ mass^{-1}, and ρ is the material density. In this work the word 'opacity' will refer to the mass attenuation coefficient, κ_{ν}, which is used throughout and, thus, factors of ρ will accompany this quantity where appropriate. The total macroscopic cross section of a material is given by the sum over all species of the number density of that species multiplied by its microscopic cross section. In this case the differentiation between species is chosen to be any feature (i.e. proton number, neutron number, nuclear excitation state, etc.) in which a change causes a change in the neutron microscopic cross section. In the case of TRT in HEDP, the material is hot enough to cause bound electrons to become excited out of their ground state and potentially become unbound (i.e. ionized). Atoms can thus have any number of bound electrons (assumed $[0, Z]$) where Z is the proton number, and the remaining bound electrons can be in any of a countably infinite set of configurations. This is not computationally tractable and so approximations are made. Currently, most inline capabilities take advantage of an approximation called the average atom model. This model averages both electronic configurations both within an ion stage and across ion stages. The benefit of this method is that the total number of unknowns can be reduced to $\mathcal{O}\left(10\right)$ thus making inline capabilities very fast, but sacrificing the ability to model potentially relevant physics. Another method of approximation, called configuration average, can contain many more unknowns as it only averages across electron configurations within the same ion stage. This work takes advantage of the configuration average approach using a reduced detailed configuration accounting (rDCA) model [1, 2]. DCA models can vary in the total amount of incorporated physics, but contain less averaging than average atom models [7, §2.5]. Each configuration average is called an energy level. The total number of species is thus the total number of energy levels in all ion stages, where the number density of atoms in ion stage i and energy level ℓ is $N_{i\ell}$. Then, the opacity is given by

$$\begin{aligned}\kappa_{tot}\left(\rho, T_e, T_r, h\nu\right) = &\sum_{i=1}^{I}\sum_{\ell=1}^{L_i}\frac{N_{i\ell}(\rho, T_e, T_r)}{\rho}\left(\sigma_{i\ell}^{(\text{bound–bound})}(h\nu) + \sigma_{i\ell}^{(\text{bound–free})}(h\nu)\right)\\ &+\frac{N_e}{\rho}\int\limits_{-1}^{1}\sigma_s^{(\text{free–free})}\left(\mu, h\nu\right)\mathrm{d}\mu + \kappa^{(\text{free-free})}(h\nu),\end{aligned}$$

where I is the total number of ion stages, L_i is number of energy levels in ion stage i, ρ is the mass density of the material, $\sigma_{i\ell}^{(\text{bound–bound})}(h\nu)$ is the energy dependent microscopic cross section due to bound-bound transitions, $\sigma_{i\ell}^{(\text{bound–free})}(h\nu)$ is the energy dependent microscopic cross section due to bound-free transitions, N_e is the free electron density, $\sigma_s^{(\text{free–free})}(\mu, h\nu)$ is the differential free-free scattering cross section due to scattering off of free electrons, and $\kappa^{(\text{free-free})}(h\nu)$ is the contribution to the opacity from inverse Bremsstrahlung absorption. The number densities of atoms in any given ion stage and energy level are calculated using local information about the system; namely, the material mass density ρ, the electron energy distribution $F(E_e)$, and the photon energy distribution $G(\nu)$. Solving this particular problem falls under the field of collisional radiative modeling and fortunately, it is a well known and well studied problem [18].

1.1.1.1 LTE vs NLTE

In terms of opacity computation, the difference between LTE and NLTE come down to the method used to calculate the populations, $N_{i\ell}$. Once known, the process of calculating the opacity is identical. In LTE, populations attain an equilibrium governed by the Saha relationship. In problems in which the density is an independently determined state variable, both LTE and NLTE steady state calculations require an iterative procedure to arrive at the correct electron density, yet they differ greatly in asymptotic complexity. The computational complexity for solving for the populations under the Saha-Boltzmann relationship is $\mathcal{O}\left(N\bar{M}\right)$ whereas solving the full rate matrix is of order $\mathcal{O}\left(N\bar{M}^3\right)$ where N is the number of ion stages being modeled and $\bar{M^\alpha} \equiv \frac{1}{N}\sum_{i=1}^{N} M_i^\alpha$. Even more importantly than these complexities is the fact that in LTE, electron energy distributions have relaxed to a Maxwell Boltzmann distribution at T and photons have relaxed to a Planck distribution at T. Thus the populations become a function of 2 variables and are thus amenable to tabulation. Even in the case when $T_r \neq T_e$, the 3-D (ρ, T_r, T_e) space could be stored in many situations. However, in the case in which the rate matrix is calculated from the spectral energy density, the input space becomes high dimensional and impractical to tabulate. While this is not yet implemented, this is one of the major drivers for developing `inlinlte` rather than attempting to tabulate NLTE opacities by adding T_r as a table variable. Thus in the case of LTE the cost of calculating the opacity is independent of the complexity of the atomic models used and only related to the number of interpolation points. This is not the case for calculating the opacity under NLTE conditions. Significant performance is necessary in order to mitigate the high computational cost of NLTE physics. Performance aspects of the `inlinlte` library will be discussed in greater detail chapter 4.

1.2 Novel Accomplishments

The aim of this work was to study the effect of relaxing the assumption of LTE in physical systems in the HEDP regime using an approach in which the atomic model complexity could vary from a highly averaged reduced detailed configuration accounting (rDCA) atomic models to very complex DCA atomic models. Even the reduced models have exhibited benefits over average atom models commonly used in an inline context [1]. The benefits in physical fidelity gained by these models come at a non-trivial computational cost. Due to the increased cost incurred, optimization was necessary if this capability was to be used for a realistic physical system. Improvements came both in software and algorithms. This was accomplished through writing the `inlinlte` library, taking advantage of the `Kokkos` library which aims to provide high performance and remain portable to current and upcoming computing hardware [6]. Time was spent to improve compilers' ability to vectorize performance of critical portions of inlinlte such that vector processing units (VPUs) could be effectively utilized. Additionally, inlinlte is portable to nVIDIA graphics processing units (GPUs), which will become increasingly important as GPUs become more ubiquitous on upcoming supercomputers. The issue of load imbalance arises in RH simulations and occurs due to the use of a temperature cutoff in which NLTE physics is only applied to spatial cells whose temperature is greater than the cutoff. Load balancing was implemented using a scalable algorithm which can be completed in logarithmic time. The load balancing algorithm is based on one developed for particle transport codes [16] but was modified to support non-power-of-two hardware resources using a weighted scheme. This scheme was implemented in order to be portable to computers whose compute nodes have heterogeneous performance characteristics. These improvements resulted in more than an order of magnitude performance improvement over the previous capability available in xRAGE in two of the problems examined. This improvement resulted in a parameter study of a 1D problem meant to be similar to a laser driven hohlraum as well as a 2D simulation of problem with the geometry of an HEDP experiment. The ability to run 2D simulations with realistic geometries and taking into account NLTE effects in a few weeks time is a boon for the HEDP community.

2. ATOMIC PHYSICS

For high energy density physics applications, calculating the macroscopic opacity of a material requires knowledge of the fundamental, microscopic interactions between photons and electrons. Atomic physics is the study of these interactions, which includes both free electrons and those that are bound to atomic nuclei.

In systems in which the RH equations are valid, photons, free electrons, and ions interact with each other. The probabilities at which these interactions occur depend on quantities such as the incident photon or electron energy, atomic species, and bound-electron configurations. Outlined in the following sections are the atomic processes most commonly considered in collisional radiative (CR) modeling. For brevity, the word "atom" in this work can refer to both a neutral atom or a charged ion.

In the following section, it will be necessary to use a notation that can distinguish between atoms with different numbers of bound electrons. Atoms that possess the same number of bound electrons belong to the same ion stage, denoted by the index i. The neutral atom is chosen to be in the 0^{th} ion stage ($i = 0$). Atoms in which i electrons have been removed relative to the neutral stage possess a charge i and are said to be in the i^{th} ion stage. Within an ion stage, the atoms can be further separated into energy levels, which arise form different bound-electron configurations. An atom in the i^{th} ion stage and ℓ^{th} energy level is denoted by these two indices enclosed in parentheses, $(i\ell)$. The number density, or "population", of atoms in this level is denoted by $N_{i\ell}$. A complete set of populations is obtained by solving the CR equations for a given set of conditions and these data are of fundamental importance in calculating the opacity.

2.1 Fundamental Atomic Processes

Understanding the mechanisms that describe how atoms transition between different ion stages and energy levels is necessary for quantitatively studying HEDP. The following subsections describe the various atomic processes important for CR modeling.

2.1.1 Electron-Impact Excitation and De-excitation

Electron-impact excitation (EIE) is the process in which an atom in a particular energy level $(i\ell)$ is excited to a higher energy level (im) by a free electron. In this process, the excitation energy is transferred from the free electron to the atom. Electron-impact de-excitation is the inverse process in which an atom transitions to a lower energy level and the de-excitation energy is transferred to a free electron. The excitation process can be symbolically represented by an expression of the form

$$(i\ell) + e \rightarrow (im) + e', \tag{2.1}$$

where e is a free electron and the prime denotes that the final electron has a kinetic energy that is lower than that of the incident electron. The change in the free electron's kinetic energy is determined from energy conservation and is exactly equal to the transition energy,

$$E_0 \equiv E_{im} - E_{i\ell}. \tag{2.2}$$

In an excitation process, the free electron's kinetic energy is *decreased* by E_0, whereas it is increased by the same amount in a de-excitation process.

A quantitative description of EIE rates is necessary for solving the CR equations.

The number of transitions per unit volume per unit time at which atoms are being excited from level $(i\ell)$ to (im) due to free electrons with energies $\mathrm{d}E$ about E is

$$\mathfrak{S}_{i,\ell\to m}\left(E\right)\mathrm{d}E = N_{i\ell}N_e F\left(E\right) v\left(E\right)\sigma_{i,\ell\to m}\left(E\right)\mathrm{d}E, \tag{2.3}$$

where $\mathfrak{S}_{i,\ell\to m}\left(E\right)$ is the (energy) differential transition rate per unit volume, $N_{i\ell}$ is the population of atoms in level $(i\ell)$, N_e is the free electron number density, $F\left(E\right)$ is the free electron energy distribution, $v\left(E\right)$ is the free electron speed, and $\sigma_{i,\ell\to m}\left(E\right)$ is the electron-impact excitation cross section for the transition from level $(i\ell)$ to (im). The total rate per unit volume, $\mathcal{S}_{i,\ell\to m}$, is obtained by integrating the differential transition rate per unit volume over all free electron energies. Since the cross section is 0 for energies below the excitation energy E_0, the integration range can be truncated as follows

$$\mathcal{S}_{i,\ell\to m} \equiv \int\limits_{0}^{+\infty} \mathfrak{S}_{i,\ell\to m}\left(E\right)\mathrm{d}E = N_{i\ell}N_e \int\limits_{E_0}^{+\infty} F\left(E\right) v\left(E\right)\sigma_{i,\ell\to m}\left(E\right)\mathrm{d}E. \tag{2.4}$$

A related quantity of particular convenience when constructing the CR equations is the rate coefficient $s_{i,\ell\to m}\left(F\right)$, which is defined as

$$s_{i,\ell\to m}\left(F\right) \equiv \int\limits_{E_0}^{+\infty} F\left(E\right) v\left(E\right)\sigma_{i,\ell\to m}\left(E\right)\mathrm{d}E. \tag{2.5}$$

In the special case where $F\left(E\right)$ is well described by its equilibrium distribution, that is the Maxwell-Boltzmann distribution, the distribution function $F\left(E\right)$ is characterized by the scalar temperature T and the rate coefficient can be described by a function of T rather than a functional of $F\left(E\right)$. In the DCA approach, $F\left(E\right)$ is assumed to be a Maxwell-Boltzmann distribution, and the rate coefficients are represented by the formula

$$s_{i,\ell \to m}^{\mathrm{FIT}}\left(T\right) = \frac{8\pi^2 a_0^2}{E_0}\sqrt{\frac{8}{3\pi m_e kT}}\exp\left(\sum_{n=0}^{3} a_n x^n\right), \tag{2.6}$$

where $x = \log\left(\frac{E_0}{kT}\right)$ and a_n are fit coefficients calculated by ATOMIC, which is a plasma modeling code in the LANL suite of atomic physics codes [18, Chapter 2.2.1] [7]. The electron-impact de-excitation rate coefficient can be calculated from the relationship

$$t_{i,m \to \ell}\left(T\right) = \frac{g_{i\ell}}{g_{im}}\exp\left(\frac{E_0}{kT}\right) s_{i,\ell \to m}\left(T\right), \tag{2.7}$$

where $g_{i\ell}$ is the statistical weight of level $(i\ell)$. The relationship described in (2.7) can be derived from the principle of detailed balance. The relationship described in (2.7) holds when using equation (2.6) to obtain an analogous formula for the inverse process, $t_{i,m \to \ell}^{\mathrm{FIT}}\left(T\right)$.

2.1.2 Electron-Impact Ionization and Three-Body Recombination

Electron-impact ionization (EII) is the process in which a free electron transfers energy to an atom, resulting in the ejection of a bound electron. This process changes the electron configuration by removing an electron and increases the charge of the atom by one. Three-body recombination is the inverse process in which a free electron recombines and becomes bound, transferring its kinetic energy to the atom. The ionization process can be represented by an expression of the form

$$(i\ell) + e \to (jm) + e' + e'', \tag{2.8}$$

where the two electrons e' and e'' have a lower total kinetic energy than the initial energy of the impacting electron. In this work, only single ionization events will be considered due to their relatively high probability of occurrence when compared multiple ionization events.

A discussion of EII rates proceeds in a similar manner to the one above for EIE rates. The number of transitions per unit volume per unit time that atoms are being ionized from level $(i\ell)$ to (jm) due to EII from free electrons with initial energies of $\mathrm{d}E$ about E and all physically possible final energies is

$$\mathfrak{C}_{i\to j,\ell\to m}\mathrm{d}E = N_{i\ell}N_e F\left(E\right)v\left(E\right)\sigma_{i\to j,\ell\to m}\left(E\right)\mathrm{d}E, \tag{2.9}$$

where $\mathfrak{C}_{i\to j,\ell\to m}$ is the differential electron-impact ionization rate per unit volume, $\sigma_{i\to j,\ell\to m}\left(E\right)$ is the cross section for the specified interaction. The rate coefficient is the integral over all free electron energies above the ionization energy E_0

$$c_{i\to j,\ell\to m}\left(F\right) = \int\limits_{E_0}^{+\infty} F\left(E\right)v\left(E\right)\sigma_{i\to j,\ell\to m}\left(E\right)\mathrm{d}E. \tag{2.10}$$

In the DCA approach, when $F\left(E\right)$ is well described by a Maxwell-Boltzmann distribution, these rate coefficients are represented by the formula

$$c^{\mathrm{FIT}}_{i\to j,\ell\to m}\left(T\right) = \frac{8\pi^2 a_0^2}{E_0}\sqrt{\frac{8}{3\pi m_e kT}}\exp\left(-\frac{E_0}{kT}+\sum_{n=0}^{3}a_n x^n\right), \tag{2.11}$$

where $x = \log\left(\frac{E_0}{kT}\right)$ and a_n are fit coefficients calculated by the ATOMIC code [18, Chapter 2.2.1] [7]. The three-body recombination rate coefficient can be calculated from the concept of detailed balance, through the relationship

$$b_{j\to i,m\to \ell}\left(T\right) = \frac{g_{i\ell}}{g_{jm}} \frac{h^3 \exp\left(\frac{E_0}{kT}\right)}{2\left(2\pi m_e kT\right)^{\frac{3}{2}}} c_{i\to j,\ell\to m}\left(T\right). \tag{2.12}$$

This relationship holds when using the fit formula described in (2.11).

2.1.3 Auto-ionization and Dielectronic Recombination

Auto-ionization (AI) is the process that results in the ionization of a bound electron through de-excitation of a different bound electron. This process can be represented with an expression of the form

$$(i\ell) \to (jm) + e. \tag{2.13}$$

The number of AI transitions per unit volume per unit time of atoms from level $(i\ell)$ to (jm) accompanied by an emitted free electron with energies $\mathrm{d}E$ about E is

$$\mathfrak{A}_{i\to j,\ell\to m}\mathrm{d}E = N_{i\ell} A_{i\to j,\ell\to m} \beta_a\left(E\right)\mathrm{d}E, \tag{2.14}$$

where $\mathfrak{A}_{i\to j,\ell\to m}$ is the differential auto-ionization rate per unit volume. $\beta_a\left(E\right)$ is a normalized Cauchy or Lorentz distribution function, which depends on the lifetime associated with the transition. Thus the rate coefficient, which is simply the rate per atom in this case, is the integral over all free electron energies, i.e.

$$a_{i\to j,\ell\to m} = A_{i\to j,\ell\to m}. \tag{2.15}$$

This quantity can be calculated by ATOMIC, the LANL suite of atomic physics codes [18, Chapter 2.2.1] [7]. The dielectronic recombination rate coefficient can be calculated from the concept of detailed balance, through the relationship

$$d_{j\to i,m\to\ell} = \frac{g_{i\ell}}{g_{jm}} \frac{h^3 F\left(E_0\right)}{4\pi\left(2m_e\right)^{\frac{3}{2}}} a_{i\to j,\ell\to m}. \tag{2.16}$$

2.1.4 Photo-excitation and De-excitation

Photo-excitation is the process in which a photon is absorbed by an atom and its energy excites a bound electron. Photo-de-excitation is the inverse process in which an electron de-excites releasing its energy in the form of a photon. The excitation process can be described by an expression of the form

$$(i\ell) + h\nu_0 \to (im), \tag{2.17}$$

where $h\nu_0$ is a photon with frequency ν_0. The change in the photon's energy can be expressed in terms of the transition energy,

$$h\nu_0 \equiv E_{im} - E_{i\ell}. \tag{2.18}$$

The rate coefficients for processes in which the incident particle is a photon depend on quantities related to the intensity I_ν, which is one of the primary unknowns in the radiative transfer equations. Particularly, rate coefficients depend on the mean spectral intensity, denoted henceforth as $G\left(h\nu\right)$. The mean spectral intensity is the average intensity per unit solid angle given by

$$G\left(h\nu\right) = \frac{1}{4\pi} \iint_{4\pi} I_\nu \, \mathrm{d}\Omega. \tag{2.19}$$

In thermal equilibrium (TE), the mean spectral intensity is given by the Planck distribution at the equilibrium temperature. The number of transitions per unit volume per unit time of atoms changing from $(i\ell)$ to (im) due to photo excitations with energies d$h\nu$ about $h\nu$ is

$$\mathfrak{U}_{i,\ell\to m}\mathrm{d}h\nu = N_{i\ell}G\left(h\nu\right)\sigma_{i,\ell\to m}\left(h\nu\right)c\ \mathrm{d}h\nu, \tag{2.20}$$

where $\mathfrak{U}_{i,\ell\to m}$ is the differential radiative excitation rate per unit volume, $\sigma_{i,\ell\to m}\left(h\nu\right)$ is the radiative excitation cross section for the excitations from $(i\ell)$ to (im). The cross section is related to the statistically weighted oscillator strength, $gf_{i,\ell\to m}$

$$\sigma_{i,\ell\to m}\left(h\nu\right) = \frac{he^2\pi}{m_ec}\frac{gf_{i,\ell\to m}}{g_{i\ell}}\beta\left(h\nu\right), \tag{2.21}$$

where $\beta\left(h\nu\right)$ is a normalized line shape function. A common choice for the line shape function is the Cauchy distribution, often called a Lorentzian or Breit-Wigner distribution which is given by

$$\beta_{\text{Lorentzian}}\left(h\nu\right) = \frac{\Gamma}{\pi\left(\left(h\nu - h\nu_0\right)^2 + \Gamma^2\right)}, \tag{2.22}$$

where Γ is the full width at half maximum of the distribution and $h\nu_0$ is the transition energy. Thus the radiative excitation rate coefficient, which is the rate per atom, is the integral over all photon energies

$$u_{i,\ell\to m}\left(G\right) = \frac{he^2\pi}{m_e}\frac{gf_{i,\ell\to m}}{g_{i\ell}}\int\limits_0^{+\infty} G\left(h\nu\right)\beta\left(h\nu\right)\mathrm{d}h\nu. \tag{2.23}$$

In the case where $G\left(h\nu\right)$ is slowly varying relative to the line shape function, the highly peaked line shape function will select the photon energy distribution $G\left(h\nu\right)$

at the transition energy, such that the integral is well approximated by $G\left(h\nu_0\right)$. This selection is a property of delta function distributions called the sifting property. Radiative de-excitation contains contributions both from spontaneous decay and stimulated emission. The spontaneous decay rate, also called the Einstein A coefficient, can be written as

$$y_{i,m\to\ell} = \int_0^{+\infty} \sigma'_{i,m\to\ell}\left(h\nu\right) \mathrm{d}h\nu, \tag{2.24}$$

where the primed cross section is obtained through micro-reversibility and given by

$$\sigma'_{i,m\to\ell}\left(h\nu\right) = \frac{8\pi}{h^3c^2}\frac{g_{i\ell}}{g_{im}}\left(h\nu\right)^2 \sigma_{i,\ell\to m}\left(h\nu\right). \tag{2.25}$$

Again, the line shape function will sift the integrand, yielding

$$y_{i,m\to\ell} = \frac{8\pi^2e^2}{h^2c^3m_e}\frac{gf_{i,\ell\to m}}{g_{im}}\left(h\nu_0\right)^2. \tag{2.26}$$

The contribution to the radiative de-excitation rate coefficient due to stimulated emission can be expressed in terms of the radiative excitation rate coefficient as

$$z_{i,m\to\ell}\left(G\right) = \frac{g_{i\ell}}{g_{im}}u_{i,\ell\to m}\left(G\right). \tag{2.27}$$

2.1.5 Photo-ionization and Radiative Recombination

Photo-ionization (PI) is the process in which a photon is absorbed by an atom and results in the ionization of a bound electron. Radiative recombination is the inverse process in which a free electron recombines with an atom becoming bound, resulting in the emission of a photon. Photo-ionization of a single electron can be described by an expression of the form

$$(i\ell) + h\nu \to e + (jm)\,. \tag{2.28}$$

The corresponding energy conservation equation is given by

$$h\nu = \underbrace{\phi_i + E_{jm} - E_{i\ell}}_{h\nu_0} + E, \tag{2.29}$$

where ϕ_i is the ionization potential. Note that a photon with energy $h\nu_0$ would yield a free electron with 0 kinetic energy. The rate per unit volume at which atoms in state $(i\ell)$ are being ionized to to state (jm) due to photons with energies $\mathrm{d}h\nu$ about $h\nu$ is

$$\mathfrak{P}_{i\to j,\ell\to m}\mathrm{d}h\nu = N_{i\ell}G\left(h\nu\right)\sigma_{i\to j,\ell\to m}\left(h\nu\right)c\ \mathrm{d}h\nu, \tag{2.30}$$

where $\sigma_{i\to j,\ell\to m}\left(h\nu\right)$ is the interaction cross section. Thus, the PI rate coefficient is

$$p_{i\to j,\ell\to m}\left(G\right) = \int\limits_{h\nu_0}^{+\infty} G\left(h\nu\right)\sigma_{i\to j,\ell\to m}\left(h\nu\right)c\ \mathrm{d}h\nu. \tag{2.31}$$

In the DCA approach, these cross sections are well represented by the formula

$$\sigma^{\mathrm{FIT}}_{i\to j,\ell\to m}\left(h\nu\right) = \frac{1}{h\nu_0}\exp\left(\sum_{n=0}^{3} a_n x^n\right), \tag{2.32}$$

where $x \equiv \log\left(h\nu\right) - \log\left(h\nu_0\right)$ and a_n are fit coefficients calculated by the ATOMIC code [18, Chapter 2.2.1] [7]. For recombination rates, the cross section formula given in (2.32) only holds if $F\left(E\right)$ is well described by a Maxwell-Boltzmann distribution. The rate coefficient for radiative recombination without taking stimulated emission into account is

$$r_{j\to i,m\to \ell}\left(F\right)=\frac{g_{i\ell}}{g_{jm}}\frac{1}{\sqrt{2m_e^3}c^2}\int\limits_{0}^{+\infty}F\left(E\right)\frac{\left(h\nu_0+E\right)^2}{\sqrt{E}}\sigma_{i\to \ell,j\to m}\left(h\nu_0+E\right)\mathrm{d}E, \tag{2.33}$$

where $F\left(E\right)$ is the free electron distribution. The contribution to the radiative recombination rate coefficient due to stimulated emission is

$$q_{i\to \ell,j\to m}\left(F,G\right)=\frac{g_{i\ell}}{g_{jm}}\frac{h^3c}{\pi\sqrt{128m_e^3}}\int\limits_{0}^{+\infty}F\left(E\right)\frac{\left(h\nu_0+E\right)^2}{\sqrt{E}}\sigma_{i\to \ell,j\to m}\left(h\nu_0+E\right)G\left(h\nu_0+E\right)\mathrm{d}E. \tag{2.34}$$

Notice that this approximation is for the cross section itself and not for the rate coefficient, meaning that these integrals have to be carried out explicitly. For some CR calculations, calculating these integrals can dominate the computation time.

2.1.6 Inverse Bremsstrahlung

Inverse bremsstrahlung, also called free-free absorption, is the process in which a photon is absorbed by a free electron, causing a change in the energy and momentum of the free electron. This process is not an aspect of solving the CR equations for the populations, but is a component of the opacity. This work uses the Kramers opacity for the free-free contribution to the opacity, given by the following relationship [24, Chapter 5.3]

$$\kappa_\nu^{(\text{free-free})}\propto\frac{\left\langle Z^2\right\rangle}{\sqrt{T_e}\nu^3}, \tag{2.35}$$

where $\left\langle Z^2\right\rangle$ is the mean squared charge.

2.1.7 Scattering

In general, photon scattering is a process in which a photon and electron interact and the result is a photon with a potentially different energy and momentum. During a scattering process, photons can interact with bound and free electrons. In the case of scattering events, the probability of interaction depends on the energies of the participating photon and electrons.

2.1.7.1 Thomson Cross Section

Thomson scattering is a model for scattering that only changes the direction of the photon and electron, i.e. elastic scattering. It can be derived from classical electromagnetism and is accurate at low photon energies. The Thomson cross section is the result of integrating the differential cross section over all angles

$$\sigma_{\text{Thomson}} = 2\pi \int_{-1}^{+1} \frac{r_0^2}{2} \left(1 + \mu^2\right) \mathrm{d}\mu = \frac{8\pi}{3} r_0^2, \tag{2.36}$$

where r_0 is the classical electron radius given by $\frac{e^2}{4\pi\epsilon_0 m_e c^2}$ where e is the electron charge, ϵ_0 is the vacuum permittivity, m_e is the electron mass, and c is the speed of light.

2.1.7.2 Scattering Opacity

The scattering opacity can be calculated given a scattering cross section. Given the Thomson cross section, the scattering opacity is given by

$$\kappa_{\nu,s} = \frac{N_e \sigma_{\text{Thomson}}}{\rho}. \tag{2.37}$$

Both the electron and mass densities have linear dependence on the total atomic number density, thus this formula can be rewritten as

$$\kappa_{\nu,s} = \frac{\langle Z \rangle}{M} \underbrace{\sigma_{\text{Thomson}} N_A}_{\approx 0.4\text{cm}^2}, \tag{2.38}$$

where M is the molar mass of the material and N_A is Avogadro's number. M is well approximated by the material mass number A, leading to scattering opacity used in this work, given by

$$\kappa_{\nu,s} \left[\frac{\text{cm}^2}{\text{g}} \right] = \frac{0.4 \langle Z \rangle}{A}. \tag{2.39}$$

The scattering term is independent of energy, and may not be accurate at high energies, where Compton scattering is more representative.

2.1.8 Omitted Processes

The presented processes do not represent a complete list of all possible atomic processes. As previously mentioned, multi-ionization processes were not presented due to their low probability relative to single ionization events. Pair production can contribute to the opacity by increasing the absorption probability for photons. However, photons must at least exceed the threshold energy of $\sim$ 1.022 MeV to have a non-zero pair production probability and that cross sections peaks at a much higher energy. Similarly, nuclear interactions such as electron capture and internal conversion are able to change the bound electron configuration and could affect the populations. However, these events occur with a much lower probability in systems of interest than do the processes described above. While certain problems may call for explicit accounting of these processes, details associated with them are beyond the scope of this work.

2.2 Rate Equations

The rate coefficients for the atomic processes as described in the previous section allow one to write down a coupled set of ordinary differential equations (ODEs) to describe the atomic level populations for a given set of conditions.

2.2.1 Background

When a system is in LTE, the free electrons have relaxed to a Maxwellian distribution with temperature T_e and the populations can be described by the Saha-Boltzmann relationship

$$\frac{N_{i\ell}}{N_{jm}} \propto \exp\left(-\frac{E_0}{k_B T_e}\right), \tag{2.40}$$

where E_0 is the energy difference between the two levels and k_B is the Boltzmann constant. Under LTE, the EIE, EII, and AI rates are each balanced by the rate of their respective inverse processes. In the case when photons have relaxed to a Planckian distribution with temperature T_r, and $T_r = T_e$, all collisional and radiative processes are in equilibrium with their respective inverse processes.It is also possible to be out of LTE and for the system to be in steady state. An example of this is called coronal equilibrium in which the electron density is sufficiently small that each ion stage is expected to contain significant population only in its ground state. In that case, electron-impact excitation and ionization is balanced by radiative de-excitation and recombination. This steady-state case requires that one solve the system of ODEs with the simplifying constraint $\frac{\mathrm{d}N_{i\ell}}{\mathrm{d}t} = 0$. In the numerical implementation, the steady-state case has the benefit that the populations need not be stored between RH time steps, significantly reducing memory requirements.

2.2.2 Overview

In order to write down the CR rate equations, the net rate of change for each process under consideration is needed. In the previous section, an expression was obtained for the various rate coefficients between two levels. The net rate of change for a given level due to a particular process is a sum of the rate coefficients into and out of that level, multiplied by the number density of their respective reactants (ions and free electrons).

2.2.2.1 Net Rate due to Electron-Impact Excitation and De-excitation

Rate coefficients given in equations (2.6) and (2.7) can describe the net change rate for atoms in the $(i\ell)$ level due to electron impact excitations and de-excitations, given by

$$\begin{aligned} \mathcal{S}_{i\ell} = \sum_m N_e\{ & N_{im}\left[s_{i,m\to\ell}\left(T_e\right) + t_{i,m\to\ell}\left(T_e\right)\right] \\ & - N_{i\ell}\left[s_{i,\ell\to m}\left(T_e\right) + t_{i,\ell\to m}\left(T_e\right)\right]\}. \end{aligned} \tag{2.41}$$

2.2.2.2 Net Rate due to Electron-Impact Ionization and Three-Body Recombination

Rate coefficients given in equations (2.11) and (2.12) can describe the net change rate for atoms in the $(i\ell)$ level due to electron impact ionizations and three-body recombinations, given by

$$\begin{aligned} \mathcal{C}_{i\ell} = \sum_{jm} N_e\{ & N_{jm}\left[c_{j\to i,m\to\ell}\left(T_e\right) + N_e b_{j\to i,m\to\ell}\left(T_e\right)\right] \\ & - N_{i\ell}\left[c_{i\to j,\ell\to m}\left(T_e\right) + N_e b_{i\to j,\ell\to m}\left(T_e\right)\right]\}. \end{aligned} \tag{2.42}$$

2.2.2.3 Net Rate due to Autoionization and Dielectronic Recombination

Rate coefficients given in equations (2.15) and (2.16) can describe the net change rate for atoms in the $(i\ell)$ level due to autoionizations and dielectronic recombinations, given by

$$\begin{aligned}\mathcal{A}_{i\ell} = \sum_{jm} & N_{jm}\left[a_{j\rightarrow i,m\rightarrow\ell} + N_e d_{j\rightarrow i,m\rightarrow\ell}\left(T_e\right)\right] \\ & -N_{i\ell}\left[a_{i\rightarrow j,\ell\rightarrow m} + N_e d_{i\rightarrow j,\ell\rightarrow m}\left(T_e\right)\right].\end{aligned} \tag{2.43}$$

2.2.2.4 Net Rate due to Photo-excitation and De-excitation

Rate coefficients given in equations (2.23), (2.26), and (2.27) can describe the net change rate for atoms in the $(i\ell)$ level due to photo-excitations and de-excitations, given by

$$\begin{aligned}\mathcal{U}_{i\ell} = \sum_{m} & N_{im}\left[u_{i,m\rightarrow\ell}\left(T_r\right) + y_{i,m\rightarrow\ell} + z_{i,m\rightarrow\ell}\left(T_r\right)\right] \\ & -N_{i\ell}\left[u_{i,\ell\rightarrow m}\left(T_r\right) + y_{i,\ell\rightarrow m} + z_{i,\ell\rightarrow m}\left(T_r\right)\right].\end{aligned} \tag{2.44}$$

2.2.2.5 Net Rate due to Photo-ionization and Radiative Recombination

Rate coefficients given in equations (2.31), (2.26), and (2.27) can describe the net change rate for atoms in the $(i\ell)$ level due to photo-ionizations and radiative recombinations, given by

$$\begin{aligned}\mathcal{P}_{i\ell} = \sum_{jm} & N_{jm}\left[p_{j\rightarrow i,m\rightarrow\ell}\left(T_r\right) + N_e\left(r_{j\rightarrow i,m\rightarrow\ell}\left(T_e\right) + q_{j\rightarrow i,m\rightarrow\ell}\left(T_e,T_r\right)\right)\right] \\ & -N_{i\ell}\left[p_{i\rightarrow j,\ell\rightarrow m}\left(T_r\right) + N_e\left(r_{i\rightarrow j,\ell\rightarrow m}\left(T_e\right) + q_{i\rightarrow j,\ell\rightarrow m}\left(T_e,T_r\right)\right)\right].\end{aligned} \tag{2.45}$$

2.2.3 Constructing the Rate Matrix

Given the above equations, the CR equations can be written as

$$\frac{\mathrm{d}N_{i\ell}}{\mathrm{d}t} = \mathcal{C}_{i\ell} + \mathcal{P}_{i\ell} + \mathcal{A}_{i\ell} + \mathcal{S}_{i\ell} + \mathcal{U}_{i\ell}, \tag{2.46}$$

where each term represents the net contribution for a given process and its inverse to the total time rate of change of population $N_{i\ell}$ as prescribed in 2.2.2. The system is linear with respect to the populations and additionally can be written as the product of a matrix and vector,

$$\frac{\mathrm{d}\mathbf{N}}{\mathrm{d}t} = \underline{R}\left(F, G, N_e\right)\mathbf{N}, \tag{2.47}$$

where $\mathbf{N}$ is the population vector and $\underline{R}$ is the rate matrix, which is a function of the electron and photon energy distributions. Given the atomic processes described earlier in this chapter, the rate matrix has a block tri-diagonal structure with non-uniform blocks. Neither excitation nor de-excitation processes change the number of bound electrons, thus their contributions necessarily lie along the block diagonal. The ionization and recombination processes described change the number of bound electrons by one. This means that the block diagonal matrices are only coupled to their neighboring blocks, leading to a block tri-diagonal structure.

2.2.4 Solving the Rate Equations

A priori knowledge of the block tri-diagonal structure of the rate matrix can be used not only to save memory, but also to apply methods specifically designed for such systems. A more detailed discussion of the reasoning behind algorithmic choices is given in the implementation chapter. While algorithms exist that allow for block matrices to be processed in parallel, it was decided to go with the Thomas algorithm. This algorithm, while sequential, has the lowest total complexity and memory requirements. Additionally, parallelism can be exposed within the linear algebra functions themselves. This combination, when compared with other block tri-diagonal solvers, yields lower memory requirements, lower computational complexity, and exposes low level parallelism.

3. IMPLEMENTATION DETAILS

This work resulted in the production of a library known as `inlinlte`. This library in written primarily in C++ with an application programming interface (API) compatible with the C, C++, FORTRAN, and python languages. `inlinlte` utilizes the `Kokkos` library to obtain intranode parallelism, the message passing interface (MPI) for internode parallelism, and `libquo` for dynamic process mapping. The use of MPI, `Kokkos`, and `libquo`, will be discussed in more detail in this chapter. In terms of algorithms used, `inlinlte` uses the Brent-Dekker nonlinear solution method in solving for the electron density, the Thomas algorithm for the block tri-diagonal linear solver, and an automatic adaptive 8-panel Newton-Cotes quadrature rule for photoionization integrals. `inlinlte` leverages modern tools and practices to maintain correctness, performance, and developer productivity. The code is hosted on LANL's internal gitlab page and uses git for version control. Google's test framework is used for unit and integration tests to make sure individual components are operating correctly as well as ensuring that `inlinlte` is able to obtain correct charge states and monochromatic opacities. Performance, though not automated, is extensively tested with intel's VTune, intel's vectorization advisor, and the Kokkos performance tools. The tools provided through the `Kokkos` ecosystem also provide profiling to ensure kernel runtimes do not increase. A build of `inlinlte` without optimization, without OpenMP, and without MPI, has also been run with valgrind to ensure that no memory leaks are present.

3.1 Performance Portability

On the path to exascale computing, the department of energy (DOE) laboratories have committed to purchasing supercomputers whose compute nodes have heterogeneous compute resources. Advanced technology system (ATS) 1, known as Trinity at Los Alamos National Laboratory (LANL), will have compute nodes that use the many core intel knights landing (KNL) processor. In contrast, ATS 2, known as Sierra at Lawrence Livermore National Laboratory (LLNL), will have compute nodes that will use multi-core IBM Power 9 central processing units (CPU) and nVIDIA Volta graphics processing unit (GPU) coprocessors. ATS1 and ATS2 will have compute nodes with very different architectures and requirements for performance. Despite differences, it is a stated goal of the DOE to have single source code bases capable of running on both machines in a performant manner. This is a very difficult problem in performance portability and software developers at SNL attempted to solve this problem with the `Kokkos` library.

Initially a package within the Trilinos solver library [12], `Kokkos` is a C++ library which includes parallel patterns and data structures that allow for efficient mapping of work and memory to a targeted hardware backend. The parallel patterns included are `parallel_for`, `parallel_reduce`, and `parallel_scan` which will efficiently map parallel work to computing resources. `Kokkos` has also recently added dynamic task directed acyclic graph (DAG) and static work DAG capabilities [5]. The `parallel_{for,reduce,scan}` patterns intake an iteration index range and an object containing the work to be done for each iteration index. These patterns can be nested to allow for hierarchical parallelism. The outermost level of parallelism is a league-team policy in which teams of threads are launched simultaneously to perform parallel work. Each team has at its disposal a number of threads that can be set by the user, but can also be automatically set by the runtime to a number suited for the hardware. In addition, each thread within a team can utilize vector processing units (VPU). Thus `Kokkos` contains up to 3 levels of hierarchical parallelism that can be tuned to specific hardware types: the `TeamPolicy`, `TeamThreadRange`, and

`ThreadVectorRange`. In addition to parallel patterns with hierarchical parallelism, multidimensional arrays called views are provided as these are common in scientific computing. Views separate indexing from data layout allowing for data layouts can change without changing the access signature. Backend specific default layouts are designed to yield coalesced loads on GPUs and cache efficiency on CPUs [6].

Other parallel runtime systems and libraries were investigated and eventually ruled out. Investigations in parallelization started with looking into the use of the nVIDIA's compute unified device architecture (CUDA) platform. Due to its asymptotic complexity, the linear solver was the first component to be investigated. Three different block tri-diagonal solver algorithms were selected for investigation: Thomas algorithm, cyclic reduction, and parallel cyclic reduction [13, 20, 21, 11]. These algorithms all require certain BLAS level 2 and 3 algorithms for block matrix operations. Necessary linear algebra functions include matrix-matrix multiply, LU decomposition with partial pivoting, solving a triangular matrix (for use with LU decomposition), and matrix-vector multiply. All of the listed algorithms are implemented in cuBLAS, which is a CUDA capable library for dense linear algebra. While this showed useful speedup for the low Z, high fidelity atomic models that were tested, it proved difficult to hide data transfer costs for reduced atomic models. This issue arose when calculating the rate coefficients.

Another approach to exposing parallelism is done by running the calculation for each cell in a spatial mesh concurrently. This was implemented via the open multiprocessing (OpenMP) intranode parallelization API in which each thread would perform the calculation on a "batch" of cells. The batch size would be based on the size of a targeted memory space such as an L2 data cache. Many BLAS libraries such as cuBLAS, MAGMA, and MKL provide a batched API that would allow for a linear algebra operation to be applied to a batch of data which would reduce kernel invocation overhead and targeted for the large number and relatively small sized matrices that are common in rDCA atomic models. A requirement of the batched implementations of linear algebra operations is that all matrices and vectors must be the same size across all elements of a batch. There is no guarantee that block matrices in different block rows will have the same size, which would be necessary if attempting to use this API for cyclic reduction or parallel cyclic reduction. This implementation is limited to treating only a single element with NLTE physics. Cells within the mesh that will be treated with NLTE physics will therefore all have block tri-diagonal systems of the same size and structure. Thus, an element of a batch maps to a cell in the mesh. This means that using batched BLAS calls, a block tri-diagonal solver algorithm can be applied to a batch of block tri-diagonal matrices simultaneously. Due to exploiting parallelism at a higher level, the sequential Thomas algorithm was chosen as the solver of choice as it had the lowest memory requirements and asymptotic complexity [21]. A potential issue of this approach is that cells in the mesh with different conditions will converge to an electron density at different rates. The use of continuum lowering to account for pressure ionization effects also precludes the use of batching because it breaks the assumption that block matrix sizes are the same across all cells. It was also shown that dynamic on node work scheduling is necessary to help reduce thread idle time as well as the overall wall clock time. This combination of conditions made `Kokkos` the ideal candidate for `inlinlte`. This switch precluded the use of the batched APIs provided by MAGMA, MKL, and cuBLAS.

3.2 MPI+Threads With `libquo`

In practice, `inlinlte` uses the OpenMP backend of `Kokkos`. However, this calculation is one of a large number of different calculations that are part of a RH or

TRT simulation. Many of the production level RH codes used by DOE laboratories have been in use for an extended period of time and were developed when single program multiple data (SPMD) via MPI, and only MPI, was sufficient for performance on state of the art supercomputers. Many of the production codes run most efficiently when running 1 MPI process per compute *core*. A naïve implementation of `inlinlte` would restrict 1 process per compute *node* in order to allow OpenMP to utilize all available resources. This would incur an unacceptable slowdown for the rest of the application. `libquo` can be used to dynamically set optimal MPI policies for libraries that have different requirements [10]. This allows for the RH portions of a calculation to use 1 MPI rank per core and NLTE calculations to use 1 MPI rank per node, which is the preferred mode of operation for each calculation.

When an opacity is requested, all active MPI ranks on a node gather the input data to a single MPI rank. After the gather has occurred, the gathered to rank performs the calculation while the remaining ranks are waiting at a barrier, thus leaving their hardware resources available. The gathered rank is able to spawn threads on the other ranks' resources. Once the calculation has completed, the active rank reaches the barrier, all ranks become active again, and output data is scattered. This process is illustrated in figure 3.1.

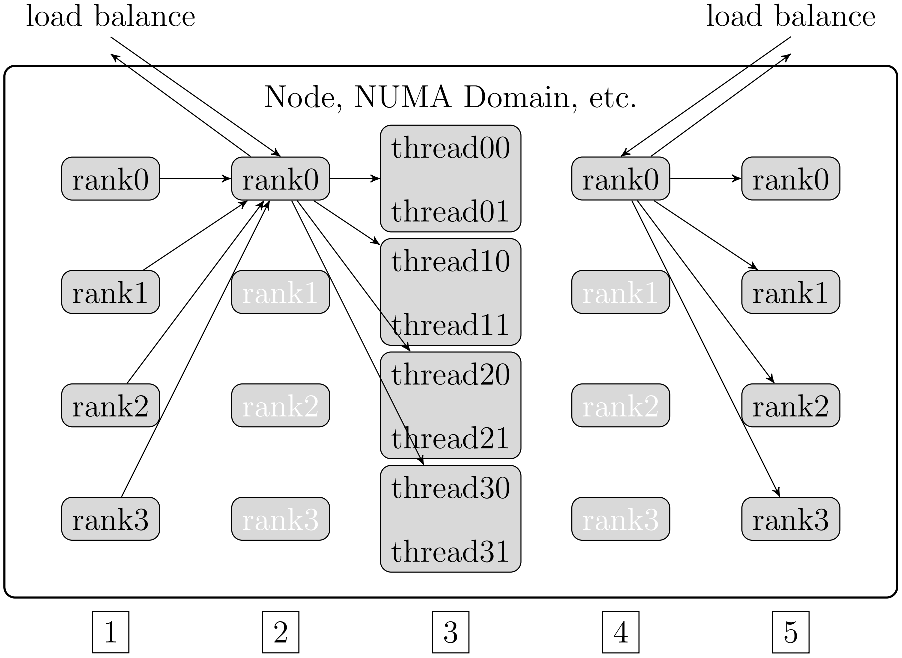


Figure 3.1: 1. All ranks gather input data to rank0. 2. All ranks but rank0 barrier, relinquishing their hardware resources. rank0 carries out global load balance procedure. 3. rank0 spawns threads on all available hardware resources, taking advantage of hyperthreads when available and the calculation is performed. 4. rank0 redistributes both local and non-local output data to other ranks. 5. all ranks reach the barrier and become active, reaching their initial state.

While the current release version of `inlinlte` is limited to using a 1 rank per node layout after the remap, that requirement could be relaxed on nodes with larger numbers of cores. With features of an upcoming release of `libquo`[1], using 1 rank per NUMA domain and similar setups would be possible. This generality in addition to the local behavior of this strategy should ensure `inlinlte`'s on-node scalability. Other strategies are possible, some of which would preclude the use and cost of the dynamic process mapping provided through `libquo`. One such strategy would be retain the 1 rank per core mapping and spawn OpenMP threads on that core,

[1] See issue #30 on `libquo` github for more information.

potentially taking advantage of hyperthreads where available. In order to prevent incurring the memory cost of storing one copy of the atomic model per rank, one would need to use MPI shared memory functionality, which is a relatively new feature of the MPI standard with varying support across implementations. More importantly, this strategy would not allow for on-node load balancing that occurs through shared memory dynamic scheduling. The cost of an NLTE calculation can vary based upon the input conditions. This occurs because the number of iterations required to converge the electron density is inversely proportional to the temperature. To show this disparity, a set of input conditions, shown in figure 3.2 were run using both static and dynamic scheduling.

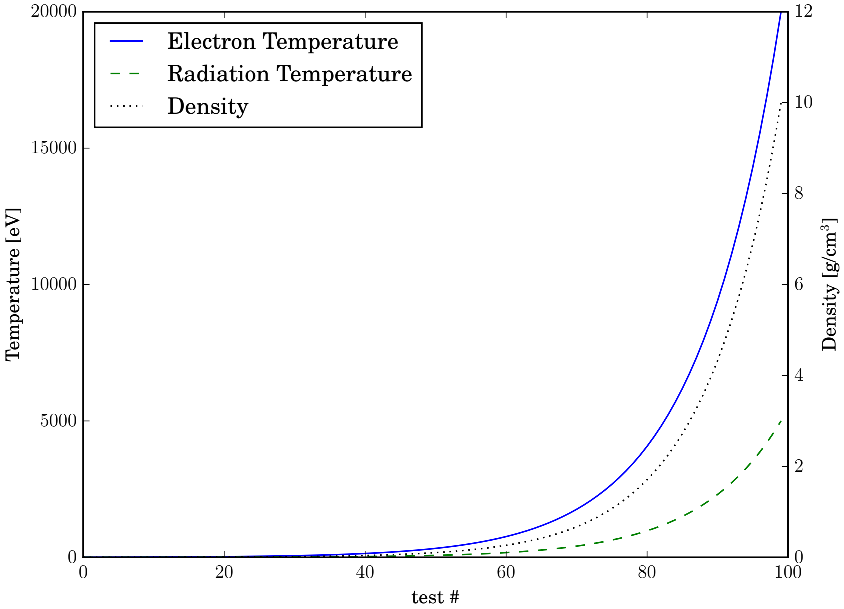
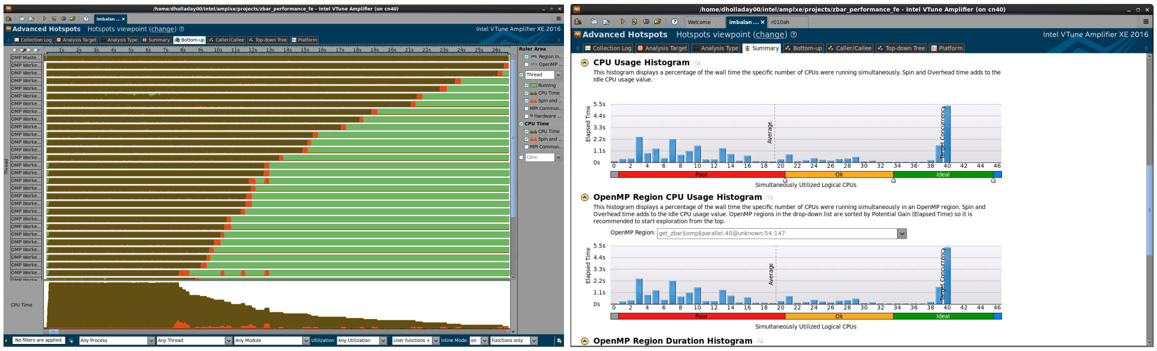


Figure 3.2: Distribution of (ρ, T_r, T_e) input conditions. Note that temperature units are on the left vertical axis and density units are on the right vertical axis.

These runs were analyzed by intel's vtune profiling tool. The analysis performed by vtune revealed that many of the threads were idle for much of computation because while every thread had the same number of input conditions, certain inputs took longer than others. This resulted in an average thread utilization of 20 out of 40 threads, as shown in figure 3.3 below.



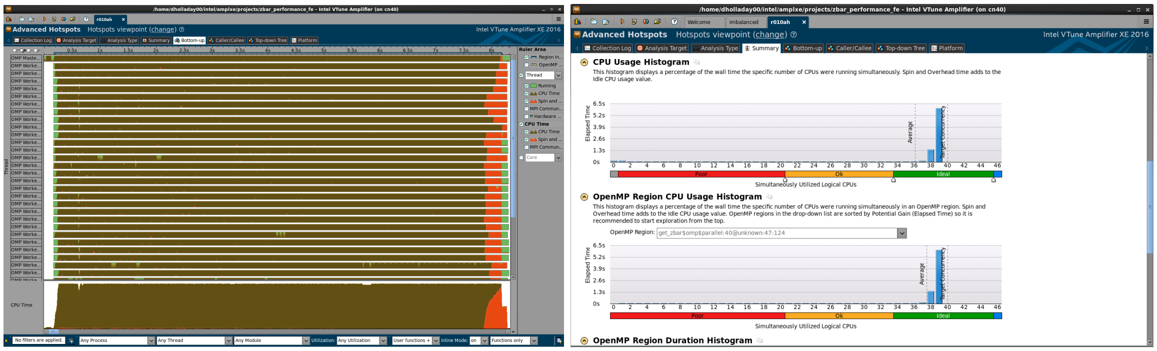
This histogram displays a percentage of the wall time the specific number of CPUs were running simultaneously. Spin and Overhead time adds to the Idle CPU usage value.

This histogram displays a percentage of the wall time the specific number of CPUs were running simultaneously in an OpenMP region. Spin and Overhead time adds to the Idle CPU usage value. OpenMP regions in the drop-down list are sorted by Potential Gain (Elapsed Time) so it is recommended to start exploration from the top.

OpenMP Region: get_zbar$omp$parallel:40@unknown:54:147

Figure 3.3: Left: For some threads, significant amount of idle time; Right: CPU usage histogram reveals about 50% average utilization of resources.

Dynamic scheduling breaks the iteration space of the parallel for loop into smaller chunks, allowing each thread to process a chunk of the iteration space before requesting an additional chunk of work. This has an overhead penalty associated with it but it is easily amortized due the large amount of work required for even a single NLTE calculation. When dynamic scheduling was used to perform the NLTE work, the result was that each thread was actively working for a greater proportion of the calculation. This resulted in a 60% improvement in calculation time and an average thread utilization of 36 out of 40 threads, as shown is figure 3.4 below.



This histogram displays a percentage of the wall time the specific number of CPUs were running simultaneously. Spin and Overhead time adds to the Idle CPU usage value.

This histogram displays a percentage of the wall time the specific number of CPUs were running simultaneously in an OpenMP region. Spin and Overhead time adds to the Idle CPU usage value. OpenMP regions in the drop-down list are sorted by Potential Gain (Elapsed Time) so it is recommended to start exploration from the top.

Figure 3.4: Left: Majority of threads active for entire calculation; Right: CPU usage histogram reveals about 90% average utilization of resources.

Dynamic scheduling presents a compelling reason to use OpenMP across the entire node. This reason, combined with the memory benefits of only requiring a single copy of the atomic model per node, is why ultimately the strategy shown in figure 3.1 was chosen.

3.3 Global Load Balance

A commonly used practice among radiation hydrodynamics codes that incorporate NLTE physics is to apply physics only for spatial zones above a certain temperature. Standard spatial domain decomposition procedures can lead to load imbalance of NLTE computational work. Because of this, `inlinlte` internally performs global load balance of NLTE work for effective hardware utilization. Global load balancing is executed after on-node data gathering has occurred. This means that each vertex in the graph is equivalent to a compute node. The load balancing algorithm used has ideal and non-ideal variants, the latter being based on the former. These will be discussed below, starting with the ideal case.

3.3.1 Ideal Case

The ideal case requires a number of conditions to be met. All participating compute units must have identical performance characteristics and the number of total compute units must be a power of 2. If those conditions are met, the method has the benefit of not requiring any global communication and has logarithmic complexity. It is a balanced binary tree approach as described by O'Brien, et al. for use in particle transport applications and illustrated in figure 3.5 [16].

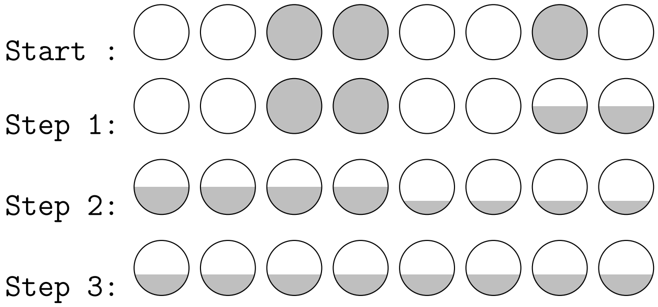


Figure 3.5: Start with imbalanced workload, after $\log_2(8) = 3$ steps, workload perfectly balanced.

3.3.2 Non-Ideal Case

In order to relax the strict requirements of the ideal case, a more robust version of this method is used in practice. This method allows for an arbitrary number of compute units and heterogeneous performance characteristics. This method requires assigning a weight to each compute unit. The weight is proportional to the performance of the compute unit. One way to implement the weighting scheme is to set the weight of the compute unit equal the rate at which it processes work. At initialization time, this weight can be calculated as

$$w_i = \frac{N_{\text{items}}}{N_{\text{runs}}} \sum_{j=1}^{N_{\text{runs}}} \frac{1}{t_j}, \tag{3.1}$$

where N_{items} is the number of NLTE cells used to evaluate node performance, N_{runs} is the number timing measurements taken, and t_j is a single timing measurement. N_{items} should be large enough to saturate node resources and hide parallel launch overheads and N_{runs} should be large enough to dampen noise associated with computation timings. This method has embedded the assumption that NLTE work is equal across cells. It can be shown that the number of non-linear iterations required is inversely proportional to temperature. However, `inlinlte` implements on node dynamic scheduling of work which helps alleviate much of the issues. Additionally, constructing the rate matrix via the matrix polynomial method reduces the time per non-linear iteration by increasing performance of rate matrix construction. This means that relatively more time is spent precomputing photo-ionization rate coefficients as opposed to the non-linear solve, meaning that the number of non-linear iterations has less overall effect on total time of NLTE work. While improvements could be made to the weighting scheme, this method appears to be sufficient for the purposes of this work.This weighting scheme provides a mapping between work and time, allowing the algorithm to balance the load such the time taken on every compute unit is roughly equal. By defining the timing estimate of compute resource i at iteration k as

$$T_i^{(k)} = \frac{N_i}{wi} \tag{3.2}$$

A measure of imbalance at iteration k, denoted as $\iota^{(k)}$, can be defined as

$$\iota^{(k)} \equiv \frac{\max_i(T_i^{(k)})}{\min_i(T_i^{(k)})}, \tag{3.3}$$

$\iota = 1$ represents perfect load balance. Through the weighting scheme used, this means that each compute resource should take the same time to finish, regardless of its particular performance characteristics. Checking this measure against a stopping criterion also introduces inter-node communication as finding the minimum and maximum time estimates is implemented via an `MPI_Allreduce`. However, if it can be shown that all compute units have approximately the same performance characteristics and that the number of compute units is a power of 2, then the algorithm can relax to the ideal case. The ideal case is guaranteed to balance the workload in a single iteration and doesn't require any global communication.

3.3.3 Data Redistribution

After the NLTE calculation has completed, the results must be returned to the original compute node on which it originated. This is currently implemented by inverting the load balance procedure. Every rank stores every rank and the number of items transferred to or from that rank. This list is iterated through in reverse order. This method has the benefit in which every message destination and size is known a priori. The sign of the message size is used to determine if the rank will be sending or receiving. If a rank needs to perform multiple consecutive sends or receives in the sequence, those operations can be done in an asynchronous fashion. Thus, the list of transfers and sizes can be preprocessed and split into spans in which the message sizes have the same sign and each span can be iterated through using non-blocking sends and receives and only needing to wait after all sends and receives in that span are completed. In the worst case, this method is equivalent to using blocking sends and receives but also offers great benefit in certain pathological cases.

1. Group items into spans such that each the message sizes have the same sign in each span
2. For each span
    (a) For each element of the span
        i. perform non-blocking send or receive
    (b) wait for all messages in span to post

The above implementation works for ideal and non-ideal cases, and never requires global communication. Other inversion methods exist, but were not explored as the chosen method proved to be performant, scalable and is easy to implement.

3.4 Data Structures

`Kokkos` provides multidimensional arrays called views. Views are able to reside on a specified memory space, such as GPU memory. Individual elements are accessed in the same way as `FORTRAN` multidimensional arrays $A_{ij...} \rightarrow A(i, j, ...)$. Additionally, their layout and access patterns are separate and can be customized. The standard layouts are multidimensional generalizations of row major and column major. Row major becomes a right layout, in which the right most index in the accessor is unit stride. Column major becomes a left layout in which the left-most index in the accessor is unit stride. One additional constraint on views is that their span must be given by a cartesian product. This is not amenable for storing block matrices whose size varies between block rows as is the case with the rate matrix. While compressed storage formats would be beneficial, it would disallow standard LAPACK implementations from being used in the solver. However, views of views are possible and this is what is used. To store block matrices, a 1D view of 2D views is used. Three such data structures are used to specify a block tri-diagonal matrix, one for each non-zero block diagonal. For RHS vectors and permutation matrices needed for partial pivoting, 1D views of 1D views are used. The data structures allow for the tri-diagonal solver algorithm to be written concisely.

3.4.1 BLAS/LAPACK Interface

A wrapper interface for the linear algebra functions required by the tri-diagonal solver algorithm was developed. The functions required are matrix-matrix multiply (`gemm`), matrix-vector multiply (`gemv`), LU decomposition (`getrf`), and solve (`getrs`). This wrapper interface must be called at the thread team level within `Kokkos`'s hierarchical parallelism framework. Standard BLAS/LAPACK implementations are supported including those offered by intel, netlib, and a hand-coded internal implementation. On CPUs, intel's math kernel libraries (MKL) offering offers great performance and is supported on computer systems used by `inlinlte` and is the default option. Other options on the CPU are possible and some are able to provide similar performance. However, these libraries are currently CPU only and not portable to the GPU. The hand-coded implementation is fully portable and does not have any additional external dependencies, allowing for `inlinlte` to be fully portable to any hardware backend supported by `Kokkos`. Additionally, this implementation was written specifically to be called from the team thread level of a `Kokkos` kernel and takes advantage of thread and vector level parallelism where possible. However, this implementation lacks some hardware specific optimizations that other implementations have such as knowledge of cache sizes and its performance will suffer when compared to the highly optimized CPU implementations, specifically for larger matrices where cache blocking can provide significant performance boosts. Additionally, the hand-coded implementation does not implement partial pivoting for LU decomposition which results in decreased robustness. Given these caveats, the hand-coded implementation is generally only used when other options are not available.

3.4.2 Atomic Model Storage

All data pertaining to a specific atomic model is also stored using `Kokkos` views, some of which have a mixture of dimensions which are known at compile time and others that are known at runtime. This is supported and can give additional performance because certain stride calculations can be performed at compile time. Data associated with an atomic model is constant and does not always follow sequential access patterns. Another benefit of `Kokkos` views is that both of these attributes can be expressed and used to increase performance. On CPU backends, constant data can help compilers more easily reason about possible optimizations and vectorization. On the GPU, the benefits are ever more substantial because the atomic data can take advantage of the specialized texture memory available on GPUs which can be accessed randomly without as much performance degradation [6].

3.5 NLTE $\langle Z \rangle$ Solution Procedure

The solution procedure outlined in this section is specific to the steady-state case in which the mass density is an externally imposed constraint. Solution requires the following:

1. Calculate precomputed rate coefficients (if any)

2. If matrix polynomial method, calculate matrix and RHS coefficients.

3. Brent's method, for each iteration:

   (a) Calculate rate matrix given input N_e

   (b) Solve Rate matrix

   (c) Calculate residual $R = \frac{\rho N_A}{M} - \sum_{i\ell} N_{i\ell}$

4. Post process population and calculate $\langle Z \rangle = \sum_{i\ell} i N_{i\ell}$

5. If continuum lowering enabled recalculate, modify data structures appropriately

   (a) Recalculate rate coefficients (if need to be recalculated)

   (b) Redo Brent's method with continuum lowered populations

   (c) Recalculate $\langle Z \rangle$

From an algorithmic standpoint, the pattern for adding the contribution to the rate matrix for any given atomic process is the same. This realization allowed for writing a matrix population function templated on atomic process type. Each process need only know how to calculate matrix indices and rate coefficients. This function can also be used in the matrix polynomial method as there is a known electron density dependence for each process. Thus each process can populate the corresponding polynomial coefficient matrices using an electron density of 1 to obtain the matrix and RHS coefficients. This can also be used to optionally precompute and store rate coefficients. In the event that the matrix polynomial method is too memory intensive, precomputed rate coefficients can provide reasonable performance with a smaller memory footprint. Due to numerical integrals necessary in calculating photo-ionization rate coefficients, they should be precomputed if possible. Using a numerical integration routine templated on the integrand proved to be much more performant compared to a routine that relied on function pointers. This is because the integrands could be inlined at compile time and loops over quadrature points could be vectorized. This is not possible unless the integrands are able to be inlined at compile time. The other processes can have their rate coefficients precomputed, but with limited benefit per byte relative to that gained by precomputing photoionization rate coefficients. The use of template metaprogramming has enabled a much more readable and maintainable code-base that is just as performant due to the ability to inline function calls at compile time. A future improvement would be to use explicit template instantiation to improve compile times and decrease compiled file sizes. Current compile times for the backends used in `xRAGE` do not warrant these changes. Additionally, this methodology is extensible in that more atomic processes could be added in the future with minimal code changes.

3.5.1 Pruning the Populations

In cases where the rate matrix is poorly conditioned, the steady state populations can be negative, which is unphysical to have a negative number density. An example when this can occur is at very low temperatures, when almost all atoms reside in the neutral ground state. Due to its unphysical nature, negative number densities are replaced with zero. An electron density can be calculated from the modified populations. If this estimate is non-zero, the populations are renormalized to ensure conservation of electron number. This normalization results in a slightly different set of populations, which can be summed to calculate a total number density. Based on the newly calculated number density, the populations are again renormalized to ensure conservation of ion number. A new ion number density and electron number density can be calculated from the final set of populations. The current choice for $\langle Z \rangle$ is

$$\langle Z \rangle = \begin{cases} \frac{\sum_{i\ell} i N_{i\ell}}{\sum_{i\ell} N_{i\ell}} & , \ \sum_{i\ell} i N_{i\ell} > 0 \\ \frac{N_e}{N_{tot}} & , \ \text{otherwise.} \end{cases} \tag{3.4}$$

The conditional exists to provide a fallback when the electron density is near 0. It has been noted previously that NLTE physics is usually only being used in high temperatures, meaning that the more rigorous definition will be the most commonly used definition.

3.5.2 Residual Choices

Simulations which have the material density ρ as a state variable place a constraint on the populations. The total number density can be calculated from the material density. In the case of single element, the number density is:

$$N_{\text{tot}} = \frac{\rho N_A}{M}, \tag{3.5}$$

where N_{tot} is the total number density, N_A is Avogadro's number, and M is the molar mass of the material. The number conservation constraint is given by

$$N_{\text{tot}} = \sum_{i\ell} N_{i\ell}. \tag{3.6}$$

The above is only true when a single material is present. The presence of multiple materials presents a much more challenging problem and is beyond the scope of this work. The populations are a function of the electron density and only a single value of the electron density, N_e, will satisfy the condition given in equation (3.6). However, the electron density can also be directly calculated from the populations as well through conservation of electron number. This constraint is given by

$$N_e = \sum_{i\ell} (i-1)\, N_{i\ell}. \tag{3.7}$$

For a given set of fixed electron and radiation energy distributions, commonly described via electron and radiation temperatures, the coupled system of rate equations only depend on the electron density. Once an electron density is specified, populations can be obtained vi a linear solve. When solving for the steady state populations (in which all net rates are 0), the system is over-constrained. Suppose there are P populations, this implies a $P \times P$ rate matrix. The electron density is also an unknown, indicating a total of $P+1$ unknowns. However, there are 2 additional constraints which would imply a total of $P+2$ equations for $P+1$ unknowns. One of the equations and unknowns can be removed from the rate matrix, making it a $P-1 \times P-1$ matrix. The resultant solution vector $\mathbf{y}$ is normalized by the removed unknown as follows

$$y_{i\ell} = \frac{N_{i\ell}}{N_{jk}},\ \forall\ i \neq j,\ \ \ell \neq k. \tag{3.8}$$

One of the constraints must then be used to solve for N_{jk}. The 2 choices are given by

$$N_{\text{tot}} = \sum_{i\ell} N_{i\ell} = \sum_{i\ell \neq jk} N_{i\ell} + N_{jk} = N_{jk} \left(\sum_{i\ell \neq jk} \underbrace{\frac{N_{i\ell}}{N_{jk}}}_{y_{i\ell}} + 1 \right), \tag{3.9}$$

$$N_e = \sum_{i\ell} (i-1)\, N_{i\ell} = \sum_{i\ell \neq jk} (i-1)\, N_{i\ell} + jN_{jk} = N_{jk} \left(\sum_{i\ell \neq jk} (i-1)\, \underbrace{\frac{N_{i\ell}}{N_{jk}}}_{y_{i\ell}} + j \right). \tag{3.10}$$

These choices will necessarily satisfy one of the given constraints. The remaining constraint will be satisfied if the electron density chosen is close to the true electron density. The remaining constraint can be reformed as a residual such that when the residual is less than a specified tolerance, the electron density is said to have converged. The previously implemented choice was

$$N_{jk} = \frac{N_e}{\sum_{i\ell \neq jk} (i-1)\, y_{i\ell} + j}, \tag{3.11}$$

$$R = N_{\text{tot}} - N_e \frac{\sum_{i\ell \neq jk} y_{i\ell} + 1}{\sum_{i\ell \neq jk} (i-1)\, y_{i\ell} + j}. \tag{3.12}$$

This method was shown to have issues when the electron density was small, in cases when very little ionization has occurred. This is due to the fact that $\lim_{N_e \to 0} R = N_{\text{tot}}$. Additionally, $\lim_{N_e \to 0} N_{jk} = 0$, which implies that all of the populations are 0. However, one is also free to choose the following

$$N_{jk} = \frac{N_{\text{tot}}}{\sum_{i\ell \neq jk} y_{i\ell} + 1}, \tag{3.13}$$

$$R = N_e - N_{\text{tot}} \frac{\sum_{i\ell \neq jk} (i-1)\, y_{i\ell} + j}{\sum_{i\ell \neq jk} y_{i\ell} + 1}. \tag{3.14}$$

This choice does not suffer from the same issues in $\lim_{N_e \to 0} \{N_{jk}, R\}$. This method has been shown to be numerically stable for a broader extent of the input phase space (ρ, T_e, T_r) in the `inlinlte` code.

Additionally, when physical bounds are placed on the electron density guesses, it can be shown that the residual moves from negative to positive as the electron density increases. However, if the mass density is high, the upper bound electron density is also very high, which can lead to large disparities between electronic process rates and photonic process rates. Specifically, electron impact ionization has quadratic dependence on the electron density. Large disparities can present numerical precision issues that must be addressed. Since this is an iterative method, many of these issues will disappear near the neighbourhood of convergence when the electron density used to calculate the rates is unphysically high. However, this issue can manifest when attempting to calculate the residual at the physically dictated upper and lower bound limits of the electron density, which are used to bracket the residual. Bracketing in this context is the notion that if a smooth function changes sign on an interval, a zero of that function must lie in the interval. The numerical precision issues in these high density regimes can result in the upper bound residual having the incorrect sign which will result in a failure for the Brent-Dekker method. Since there are physical arguments for what the sign of the residual should be, checks can be put in place to guard against this. This specific issue was encountered when the density of the iron exceeded 75 g/cc in a simulation described in section 6.1.

3.5.3 Inverse Table Lookup Method

As currently implemented in `xRAGE`, only LTE tabulated opacities are used, but the location within the table is augmented through the NLTE $\langle Z \rangle$. This methodology follows the work of Busquet [3]. With current rDCA models, even this method was very expensive, more than 2 orders of magnitude slower than an LTE equivalent simulation. It's efficacy is thus not well known and further research in this area is a key aspect of this work. The table lookup is augmented by obtaining the LTE temperature which corresponds to the NLTE $\langle Z \rangle$. This modified temperature is used in the table lookup to obtain the augmented opacities.

3.6 Block Tri-diagonal Solver

While other methods for solving the block trigiagonal coupled linear system were investigated, the Thomas method was ultimately chosen. Three different block tri-diagonal solver algorithms were selected for investigation: Thomas algorithm, cyclic reduction, and parallel cyclic reduction [13, 20, 21, 11]. As implemented, these algorithms all require certain BLAS level 2 and 3 algorithms for the block matrices. Linear algebra functions needed include matrix-matrix multiply, LU decomposition with partial pivoting, solving a triangular matrix (for use with LU decomposition), matrix-vector multiply. All of the listed algorithms are implemented in cuBLAS, which is a CUDA capable library for dense linear algebra. While this showed useful speedup for low Z, high fidelity atomic models, it proved difficult to hide data transfer costs for reduced atomic models. CUDA kernels were written to calculate the rate coefficients to investigate their speedup as well. Hiding data transfer and kernel invocation overhead on reduced atomic models such as would be used inline in a RH calculation proved difficult. It was decided to investigate parallelism across cells in the mesh as its implementation would be much more straightforward.

4. CODE PERFORMANCE

As previously stated, the additional computational work required to incorporate all of the atomic physics necessary to relax the assumption of LTE is very large. When used in `xRAGE`, it has been shown to regularly take 90% of the total computation time and represent more than an order of magnitude increase in total computational time when compared to LTE simulations with all other input parameters held constant. Focusing on improving the computation time is thus of high importance if NLTE effects are to be studied in real physical systems of interest such as ICF.The remainder of this chapter will focus on detailing code performance and the strategies and tools used to employ said improvements.

4.1 Rate Matrix Computation

When using reduced atomic models such as rDCA, matrix sizes tend to vary form $\mathcal{O}\left(10-100\right)$. It turns out that in this regime, calculating the rate matrix elements, an $\mathcal{O}\left(N^2\right)$ complexity procedure, and matrix factorization, an $\mathcal{O}\left(N^3\right)$ complexity procedure, can be similar due to the difference in proportionality coefficients associated with each. Highly optimized linear algebra libraries are used to ensure that linear algebra operations on the rate matrix are performant. Further optimization of matrix solvers will unlikely yield in significant performance gains. Thus effort was spent in increasing the performance of populating the rate matrix. Since this is being integrated into multi-physics codes with users who might not be atomic physicists or NLTE domain experts, user facing memory control was greatly simplified to facilitate ease of use. Rate matrix computation options are able to be set in `xRAGE` input files. A list of `inlinlte` specific options are listed in table 4.1.

Table 4.1: List of `inlinlte` relevant input file options for use in `xRAGE`.

Name	Values	Default	Explanation
`inlinlte_max_iterations`	int	50	Maximum number of iterations in non-linear solver.
`inlinlte_integration_tol`	real	10^{-4}	Tolerance for P.I. rate integrals
`inlinlte_non-linear_tol`	real	10×integration tol.	Non-linear solver relative tolerance
`inlinlte_memory_option`	0-6	0	Rate Matrix: 0 = matrix polynomial 1 = store P.I. RC's 2 = store all photon process RC's 3 = Option 2 + C.E. RC's 4 = Option 3 + C.I. RC's 5 = Store all RC's 6 = calculate everything in place [1]

These options have varying memory cost and one should in general use the fastest algorithm that can be afforded given memory constraints. These options provide a set of space-memory tradeoffs to satisfy a broad set of problems and gives `inlinlte` the highest opportunity possible to be utilized. The cost of calculating the rate coefficients for a given atomic process varies between processes. Photoionization is especially expensive due to the fact that integrals over photon energies are computed numerically. Figure 4.1 shows the speedup associated with storing only the photoionization rate coefficients as opposed to recalculating them every time they are needed. More than 100% improvement is observed in all 3 test cases.

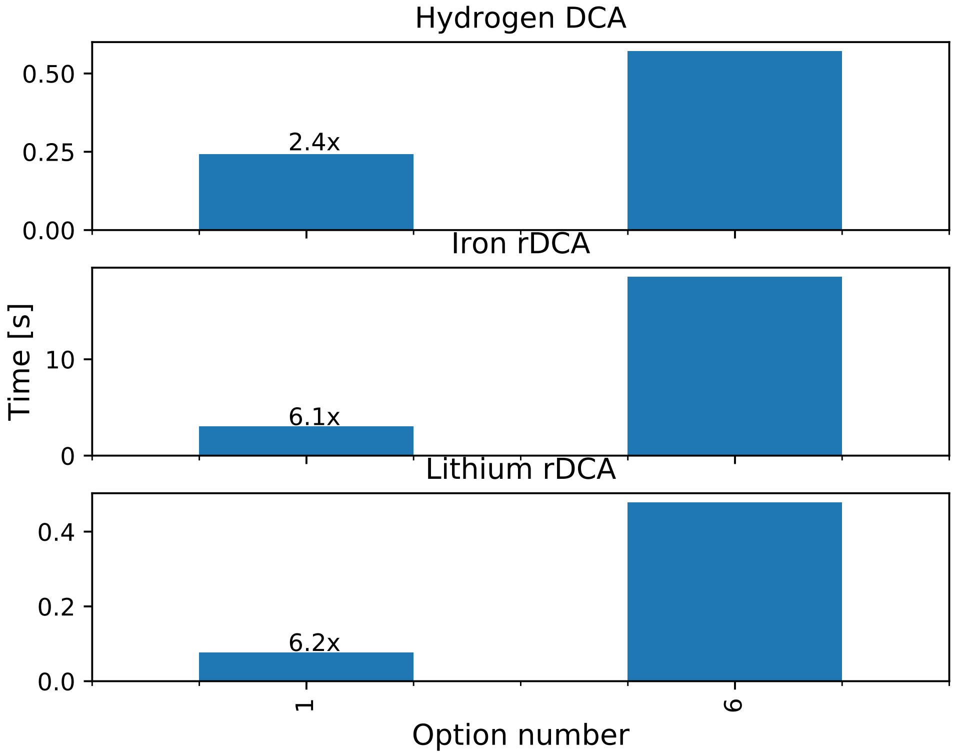


Figure 4.1: Comparing memory option 1 to memory option 6 shows significant differences in performance. It is unlikely that option 6 would provide benefits over other memory options.

The relative worth of precomputing the rate coefficients for other processes is not as notable as for photoionization, but could still prove very useful. Additionally, the matrix polynomial method performance shown. All speed comparisons are relative to using option 1. In addition, memory requirements for each option is also shown in figure 4.2. Also of note that the memory requirements are strongly dependent upon atomic model complexity as well as proton number.

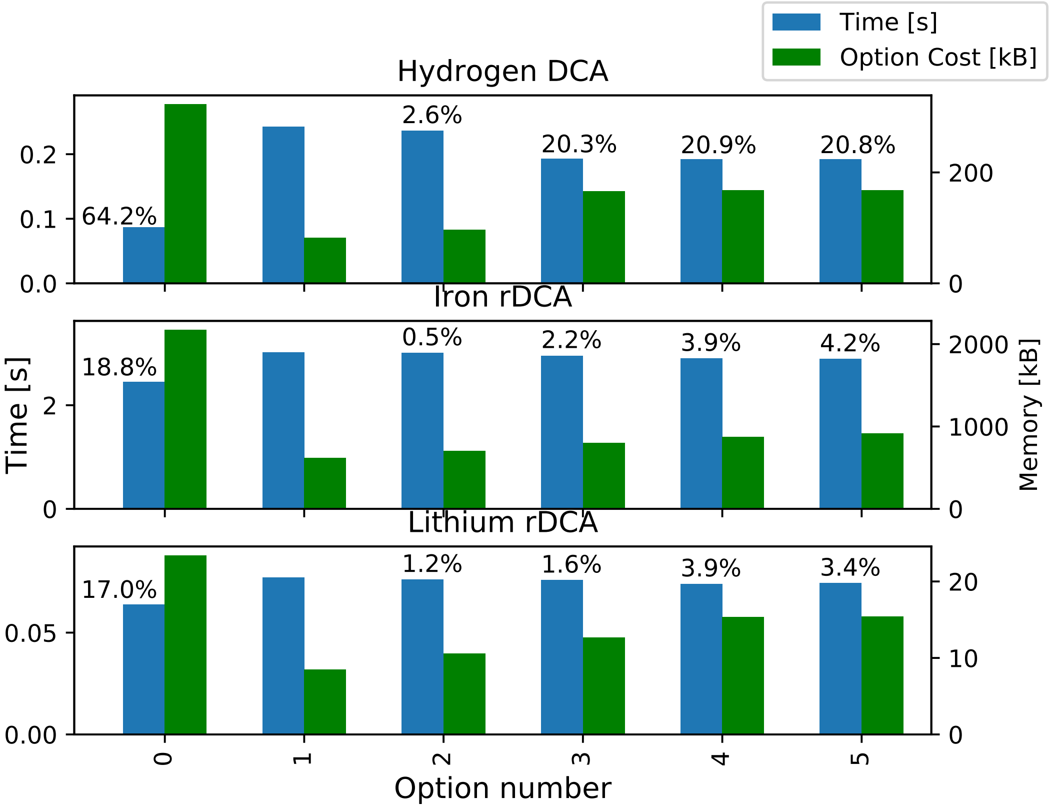


Figure 4.2: This plot compares memory options 0 through 5. The time is in blue (left bars) while memory is in green (right bars). This shows the time-space tradeoffs relative to option 1. The percentages listed are the % timing improvement over option 1.

While option 0 might not present the optimal choice in a combined context, it does always provide the best performance. The goal of these options was to provide flexibility to users with the hope that the most performant option that can be afforded with respect to memory will be chosen. In the runs presented in this work, option 0 was able to be used without running into memory issues.

4.2 Parallel Scalability

The use of both intranode and internode load balancing, as currently implemented in `inlinlte`, requires communication between MPI ranks. For load balancing to prove useful for large scale simulations, it must demonstrate excellent parallel scalability. The algorithm used was described in the previous chapter and is in theory scalable. A test was devised to simulate load imbalance in which each MPI rank started with an amount of NLTE work of $2 \times \text{rank} + 10$. This yields a total amount of work which is quadratic with respect to the number of ranks, and linear with respect to the amount of work per rank, on average. This test was run on the Grizzly platform at LANL using 4608, 9216, 18000, 18432, 36864, and 50400 cores. Note that only the 18000 and 50400 core cases are non-ideal in the sense that they require global communication and potentially multiple iterations to achieve a desired level of load balance. All other cases run were ideal and did not require global communication and only require one iteration. The results of the overall runs are shown in figure 4.3 and show that communication time is negligible relative to the time spent performing the NLTE $\langle Z \rangle$ calculation.

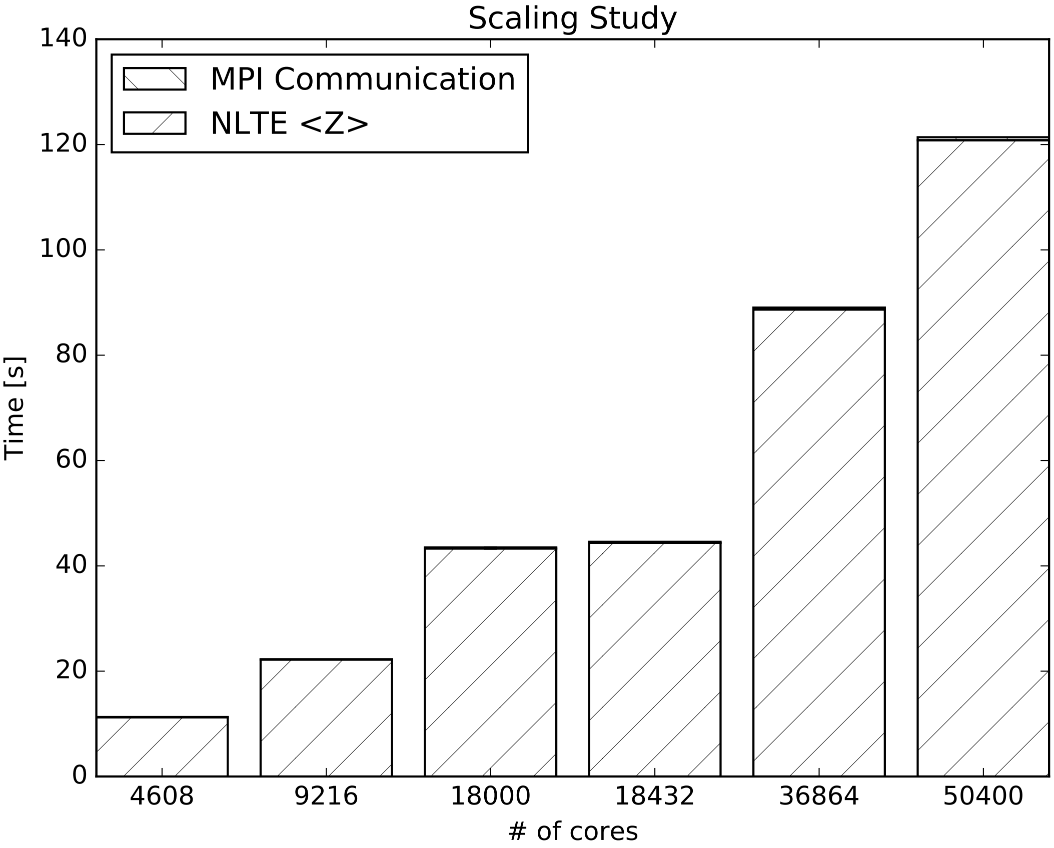


Figure 4.3: Comparing NLTE calculation time vs communication time. Error bars are plotted for each though they are very small on this scale. Communication time, while plotted, is also difficult to see on this scale. Note that the total execution time is roughly linear with respect to number of cores.

Communication times are further broken down into their four components in figure 4.4. While communication time grows with the number of cores, some increase is expected from algorithmic complexity as well as the fact that message sizes are also increasing with number of cores because the total amount of work is increasing quadratically.

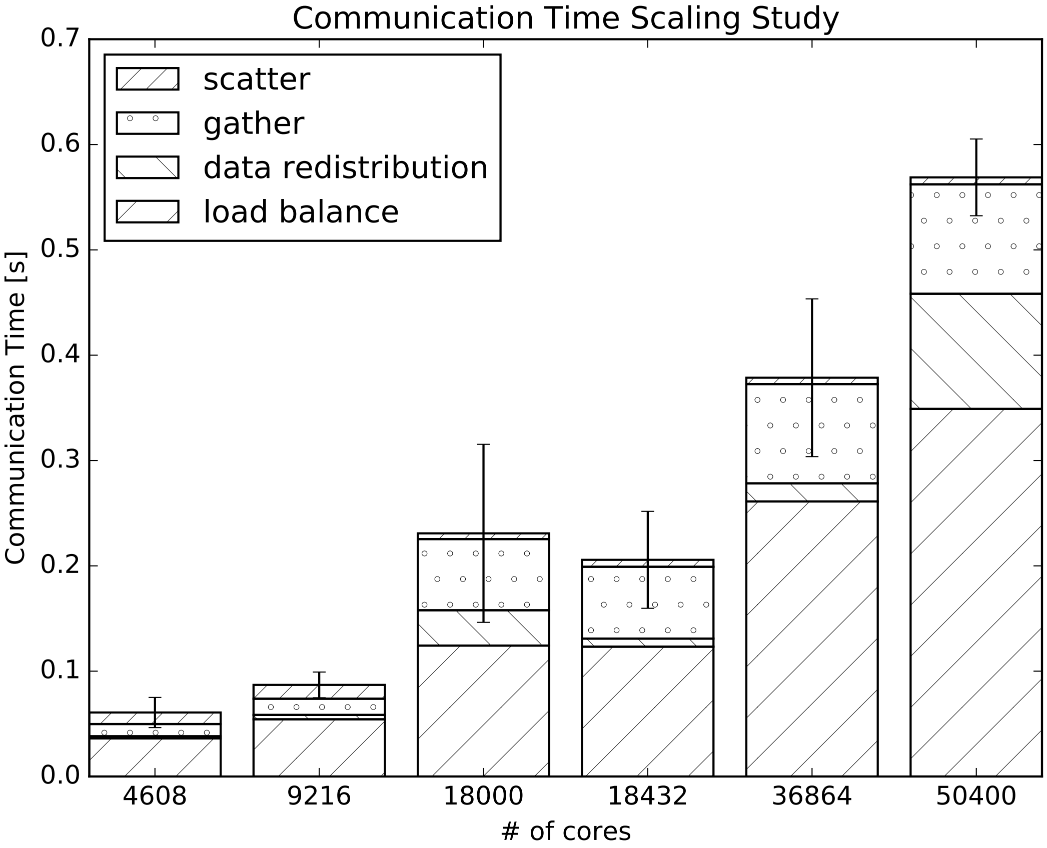


Figure 4.4: This plot breaks down communication time into its 4 components. The gather and scatter are intranode only, while the load balance and data redistribution are internode only. Error bars are shown for total communication time and not component-wise.

4.3 Sequential Speedup

In addition to load balancing in a multi-physics context, `inlinlte` provides notable performance improvements over `nlte_rdca` even when run sequentially, indicating that the NLTE solver is more efficient for a single set of inputs, not just in the aggregate. It is these gains in efficiency when combined with increased hardware utilization that result in order of magnitude or more speedup. Timing speedup is shown in figure 4.5.

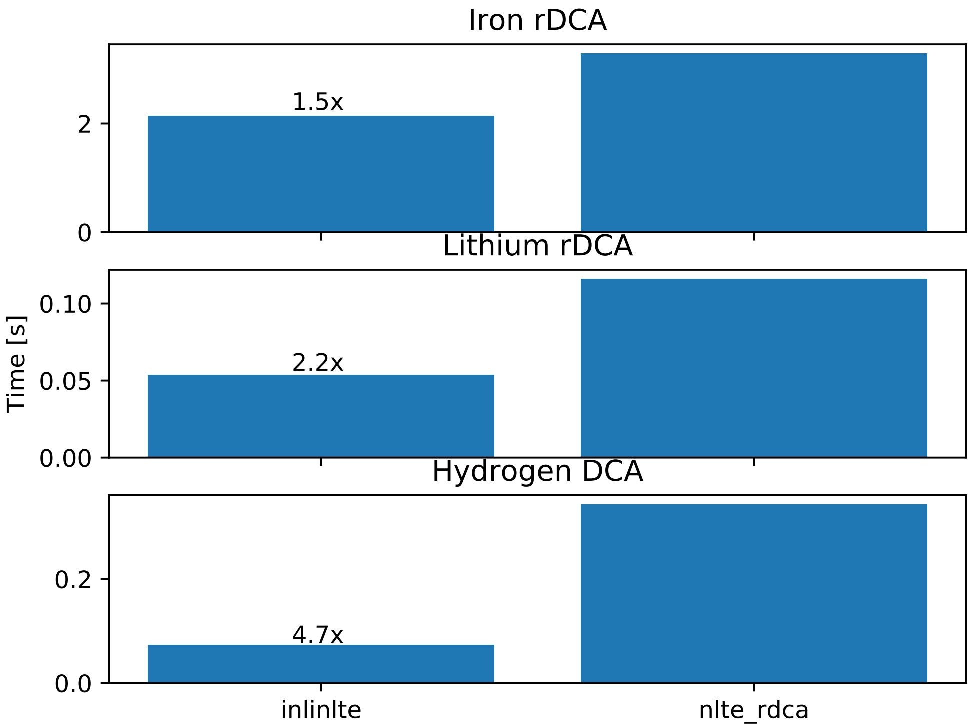


Figure 4.5: This shows performance improvements of `inlinlte` over the already implemented DCA solver when both are run without the use threading or MPI, though both will attempt to take advantage of vector processing.

4.4 Memory Consumption Improvements

Another significant improvement `inlinlte` brings is in memory consumption. Due to the nature of how `nlte_rdca` obtained parallelism, a copy of the atomic data is required per rank. `xRAGE` is designed to run well with 1 MPI rank per core, which is inefficient use of memory resources for cores that share the same memory space. Comparison is shown in table **??**.

Table 4.2: Note that despite noticeably poorer serial and parallel performance from nlte_rdca, it's memory requirements are similar to or worse than inlinlte's option 0 and universally significantly worse than inlinlte's option 1.

Model	nlte_rdca	option 0	option 1
DCA Hydrogen	16.14	11.63	3.18
rDCA Iron	69.71	77.51	22.88
rDCA Lithium	1.75	0.86	0.33

The Brent-Dekker method was used as the outer non-linear solver to converge the electron density. This method shows improvements over the Dekker method due to its use of inverse quadratic interpolation when appropriate [17, Chapter 9.3].

4.5 Speedup in Mulitphysics Simulations

These improvements manifest themselves in highly integrated radiation hydrodynamics simulations as well. Included are plots showing the speedup over the previous NLTE solver implementation in the xRAGE rad-hydro code. Figure 4.6 shows the speedup over the simulation run with the previous nlte_rdca NLTE solver. This is showing the speedup as a function of (normalized) simulation time. Speedup initially starts out low because xRAGE initialization dominates simulation time for the first cycle. However, the speedup rapidly increases as the amount of NLTE work peaks in the first fifth of the simulation. This peaking is due to the fact that NLTE work is localized to particular region of the problem domain and inlinlte is able to distribute that work effectively. After about a fifth of the way through the simulation, the material globally cools to below the NLTE cutoff temperature except in the region very near the shock, yielding almost no NLTE work for the remainder of the

Table 4.3: Walltime values for each simulation. NLTE physics about a factor of 30 slower than LTE and a factor of 3 faster than previous method.

	LTE	inlinlte	DCA solver
Wall time [hr]:	16.3	59.1	98.0
CPU time [hr]:	32.7	965.3	2943.5

simulation. Because of this, neither NLTE solver is being utilized appreciably and so the speedup factor begins to relax as they are taking roughly the same amount of time per cycle. Even so, the speedup factor appears to by relaxing to around 3×.

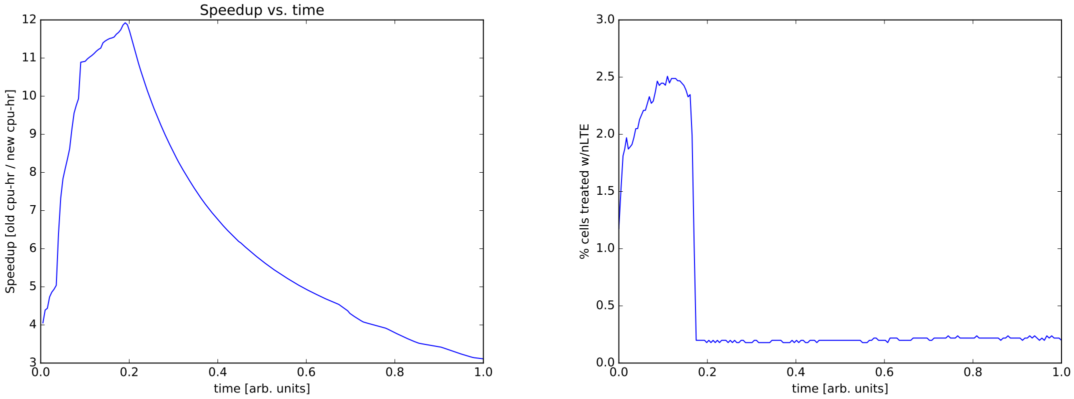


Figure 4.6: Left: Speedup vs time; Right: % cells treated with NLTE

Even for a relatively small problem, when NLTE work was present, more than an order of magnitude speedup was realized. Similar, but in many ways more promising, results were obtained for a problem involving iron, a material with a much higher proton number. Total walltime figures are presented in table 4.3 and show significant overall speedup. The time scales of these speedups should also be noted. Figure 4.7 shows similar characteristics to the supernova-like problem in that the speedup starts out small due to initialization, but rapidly jumps to a value of over 18×, well over an order of magnitude improvement. This number starts to decrease, but for a different reason than in the previous problem. Notice that the amount of NLTE work continues to rise, and more specifically, a larger portion of the simulated domain requires NLTE physics. This is reducing the degree to which the NLTE work is imbalanced because in the limit that 100% of the domain is above the NLTE cutoff temperature, the NLTE work is balanced to the same degree to which the domain has been evenly decomposed across resources. Thus in this limit, benefits obtained due to better load balance are reduced and the speedup factor continues to relax to a lower value. However, this comparison was not carried out to later solution times due to the excessive computational cost of `nlte_rdca`, which would have required well over a month of wall time to complete.

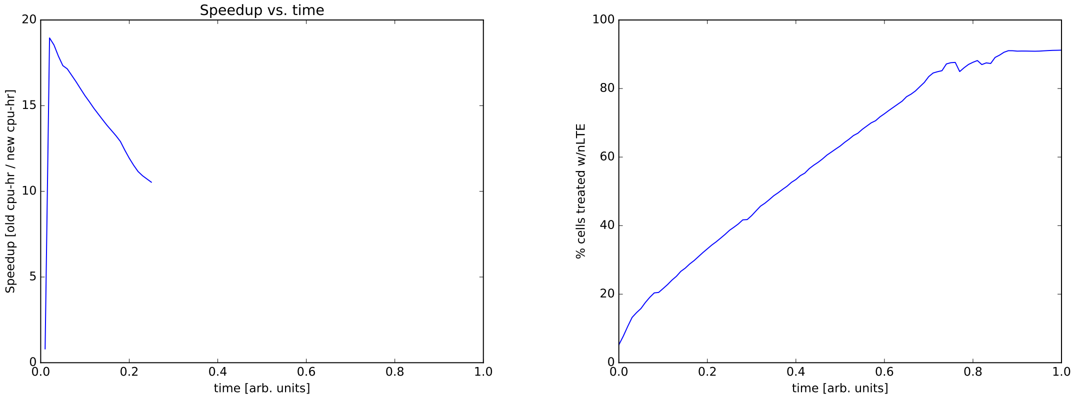


Figure 4.7: Left: Speedup vs time; Right: % cells treated with NLTE

Total walltime figures are presented in table 4.4 and show significant overall speedup. The time scales of these speedups should also be noted. While it might be possible and beneficial to complete a handful of these 1D simulations with

Table 4.4: Walltime values for each simulation. Values in parentheses are for the entire simulation to finish, those not in parentheses are for approximately the first 25% of the simulated time.

	LTE	inlinlte	DCA solver
Wall time [hr]:	0.5 (2.1)	32.7 (214.2)	230.8 (*)
CPU time [hr]:	7.8 (33.4)	673.7 (6482.3)	7096.4 (*)

nlte_rdca, its applicability does not extend much beyond that. Due to the speed improvements, the resources necessary to run similar problems is low enough that a parameter studies on 1D problems are now possible, which will be demonstrated in a later chapter. Additionally, a 2D problem was also run with the same geometry as the blastwave problem, but with slightly modified materials [14]. It appears that problems like this have been made possible thanks to the performance improvements brought about by inlinlte. Detailed performance comparisons of this 2D simulation with the previous capability were not possible due to timing constraints. Looked at another way, this simulation would not have been run without this capability in place. Instead, the simulation was restarted from a particular point in time in which NLTE physics is relevant but it was run with the previous NLTE solver. The time in particular was the time from -4 ns to -3.6 ns of the simulation time. This restart simulation ran for 400 picoseconds of simulation time and the wall time required was recorded and compared to the wall time needed for the simulation that used inlinlte for the NLTE solver. The results are shown in table 4.5.The data above show that the incorporation of NLTE physics is very computationally expensive relative to LTE by nearly 2 orders of magnitude for this problem in particular. However, that performance figure is through the use of inlinlte, the previous capability appeared to be roughly 3 orders of magnitude more expensive. In addition,

Table 4.5: Performance of `inlinlte` for the blastwave simulation. Parenthetical values represent wall time, all others are CPU time. Note that the LTE simulation ran with $1/16^{th}$ of the resources.

	LTE	`inlinlte`	DCA solver
400 ps Time [hr]:	24.7 (0.685)	1810.8 (3.145)	23543.4 (40.875)
Total time [hr]:	7503.6 (116.1)	317280 (552.6)	N/A

the wall-time figures were much closer due to the fact that the LTE simulation ran with 16× fewer computational resources. However, an RH simulation of this size would not be able to take advantage of those resources as well. The NLTE physics is embarrassingly parallel, allowing it to scale well as computational resources increase. These performance data indicate that `inlinlte` in tandem with HPC enable the study of systems of interest in which NLTE effects could be important.

5. ASTROPHYSICS: SUPERNOVA

A supernova (SN) is an explosive event that can occur in the late stages of stellar evolution. This can occur either when material accretes onto the surface of a compact star known as a white dwarf or in the core of a massive star ($m > 8M_{\odot}$). When this occurs, thermal pressure is no longer able to support the star in its current state. In the core of a massive star, when iron starts to undergo fusion, these endothermic reactions rob the core of thermal energy, thus reducing thermal pressure. This sudden deletion of pressure in the core causes the central region to implode and then explode. Given the proper conditions in the implosion and explosion, the resulting outward driven shock can gravitationally unbind a significant fraction of the star's mass and eject into the interstellar medium at extremely high velocities. These events occur at extreme scales and in extreme conditions and result in impressively complex dynamics. The primary source of information about the inner workings of supernovae is obtained from the electromagnetic radiation produced from the time the shock breaks out of the surface of the star until weeks, months, or even centuries later. These electromagnetic signals are complemented with data from cosmic rays, neutrinos, and direct observations of isotopic abundances. This data carries information about events with extreme conditions and scales that cannot be reached in a laboratory. This data can be compared against simulations to provide validation of codes used to simulate such events. Validation or invalidation of models can aid in gauging our understanding of physics. With thousands of SNe observed each year [15] this radiation provides an abundant source of information about these events that probe extreme conditions.

A light curve depicts the luminosity of an event as a function of time for time scales on the order of 100s of days. Over this time scale, the material ejected from the explosion has typically cooled to sub-eV temperatures and become rarefied. The assumption of LTE tends to over-predict the ionization of the hydrogen in this ejected material [18, Figure 2.3]. The opacity of fully stripped ion stage of hydrogen does not have any line structure and would preclude characteristic x-rays such as the Lyman alpha (Ly-α) series from being observed in a simulation that assumed LTE. This is inaccurate and when NLTE physics are accurately captured, this spectroscopic behavior is recovered [22]. This is not currently a capability available to the astrophysicists at LANL using `xRAGE` [9] in conjunction with `Rainbow-` to calculate SN [8] and `inlinlte` can provide this to both RH and post-processing codes. Here I demonstrated how these new tools can used to incorporate NLTE physics into modeling astrophysical phenomena in the LANL suite of light curve and spectra codes.

5.1 Supernovae-Like Problem

Due to current single material limitation of `inlinlte`'s NLTE post-processing capabilities, a SN simulation was carried out in which hydrogen was the only material in the simulation. This is because the `inlinlte` is unable to handle atomically mixed material opacities. Due to this limitation, this is not truly meant to simulate a supernova in full detail for direct comparison to observations. Instead, it is specifically designed to show the increased fidelity obtained when treating hydrogen with NLTE in a post-processed fashion, obtaining spectral features not currently obtained under LTE. While the approximation of a single material is used for demonstrative purposes, for certain classes of SNe this approximation is not as extreme as it might appear. While current stars are composed of approximately 2% elements heavier than helium and these elements can dominate the opacity, the earliest stars in the universe are composed entirely of H and He. Lightcurves from these earlier stars have been a major focus of the LANL collaboration [23]. The initial condition of the problem is shown in figure 5.1.

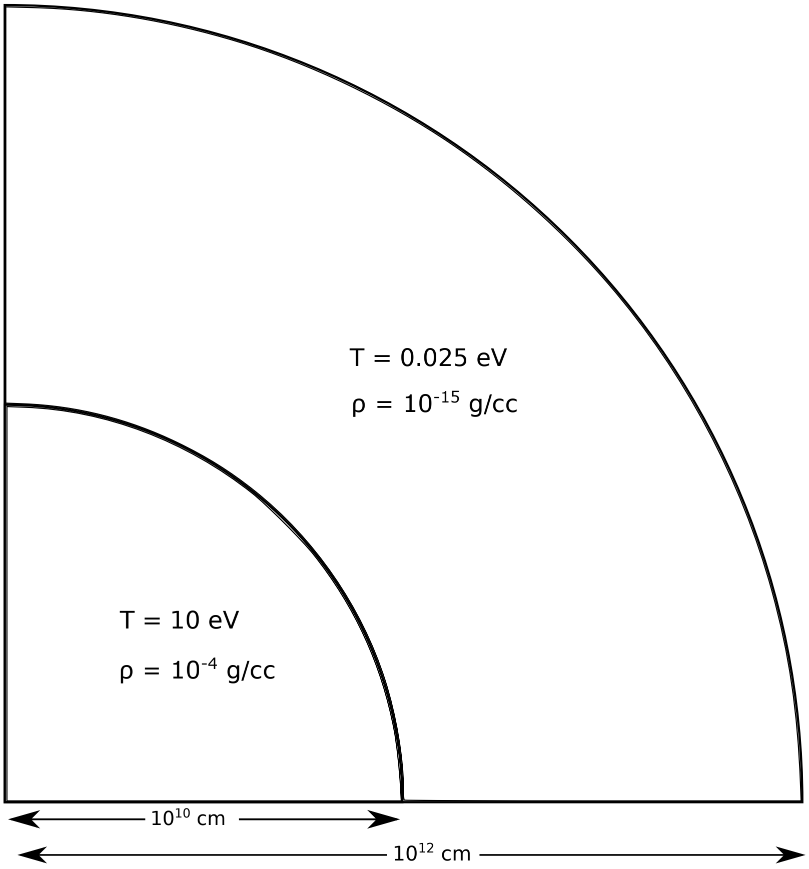


Figure 5.1: Supernova-like initial condition.

This system was simulated using the `xRAGE` code developed at LANL using a similar configuration as the one described in [8]. This simulation used an analytic ideal gas equation of state for the hydrogen and was run with and without NLTE opacity look ups. These simulations used flux limited multi-group diffusion with

49 groups as the radiation transport model. For performance characteristics of this simulation, see chapter 4. The outputs of these simulations can be post-processed to obtain a simulated electromagnetic signal of this event similar to what would be recorded by telescopes and other data collection instruments used by observational astronomers. The `scootaloo` post-processor is a tool developed at LANL for this purpose and is a stripped-down python implementation of the `mkbin` and `spectra` tools in the `Rainbow-` framework described in [8] and was modified to use `inlinlte`'s monochromatic opacity capabilities to obtain spectra. `scootaloo` was used to obtain spectra at three different times in the simulation. `inlinlte` was used to calculate opacities under both LTE and NLTE conditions. LTE opacities were obtained by assuming the radiation temperature was equal to the electron temperature and NLTE opacities were obtained by using the true radiation temperature for T_r. A spectrum was generated under each assumption for each time being considered. Additionally, the simulation being post-processed was run under LTE and NLTE assumptions in the RH code. The results for the first timing snapshot are shown in figures 5.2 and 5.3, which occurs at approximately 2 hours into the event.

The results from the NLTE simulation show that significant differences can occur between the LTE and NLTE post-processed opacities, resulting in noticeably different luminosities, particularly near the photon energies in which the luminosity peaks. The difference between the radiation and electron temperatures is large enough that the assuming both temperatures are equal results in very different opacities. Specifically, the NLTE opacity for the outer region shows little to no ionization whereas the LTE opacity indicates ionization. This observation is inferred from the fact that the NLTE opacity does not exhibit the E^{-3} dependence at low energies, which is exhibited when $\langle Z^2 \rangle > 0$.

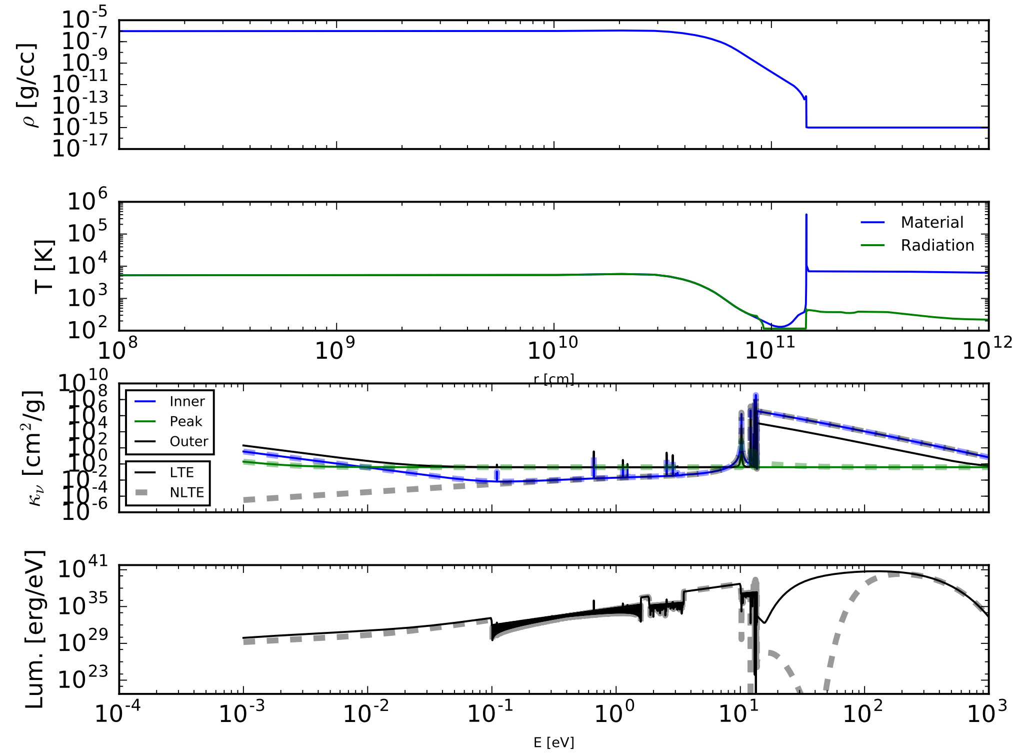


Figure 5.2: This figure is the result from post-processing the output of the NLTE RH simulation at 2 hours. Top: spatial density profile; Upper middle: spatial radiation and electron temperature profiles (note in K); Lower middle: monochromatic opacities at inner, outer, and highest temperature conditions; Bottom: spectral luminosity.

In addition to the differences between NLTE and LTE opacities, line features from the hydrogen Balmer series are observed in the spectral luminosity, including a feature from the notable $H - \alpha$ line. The SN-like simulation run with LTE physics was also post-processed at approximately 2 hours into the event yielding an analogous figure, figure 5.3. It is notable that the LTE simulation yields temperatures in the outer regions of the problem lower than in the NLTE simulation. These lower temperatures are not high enough to cause ionization for either the LTE or NLTE opacities. As a result, the NLTE and LTE opacities and spectral luminosities qualitatively agree. Although this problem is of an illustrative nature, it shows that NLTE effects were able to noticeably manifest themselves within the RH simulation in a noticeable way in the post-processing. It is important to note that NLTE physics plays a role both in the initial RH simulation and in the post-processor, the exact impacts of those effects were not completely known at the onset of this work.

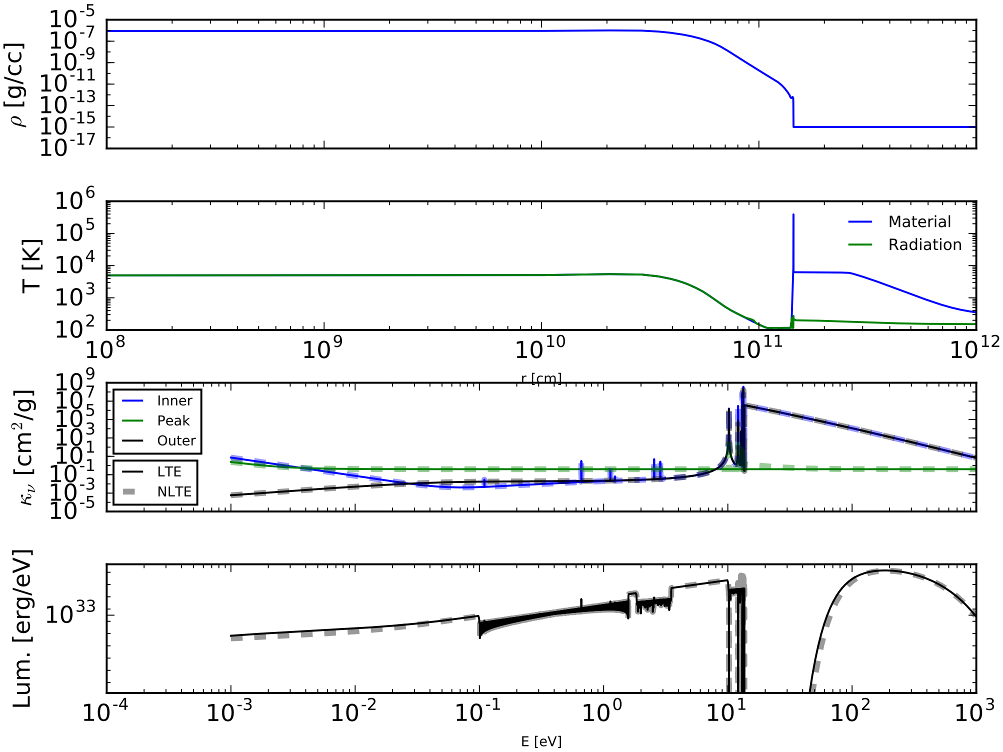


Figure 5.3: This figure is the result from post-processing the output of the LTE RH simulation at 2 hours and has the same layout as figure 5.2

While unexpected, this behavior reinforces the idea that NLTE physics play a role in the dynamics of SN ejecta. Line features are again recovered in the spectral luminosity including those from the Balmer series with qualitative agreement between NLTE and LTE. This procedure was repeated at 4 hours into the simulation and generated analogous figures. The RH simulation which used NLTE physics was post-processed, the results of which are shown in figure 5.4. The spectral luminosities that resulted from LTE and NLTE opacities again displayed qualitative differences, particularly in the photon energy ranges near where the spectral luminosity peaks. The opacity at the outermost region of the simulation shows ionization in the LTE case but shows little signs of ionization in the NLTE case. These differences result in noticeable differences in the bolometric luminosity, which is an important metric used in the characterization of astronomical events.

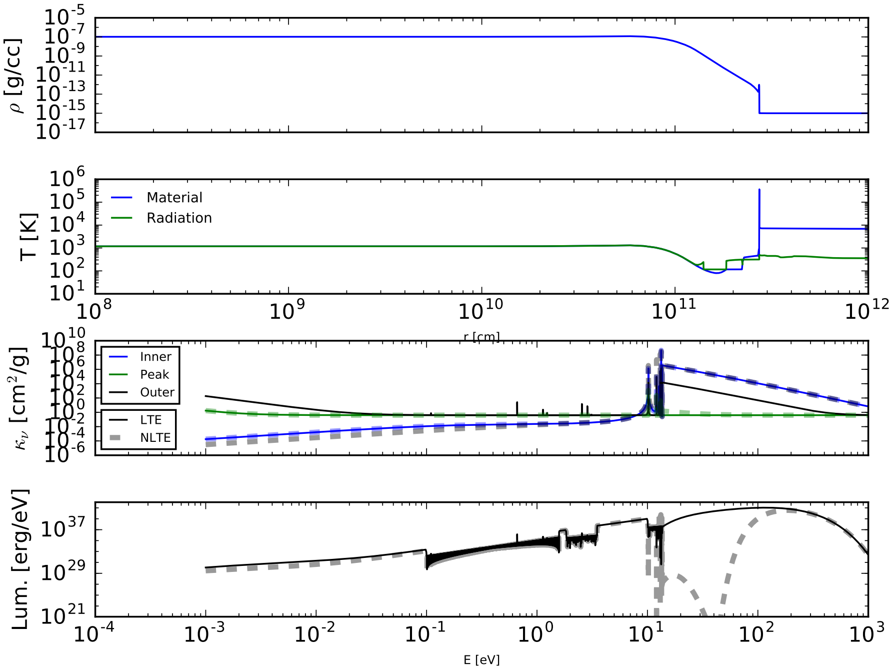


Figure 5.4: This figure is the result from post-processing the output of the NLTE RH simulation at 4 hours and has the same layout as figure 5.2

As was the case at $t = 2$hrs, line features from the Balmer series are recovered in both LTE and NLTE plots. The RH simulation under the LTE assumption was post-processed at around 4 hours into the event, the results of which are shown in figure 5.5. The spectral luminosities when calculated with LTE and NLTE opacities are qualitatively similar and both recover line features that are expected in astronomical events similar to the one simulated.

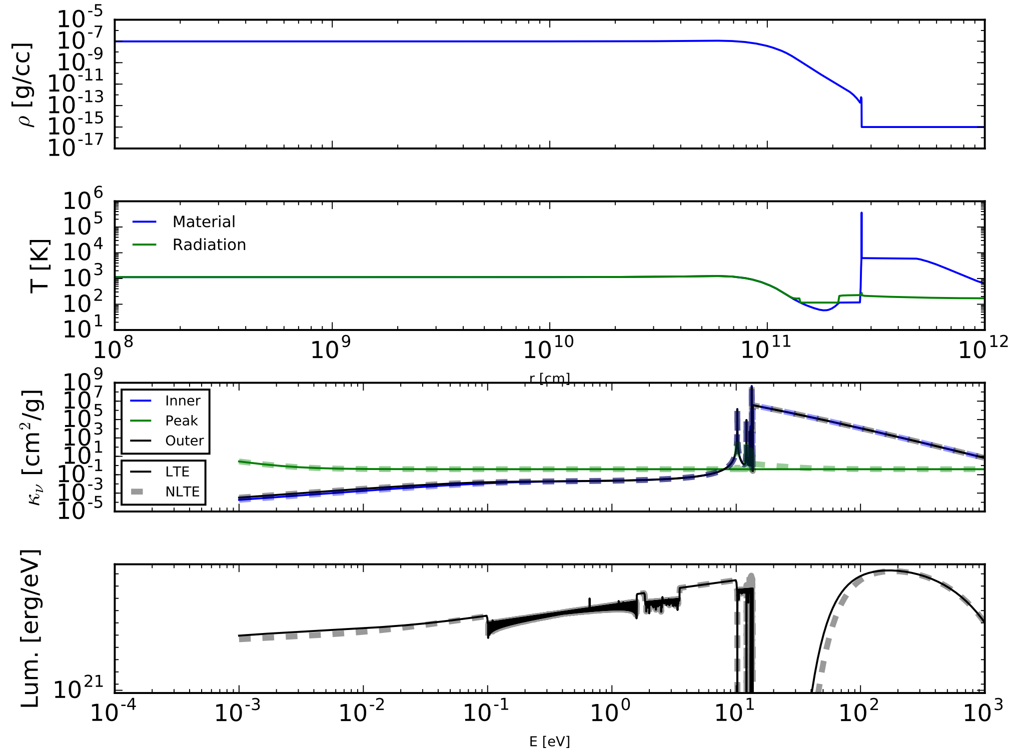


Figure 5.5: This figure is the result from post-processing the output of the LTE RH simulation at 4 hours and has the same layout as figure 5.2

The LTE and NLTE opacities in the 3 conditions presented in figure 5.5 do not show strong differences and helps explain the qualitative similarities observed in the spectral luminosities. This procedure was repeated at 8 hours into the simulation and generated analogous figures. The RH simulation which used NLTE physics was post-processed, the results of which are shown in figure 5.6. The spectral luminosities that resulted from LTE and NLTE opacities again displayed qualitative differences supported by the fact that the opacity at the outermost regions of the simulation shows ionization in the LTE case but shows little signs of ionization in the NLTE case.

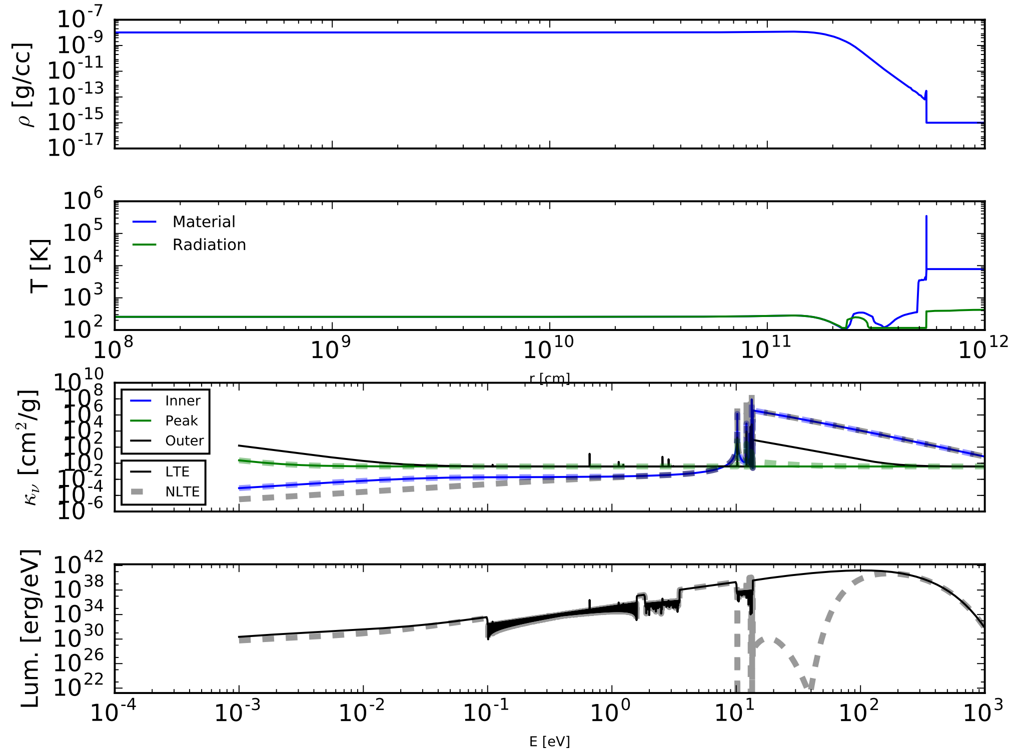


Figure 5.6: This figure is the result from post-processing the output of the NLTE RH simulation at 8 hours and has the same layout as figure 5.2

As was the case at t = 4hrs, line features from the Balmer series are recovered in both LTE and NLTE plots. The RH simulation under the LTE assumption was post-processed at around 8 hours into the event, the results of which are shown in figure 5.7. The spectral luminosities when calculated with LTE and NLTE opacities seem to show some small but noticeable differences, especially for photon energies near where the spectral luminosity peaks. These differences are supported by the fact that NLTE and LTE opacities in the outer regions of the problem differ with regard to the degree of ionization present. The opacity calculated under LTE conditions shows minor ionization while the opacity calculated with NLTE conditions does not show signs of appreciable ionization.

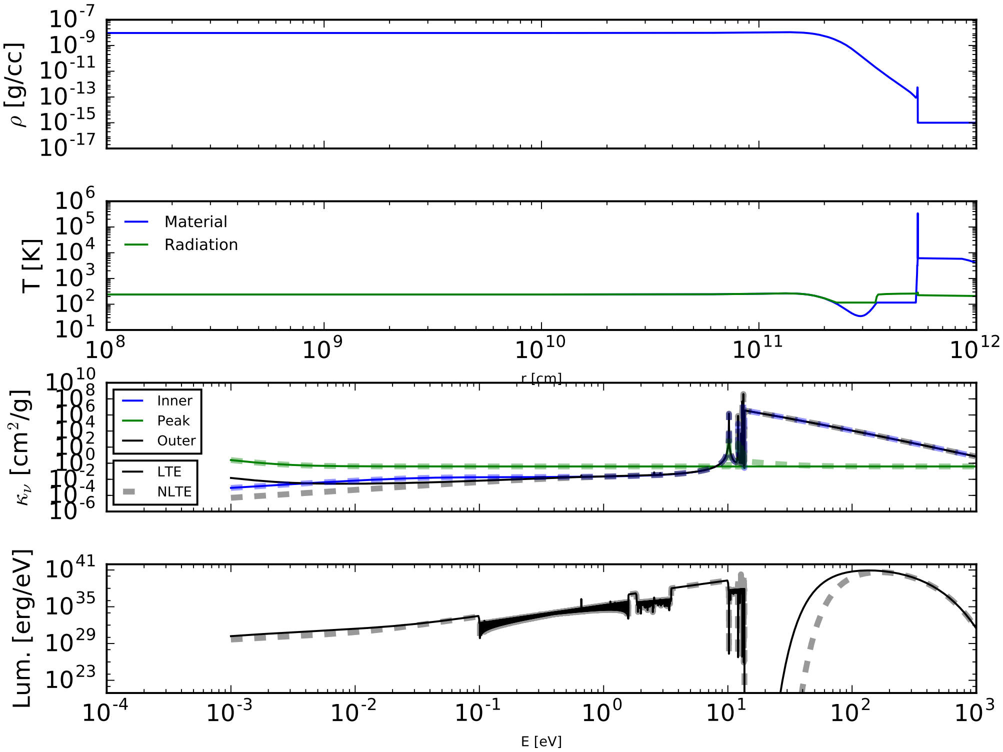


Figure 5.7: This figure is the result from post-processing the output of the LTE RH simulation at 8 hours and has the same layout as figure 5.2

An interesting result of this work is the fact that it can be difficult to predict when NLTE physics will have a large effect or not. This uncertainty is largely due to the presence of physics cliffs, namely excitation and ionization in this case. The opacity of a neutral plasma with most atoms residing in the ground state will look very different from that of a nearly fully ionized plasma. In addition, if the plasma is cold enough to have very few atoms in an excited energy level, this will result in features associated with the Balmer series being largely absent. Conversely, if the plasma is hot enough to be fully ionized, this will result in the absence of any line structure in the opacity. One should bare in mind that caveats exist with regard to the actual results obtained due to physics not currently taken into account in `inlinlte` but that could be relevant in this low temperature, low density regime. That being said, the conclusions drawn with regard to physics cliffs should still hold true.

6. HEDP: INERTIAL CONFINEMENT FUSION

Inertial confinement fusion (ICF) occurs when a system undergoes fusion reactions in a fuel region in which the inertia of the plasma outside of this region is sufficient to provide the containment and compression of the fuel region. Inertial confinement is a method of plasma confinement in which the plasma inertia provides sufficient containment. On systems of interest, inertial confinement timescales are very short when compared to magnetic confinement. A system is said to obtain ignition when it is able to sustain conditions favorable for fusion without the need for external power input. This is the namesake for the National Ignition Facility (NIF), a laser system located at Lawrence Livermore National Laboratory (LLNL). This system houses the world's most energetic laser and is used to conduct ICF research. The targets meant to undergo ignition use the laser as a heating mechanism. The UV laser impinges upon the walls of a cylinder made of a high-Z metal called the hohlraum. The outer surface of the hohlraum walls rapidly heat up, creating a hot plasma. This plasma emits thermal radiation in an attempt to symmetrically compress a plastic capsule located at the center of the hohlraum. This symmetric implosion creates a dense hot spot at the center of the capsule capable of fusing the D-T mixture. The mixture fuses, creating energy in the form of neutrons and alpha particles. Most of the energy is contained within the neutrons, the majority of which stream out of the system. However, some of the reaction energy is contained in the resulting alpha particle. The alpha particles likely deposit their energy in their surroundings before escaping. This energy deposition heats up the surrounding material, hopefully making it able to support fusion reactions. If the alpha particles created in the initial hot spot succeed in sufficiently heating the surrounding material to fusion conditions, a burn wave will move radially outward burning much of the fuel and producing a substantial amount of energy.

6.1 Hohlraums

Accurately modeling capsule implosions requires accurate spatial, spectral, and temporal characterization of xrays emitted by the hohlraum. While this can be measured, it is difficult to predict from first principal simulations. This could be in part due to the fact that the hohlraum itself is driven by an inherently non-Planckian laser source and in some experiments is observed to be out of LTE [19]. Bringing simulations in closer agreement with experiment will give design scientists more confidence in the predictive capabilities of simulations. This would make hohlraum simulations a much more powerful tool for scientists working on ICF.

To explore this problem space, and in particular the effects of NLTE physics, a simplified problem was devised that could be simulated with significantly decreased computational costs, but still capture the physical processes important in hohlraums. This problem is composed of two slabs of iron each 100 μm thick separated by hydrogen gas at standard temperature and pressure (STP) and a density of $10^{-6}\ \frac{\mathrm{g}}{\mathrm{cm}^3}$. The right slab of iron was also initially at STP. The left half of the left slab was at STP, while the right half of the left slab starts out at 300 eV and 1 atmosphere. This is shown in figure 6.1.

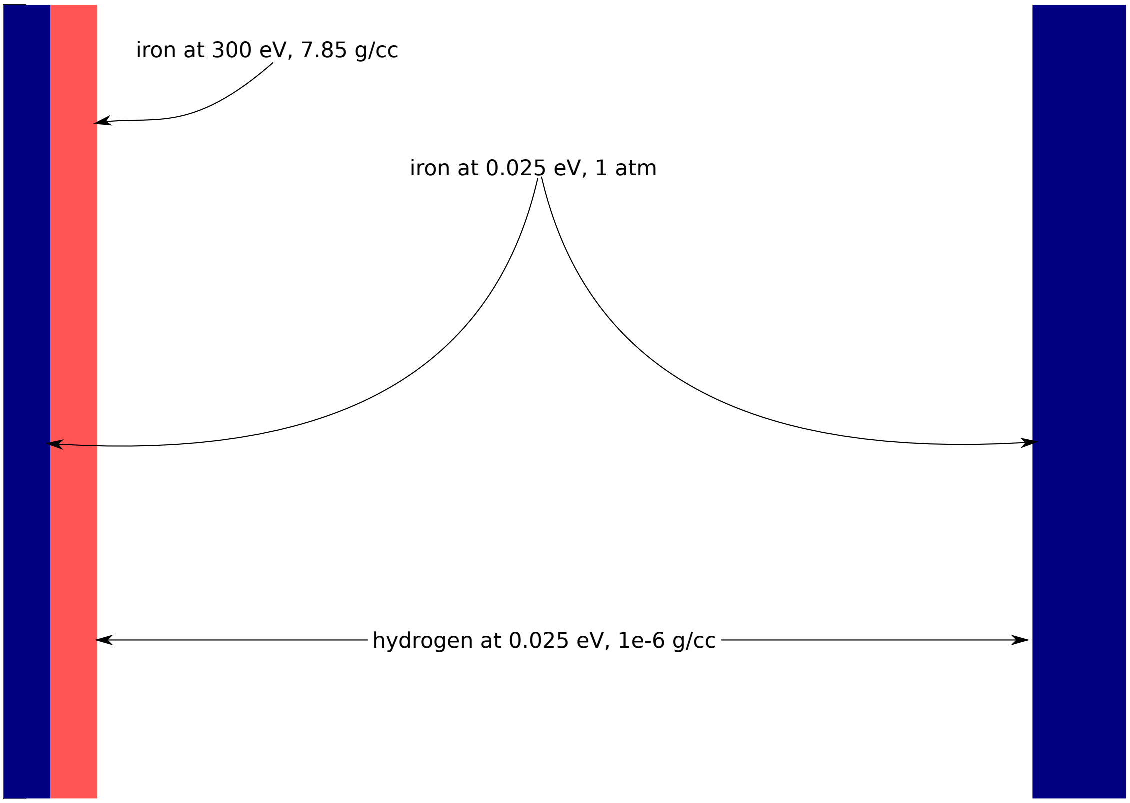


Figure 6.1: Initial Condition for exploratory problem

This problem was chosen because it contains similar physics to a hohlraum, but is not computationally expensive. This problem was run using `xRAGE` [9] using its builtin flux limited multigroup radiation diffusion solver. A large set of simulations were conducted in which the plate separation distance (1, 2, and 4 mm), number of energy groups (5, 50, and 100), temperature of the hot iron (300 eV, 750 eV, and 1000 eV), and the pressure of the iron plate on the right hand side of the problem (0.1 and 1 atm) were varied. The primary QOI is the time at which the ablation fronts collide. These simulations were run under the LTE assumption as well as NLTE with two different atomic models with varying complexity. This complexity is varied through a parameter associated with averaging of dielectronic capture states [1, §2]. The larger model uses $m = 30$ and the smaller model uses $m = 10$, where

m is explicitly defined in [1]. That data is listed for both NLTE iron atomic models as well as LTE, in table 6.1. This complete listing of the data contains the expected trend that the most significant differences between LTE and NLTE occur at higher temperatures. The largest differences in this table appear to be about 6%.

Table 6.1: Full comparison varying all possible parameters holding the cutoff temperature constant.

Gap	G	T_{hot}	P_{cold}	NLTE-L	NLTE-S	LTE	$2\frac{\lvert L-S\rvert}{\lvert L+S\rvert}$	$2\frac{\lvert L-LTE\rvert}{\lvert L+LTE\rvert}$	$2\frac{\lvert S-LTE\rvert}{\lvert S+LTE\rvert}$
[mm]	[#]	[keV]	[atm]						
4	50	0.75	0.1	1.33e-09	1.33e-09	1.36e-09	9.23e-06	2.23e-02	2.23e-02
2	50	1	0.1	5.80e-10	5.80e-10	6.00e-10	3.48e-06	3.39e-02	3.39e-02
4	50	0.3	0.1	2.65e-09	2.65e-09	2.64e-09	2.54e-06	3.79e-03	3.79e-03
1	50	0.75	1	3.30e-10	3.30e-10	3.40e-10	2.98e-06	2.98e-02	2.98e-02
2	100	0.75	1	6.60e-10	6.60e-10	6.80e-10	2.11e-06	2.98e-02	2.99e-02
1	5	0.3	0.1	6.80e-10	6.80e-10	6.80e-10	4.51e-05	3.32e-05	1.19e-05
4	50	1	1	1.18e-09	1.17e-09	1.21e-09	8.51e-03	2.51e-02	3.36e-02
1	50	0.75	0.1	3.40e-10	3.30e-10	3.40e-10	2.98e-02	2.70e-05	2.99e-02
4	100	0.75	0.1	1.33e-09	1.33e-09	1.35e-09	1.47e-06	1.49e-02	1.49e-02
4	50	0.3	1	2.65e-09	2.65e-09	2.64e-09	8.21e-06	3.77e-03	3.78e-03
1	50	0.3	0.1	6.40e-10	6.40e-10	6.40e-10	1.15e-05	2.12e-05	9.72e-06
2	50	0.3	1	1.29e-09	1.29e-09	1.29e-09	1.72e-06	9.80e-06	1.15e-05
4	5	1	1	1.16e-09	1.16e-09	1.20e-09	3.97e-06	3.39e-02	3.39e-02
4	100	1	1	1.17e-09	1.17e-09	1.21e-09	1.12e-05	3.36e-02	3.36e-02
2	50	0.3	0.1	1.29e-09	1.29e-09	1.29e-09	4.18e-06	1.95e-05	2.37e-05
4	50	0.75	1	1.33e-09	1.33e-09	1.35e-09	8.25e-06	1.49e-02	1.49e-02
2	100	1	0.1	5.80e-10	5.80e-10	6.00e-10	5.57e-07	3.39e-02	3.39e-02
2	5	0.3	1	1.35e-09	1.36e-09	1.35e-09	7.39e-03	1.50e-05	7.37e-03
1	5	1	1	2.90e-10	2.90e-10	3.10e-10	4.23e-05	6.66e-02	6.67e-02

Table 6.1: Continued

Gap	G	T_{hot}	P_{cold}	NLTE-L	NLTE-S	LTE	$2\frac{\lvert L-S\rvert}{\lvert L+S\rvert}$	$2\frac{\lvert L-LTE\rvert}{\lvert L+LTE\rvert}$	$2\frac{\lvert S-LTE\rvert}{\lvert S+LTE\rvert}$
[mm]	[#]	[keV]	[atm]						
2	100	0.3	0.1	1.29e-09	1.29e-09	1.29e-09	3.50e-06	1.14e-05	7.87e-06
4	5	0.3	0.1	2.75e-09	2.75e-09	2.75e-09	3.45e-06	2.94e-06	5.13e-07
4	50	1	0.1	1.17e-09	1.18e-09	1.21e-09	8.51e-03	3.36e-02	2.51e-02
4	100	0.75	1	1.33e-09	1.33e-09	1.35e-09	8.79e-07	1.49e-02	1.49e-02
1	100	1	1	3.00e-10	2.90e-10	3.10e-10	3.39e-02	3.28e-02	6.66e-02
4	100	1	0.1	1.17e-09	1.17e-09	1.21e-09	5.50e-06	3.36e-02	3.36e-02
1	100	0.3	0.1	6.40e-10	6.40e-10	6.40e-10	2.59e-05	1.06e-05	1.53e-05
2	50	0.75	1	6.60e-10	6.60e-10	6.80e-10	7.23e-06	2.99e-02	2.99e-02
1	100	0.75	1	3.30e-10	3.30e-10	3.50e-10	5.16e-06	5.89e-02	5.89e-02
1	50	1	1	3.00e-10	2.90e-10	3.10e-10	3.39e-02	3.28e-02	6.67e-02
2	5	1	1	5.70e-10	5.80e-10	6.00e-10	1.74e-02	5.13e-02	3.39e-02
2	5	0.75	1	6.50e-10	6.50e-10	6.70e-10	2.35e-05	3.03e-02	3.03e-02
2	50	1	1	5.80e-10	5.80e-10	6.00e-10	7.24e-06	3.39e-02	3.39e-02
1	100	1	0.1	2.90e-10	2.90e-10	3.10e-10	2.57e-05	6.67e-02	6.67e-02
4	5	1	0.1	1.16e-09	1.16e-09	1.20e-09	6.28e-06	3.39e-02	3.39e-02
1	100	0.75	0.1	3.30e-10	3.30e-10	3.50e-10	4.91e-06	5.88e-02	5.88e-02
1	50	1	0.1	2.90e-10	3.00e-10	3.10e-10	3.39e-02	6.67e-02	3.28e-02
2	5	1	0.1	5.70e-10	5.80e-10	6.10e-10	1.74e-02	6.78e-02	5.04e-02
4	5	0.3	1	2.75e-09	2.75e-09	2.74e-09	3.24e-06	3.64e-03	3.64e-03
1	50	0.3	1	6.40e-10	6.40e-10	6.40e-10	2.13e-05	1.80e-05	3.24e-06
1	5	0.3	1	6.80e-10	6.80e-10	6.80e-10	2.89e-05	2.89e-05	1.33e-08
1	5	0.75	1	3.30e-10	3.30e-10	3.50e-10	5.44e-05	5.89e-02	5.88e-02
2	100	0.3	1	1.29e-09	1.29e-09	1.29e-09	1.40e-05	2.39e-05	9.85e-06

Table 6.1: Continued

Gap	G	T_{hot}	P_{cold}	NLTE-L	NLTE-S	LTE	$2\frac{\|L-S\|}{\|L+S\|}$	$2\frac{\|L-LTE\|}{\|L+LTE\|}$	$2\frac{\|S-LTE\|}{\|S+LTE\|}$
[mm]	[#]	[keV]	[atm]						
4	5	0.75	0.1	1.31e-09	1.31e-09	1.34e-09	6.56e-06	2.27e-02	2.26e-02
1	100	0.3	1	6.40e-10	6.40e-10	6.40e-10	6.91e-06	1.16e-06	8.07e-06
1	5	0.75	0.1	3.30e-10	3.30e-10	3.50e-10	6.69e-06	5.89e-02	5.88e-02
2	5	0.75	0.1	6.50e-10	6.50e-10	6.70e-10	1.40e-05	3.03e-02	3.03e-02
2	50	0.75	0.1	6.60e-10	6.60e-10	6.80e-10	1.38e-05	2.99e-02	2.99e-02
4	5	0.75	1	1.31e-09	1.31e-09	1.34e-09	3.26e-06	2.26e-02	2.26e-02
2	100	1	1	5.80e-10	5.80e-10	6.00e-10	5.18e-06	3.39e-02	3.39e-02
1	5	1	0.1	2.90e-10	2.90e-10	3.10e-10	3.39e-05	6.67e-02	6.66e-02
4	100	0.3	0.1	2.64e-09	2.64e-09	2.64e-09	9.25e-06	8.35e-07	8.42e-06
4	100	0.3	1	2.64e-09	2.64e-09	2.64e-09	5.94e-06	1.22e-05	6.27e-06
2	100	0.75	0.1	6.60e-10	6.60e-10	6.80e-10	2.18e-06	2.98e-02	2.99e-02
2	5	0.3	0.1	1.36e-09	1.35e-09	1.35e-09	7.39e-03	7.39e-03	7.79e-06

A subset of the parameter configurations reported in table 6.1 were run with two additional atomic models, one with $m = 1$ and another with $m = 70$, yielding a total of 4 different levels of complexity in the atomic models. In addition, the configurations were run assuming LTE with S_4 transport using the LANL Eulerian Applications Project's `Cassio` code. Results for this subset are reported in table 6.2 below. From this data it can be seen that for this test problem the impact the complexity of the atomic model is less than a 3% effect. The maximum impact of LTE Sn transport vs. the NLTE solutions is at maximum 7%.

Table 6.2: Results for how atomic model complexity affects highly integrated QOIs.

Tcut	Gas Gap	Hot Temp.	Transport	XS	S	L	XL
500eV	1mm	750eV	3.50e-10	3.40e-10	3.50e-10	3.50e-10	3.40e-10
250eV	1mm	750eV	3.50e-10	3.40e-10	3.30e-10	3.30e-10	3.40e-10
250eV	2mm	750eV	6.80e-10	6.60e-10	6.60e-10	6.60e-10	6.60e-10
500eV	2mm	750eV	6.80e-10	6.70e-10	6.80e-10	6.80e-10	6.70e-10
500eV	1mm	1keV	3.10e-10	3.10e-10	3.10e-10	3.10e-10	3.10e-10
250eV	2mm	1keV	6.10e-10	5.80e-10	5.80e-10	5.80e-10	5.80e-10
500eV	2mm	1keV	6.10e-10	6.00e-10	6.00e-10	6.00e-10	6.00e-10
250eV	1mm	1keV	3.10e-10	2.90e-10	2.90e-10	2.90e-10	2.90e-10

A more detailed study was conducted at the spatial center of the simulation for each simulation is the parameter subset. At this singular location, the density, temperature, and radiation energy density was plotted as a function of time for LTE transport, LTE diffusion, and each of the 4 atomic models. In addition, the spectral energy density was plotted at 4 different times for each of the 6 models.

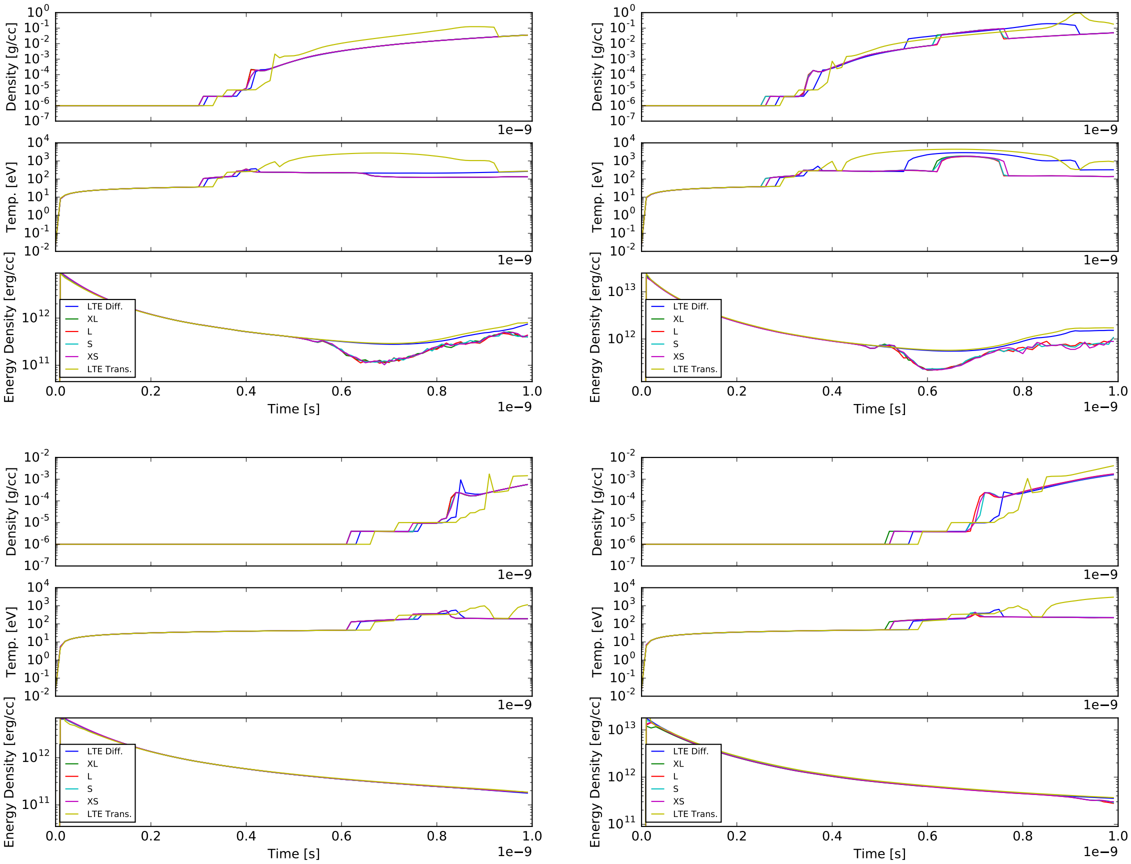


Figure 6.2: The array of figures has 1mm gas spacing simulations on the top row, 2mm simulations on the bottom row. The left column has a hot temperature of 750 eV and the right column has a hot temperature of 1 keV. All NLTE simulations assume a cutoff temperature of 250 eV.

Figure 6.2 includes results with an LTE cutoff temperature of 250 eV, while figure 6.3 shows the same results only the RH code was run using an LTE cutoff temperature of 500 eV.

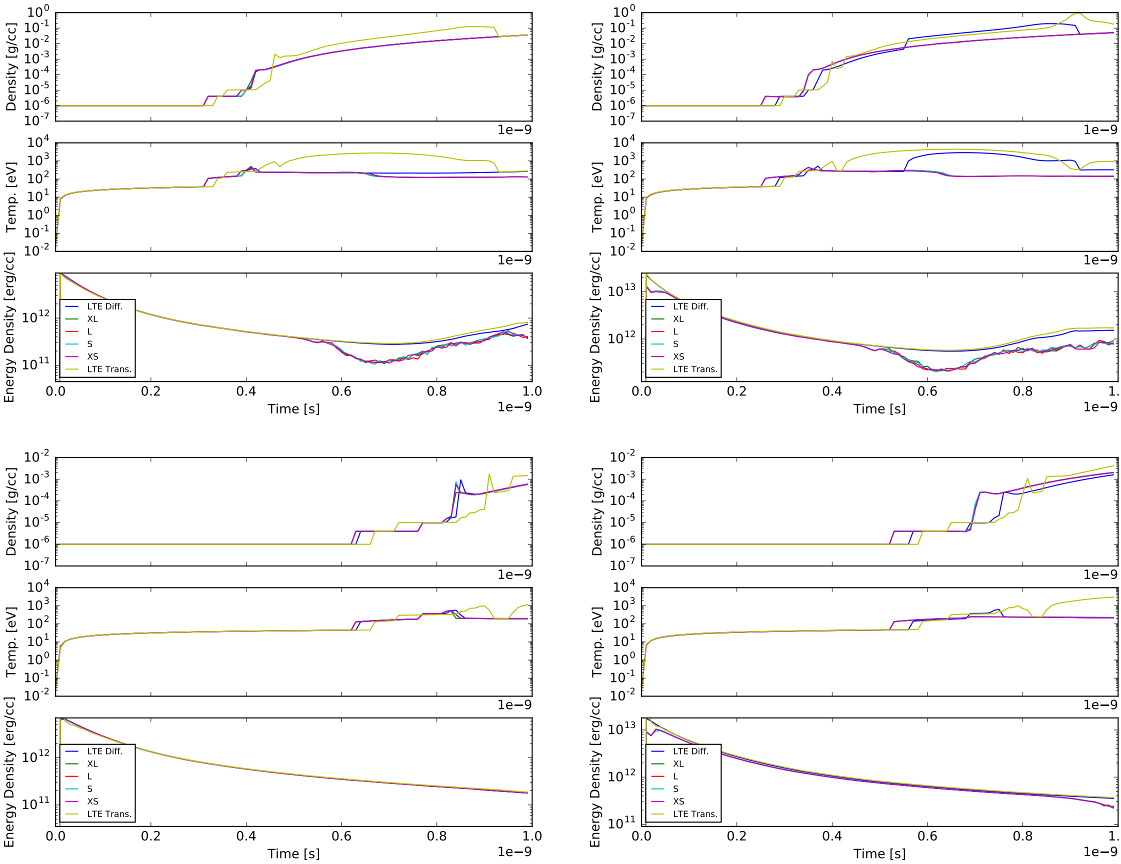


Figure 6.3: These figures have the same layout as in 6.2, but these results were obtained with a cutoff temperature of 500 eV rather than 250 eV.

The spectral energy density comparisons in figure 6.4 show qualitative agreement between LTE transport and LTE diffusion and between the four atomic models used for the NLTE simulations. However, qualitative differences exist between LTE and NLTE as is also observed in the radiation energy density comparison in figures 6.3 and 6.2. Transport appears to have a larger effect in temperature and density, while NLTE appears to have a larger effect in radiation specific quantities.

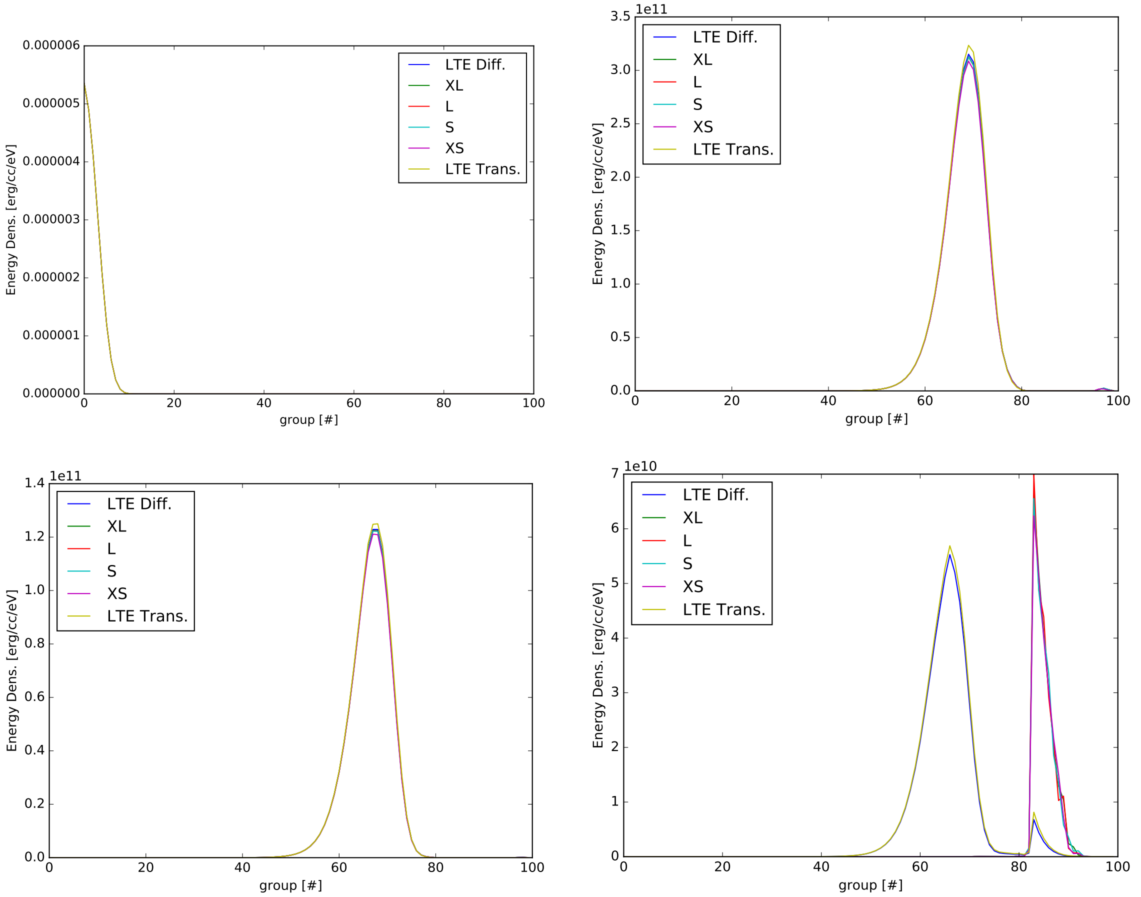


Figure 6.4: This array of figures shows group-wise spectral energy density for the simulation with a plate spacing of 1mm and a hot temperature of 1 keV at 4 different times: 0 ps, 160 ps, 320 ps, and 640 ps (read from left to right, top to bottom).

6.2 Blastwave Diagnostic

In addition to a set of 1D simulations described in section 6.1, a set of two 2D simulations with the same geometry as in the blastwave diagnostic [14] were conducted, but with modified materials for simplicity. The primary change in materials was the use of an iron hohlraum. Both simulations were run with the `xRAGE` RH code using flux limited multigroup diffusion for radiation transport. One simulation was run under the LTE assumipton, while the other used NLTE physics with `inlinlte`.

The geometry is shown in figure 6.5. At early times, slight differences are observed in the NLTE simulation, but those changes do not have much appreciable effect on the blastwave dynamics. However, this problem is designed to mitigate potential NLTE effects and this comparison provides evidence to support that.

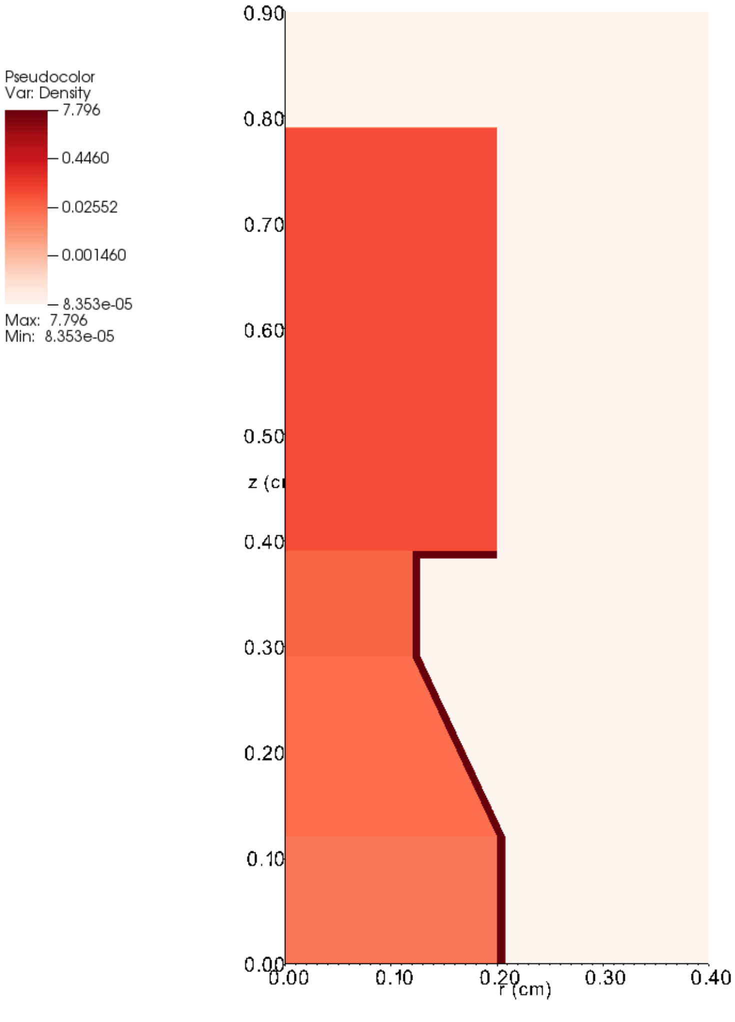


Figure 6.5: Initial Condition for blastwave diagnostic simulation.

Despite the similarities at late times, some noticeable differences exist at earlier times as shown in figure 6.6.

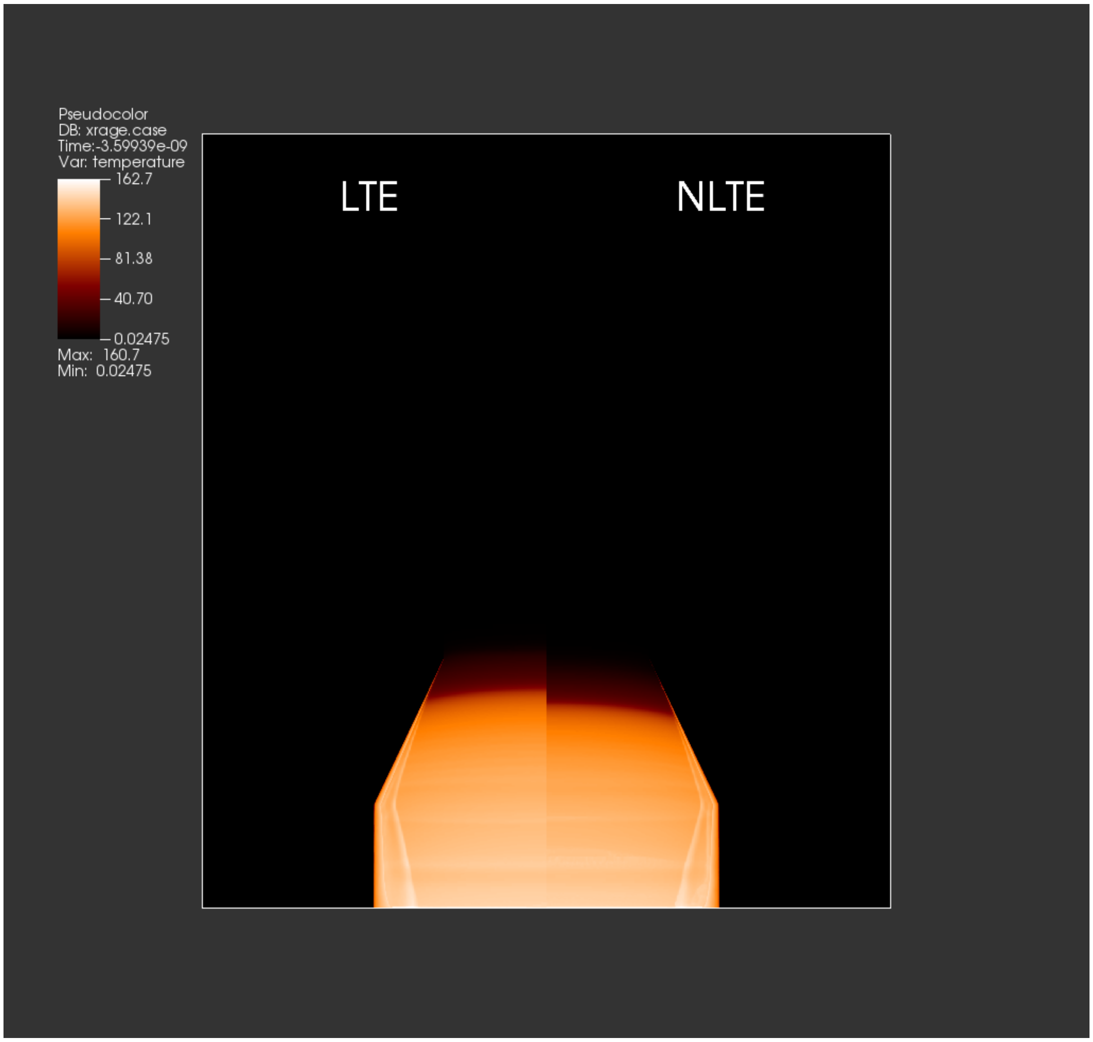


Figure 6.6: LTE to NLTE temperature comparison for the blastwave diagnostic.

A more detailed analysis was completed at 6 spatial locations in which the mass density, material temperature, radiation energy density, and the radiation spectral energy density were compared as a function of time. The first three of the quantities listed are scalar and are shown as function of time. The six locations consist of three axial locations by two radial locations. The radial locations are 0.012 cm off centerline and also at the hohlraum wall surface. The three axial locations occur at 0.13 cm, 0.3 cm, and 0.4cm, starting with 0.13cm in figures 6.7–6.10. The results appear to show minor differences particularly associated with the time at which dynamical features occur. However, at later times both LTE and NLTE simulations show reasonable agreement, especially in the scalar quantities.

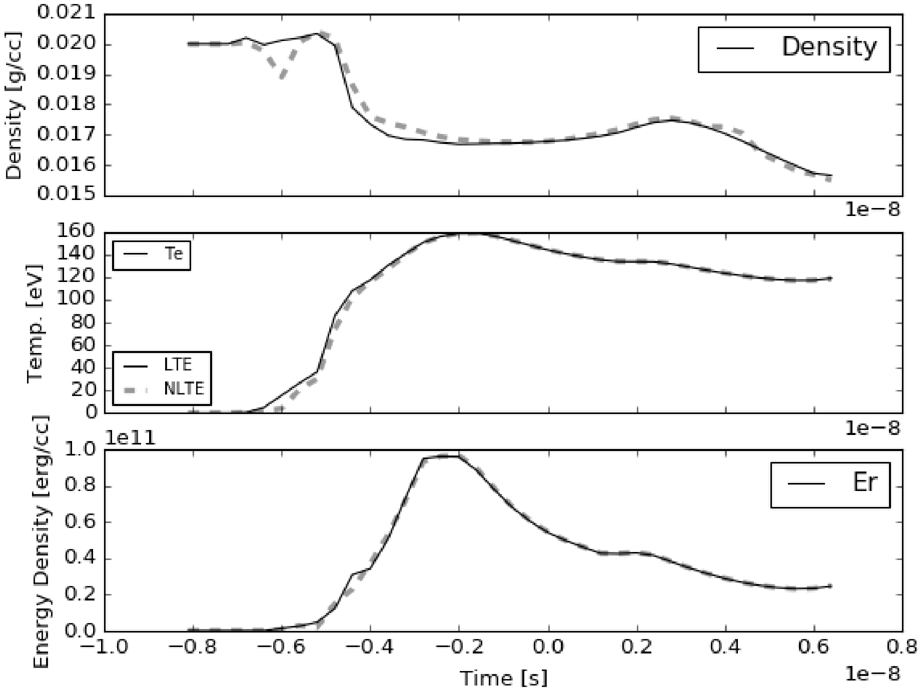


Figure 6.7: This figure shows density (top), material temperature (middle), and radiation energy denisty (bottom) as a function of time near the center of the problem at an axial location of 0.13cm. NLTE results shown is dashed line while LTE results shown in dotted line.

Even the multigroup radiation spectral energy density shows agreement except when the spatial location experiences the most changes, which occurs as the temperature front passes through that axial location.

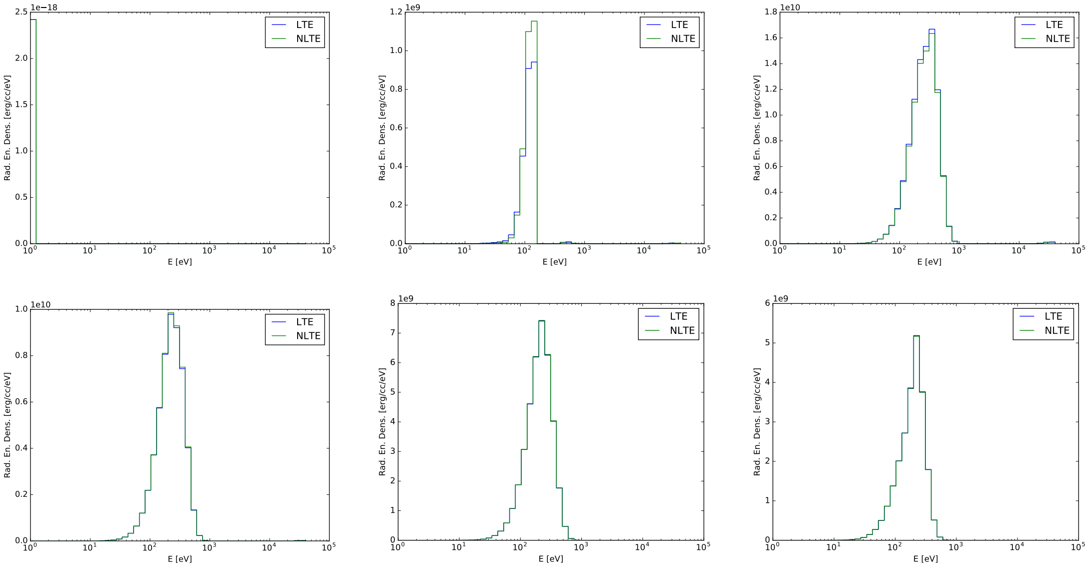


Figure 6.8: This array of figures shows group-wise spectral energy density near the center of the problem at an axial location of 0.13cm at 6 different times: -8.1 ns, -5.6 ns, -2.8 ns, 0 ns, 2.8 ns, and 5.6 ns (read from left to right, top to bottom).

At the hohlraum wall at this axial location, differences are noticeable during the dynamical portions of the simulation but appear to approach each other at later times. Figure 6.9 also shows the interesting phenomenon of compression when the temperature front reaches that portion of the hohlraum, followed by a rapid decrease in temperature due to the hohlraum wall vaporizing and ablating into the surrounding material. This is also accompanied by a rapid increase in temperature.

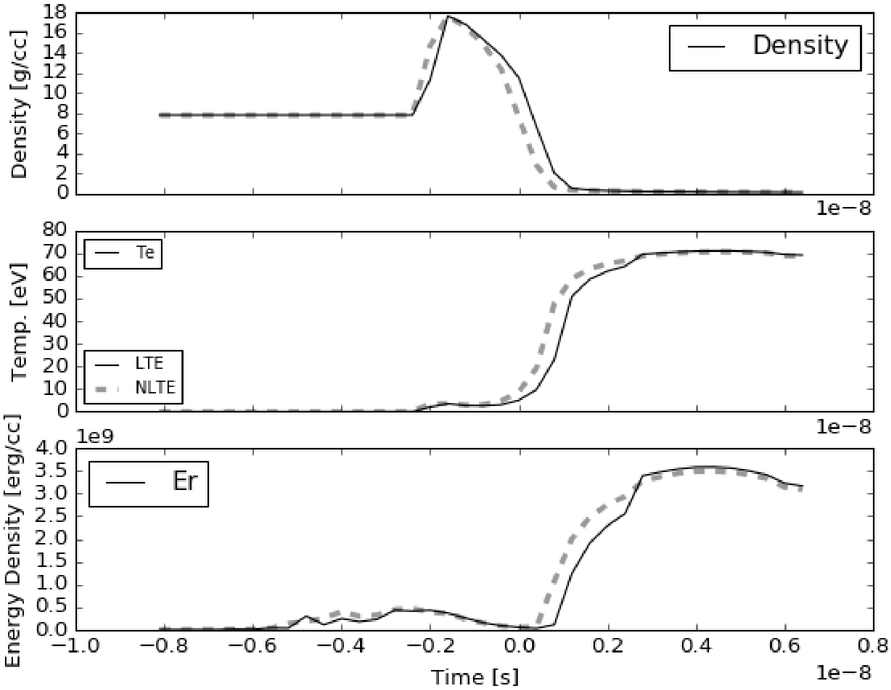


Figure 6.9: This figure shows density (top), material temperature (middle), and radiation energy denisty (bottom) as a function of time near the hohlraum wall at an axial location of 0.13cm. NLTE results shown is dashed line while LTE results shown in dotted line.

As in the previous spectra, modest differences are seen early, followed by closer agreement at late times. Also of note is that only the highest energy groups are appreciably populated at early times due to very high optical depth of the hohlraum wall preventing lower energy photons being emitted by the source from reaching that location.

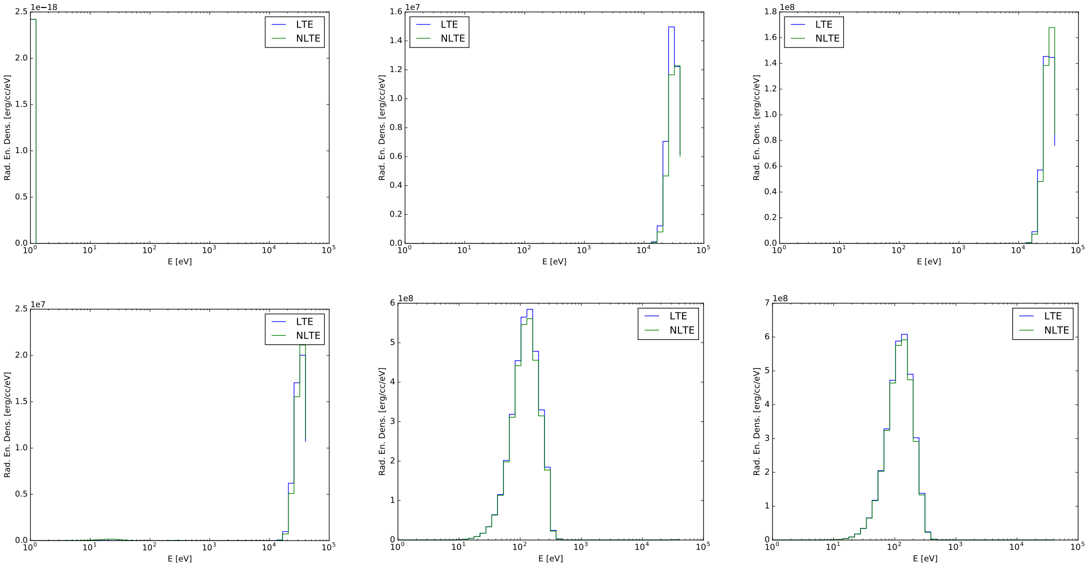


Figure 6.10: This array of figures shows group-wise spectral energy density near the hohlraum wall at an axial location of 0.13cm and has the same layout as figure 6.8.

At this location further from the source, as seen in figure 6.11 the features of the experiment designed to bring about LTE appear to be successful at this spatial location as only modest differences are observed as the front passes.

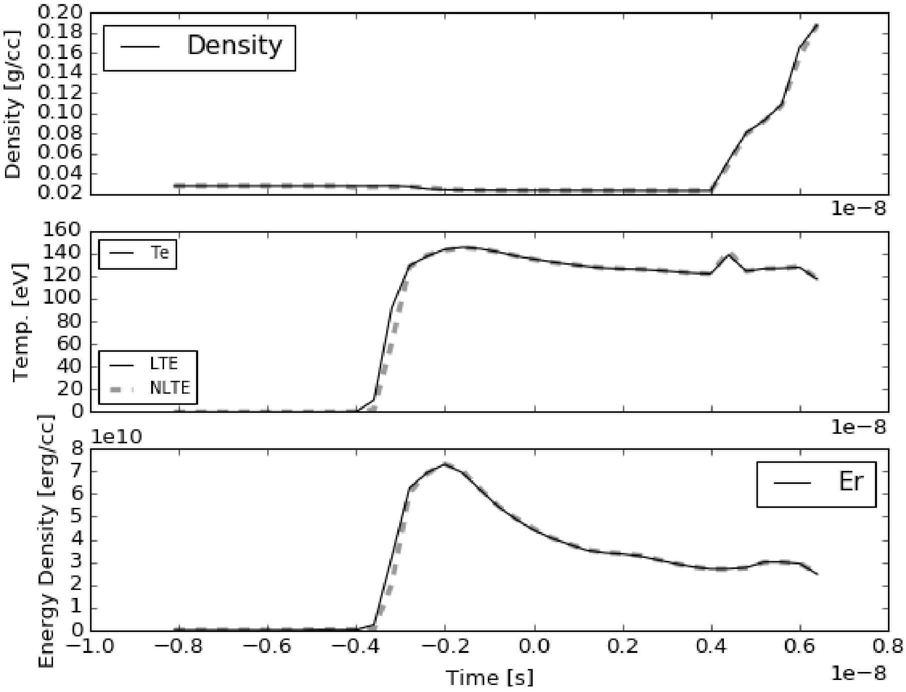


Figure 6.11: This figure shows density (top), material temperature (middle), and radiation energy denisty (bottom) as a function of time near the center of the problem at an axial location of 0.3cm. NLTE results shown is dashed line while LTE results shown in dotted line.

As little differences are noted in figure 6.11, this is appropriately mirrored in figure 6.12. Also of note is that only the highest energy photon groups are populated at early times due to the high optical depth from the source for the lower energy photons.

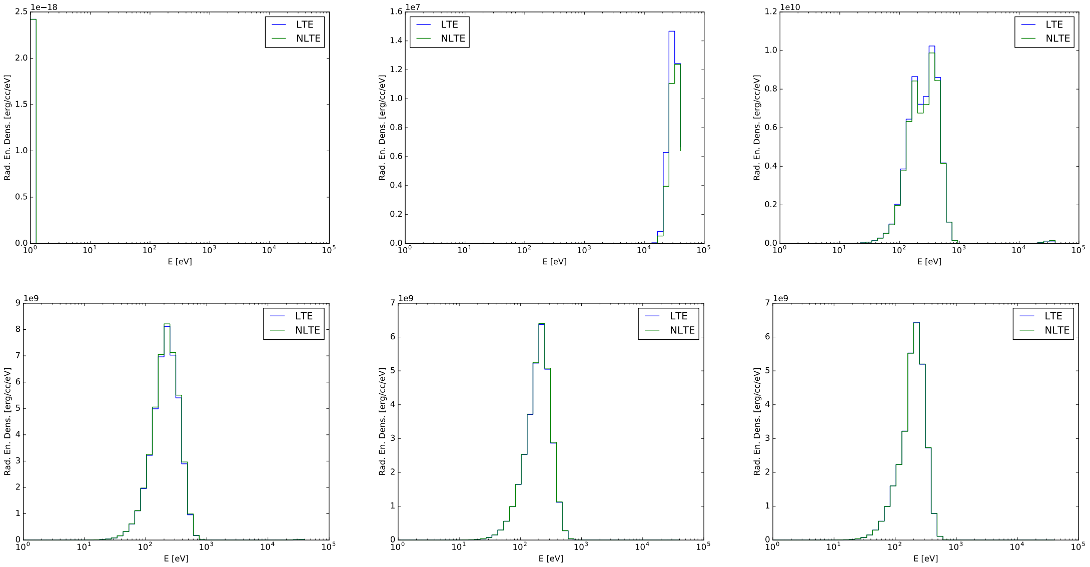


Figure 6.12: This array of figures shows group-wise spectral energy density near the center of the problem at an axial location of 0.3cm and has the same layout as figure 6.8.

At 0.3 cm at the hohlraum wall, as seen in figure 6.13, reasonable agreement is observed with the exception of modest disagreement when the conditions undergo rapid changes. Also of note is that ablation occurs immediately, indicating that this data was taken much closer to the hohlraum surface that at 0.13cm. Data was captured at a location corresponding to point on the mesh, causing variations in the actual location in which data is recorded for the figures.

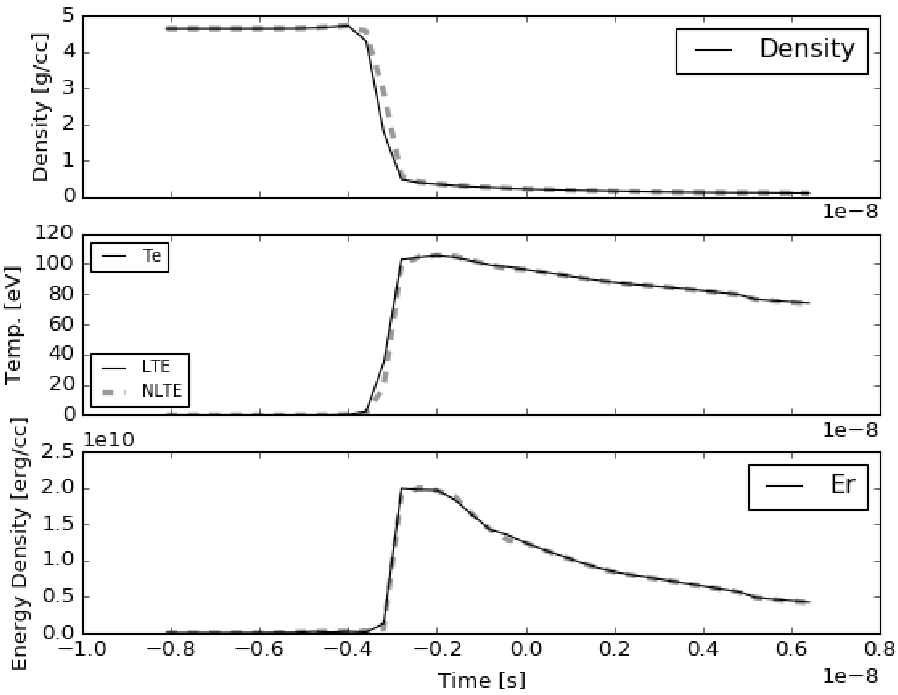


Figure 6.13: This figure shows density (top), material temperature (middle), and radiation energy denisty (bottom) as a function of time near the hohlraum wall at an axial location of 0.3cm. NLTE results shown is dashed line while LTE results shown in dotted line.

The result from figure 6.14 is consistent with the analysis of figure 6.13, that being modest differences when conditions are changing rapidly.

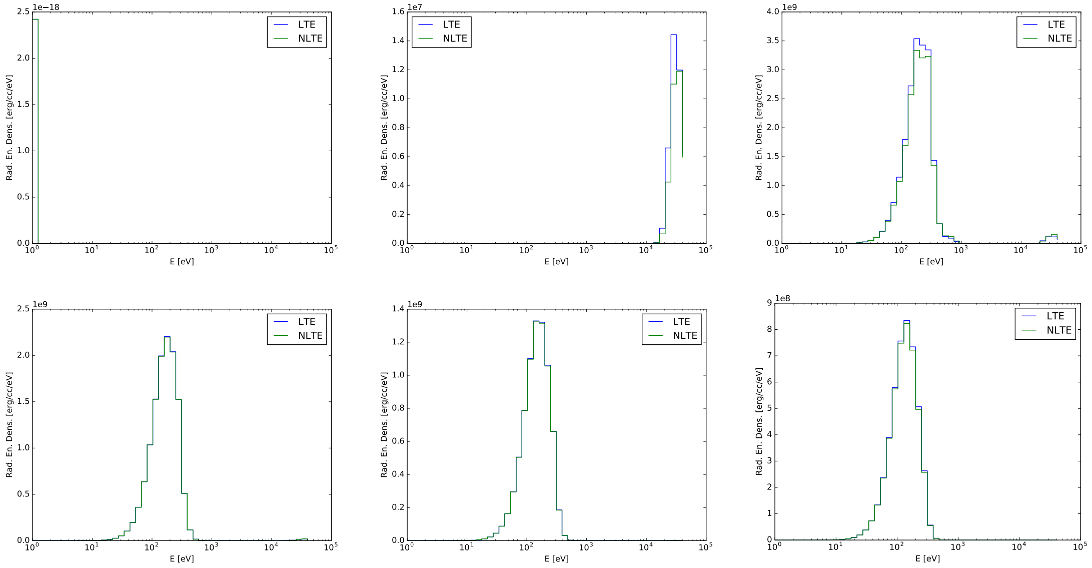


Figure 6.14: This array of figures shows group-wise spectral energy density near the hohlraum wall at an axial location of 0.3cm and has the same layout as figure 6.8.

In figure 6.15, there is a difference in density between LTE and NLTE simulations which is likely caused by an interfacial effect as this data is being take at or near a material interface is which the density is discontinuous. Besides this, only minor differences are observed.

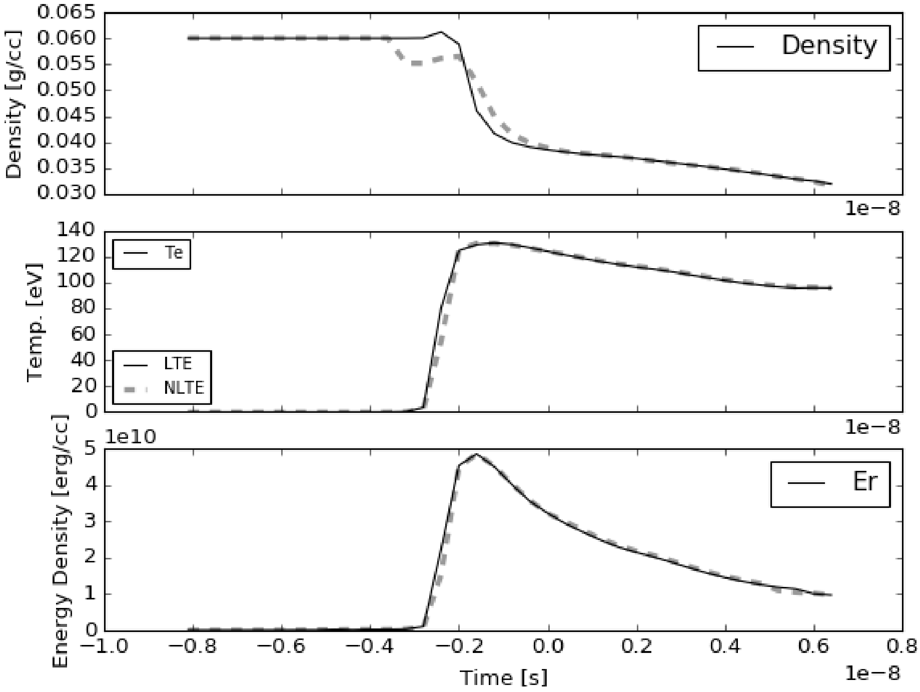


Figure 6.15: This figure shows density (top), material temperature (middle), and radiation energy density (bottom) as a function of time near the center of the problem at an axial location of 0.4cm. NLTE results shown is dashed line while LTE results shown in dotted line.

The more noticeable differences seen in the density in figure 6.15 are not observed in figure 6.16.

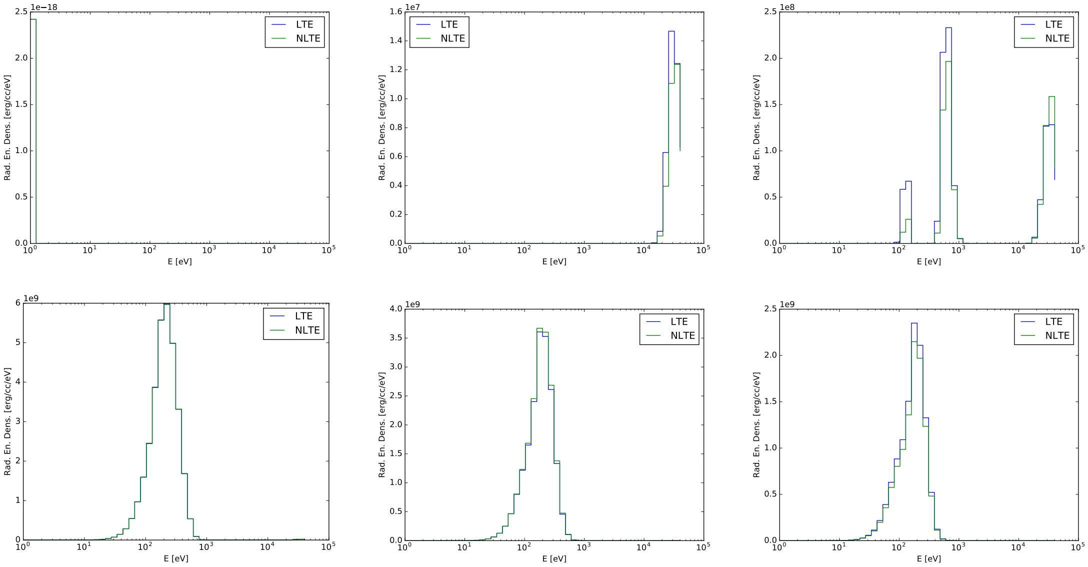


Figure 6.16: This array of figures shows group-wise spectral energy density near the center of the problem at an axial location of 0.4cm and has the same layout as figure 6.8.

Near the hohlraum wall at 0.4 cm above the origin, modest differences are observed. Note that the data was recorded slightly off the wall and into the foam, and thus as the wall ablates into the foam, the density increases.

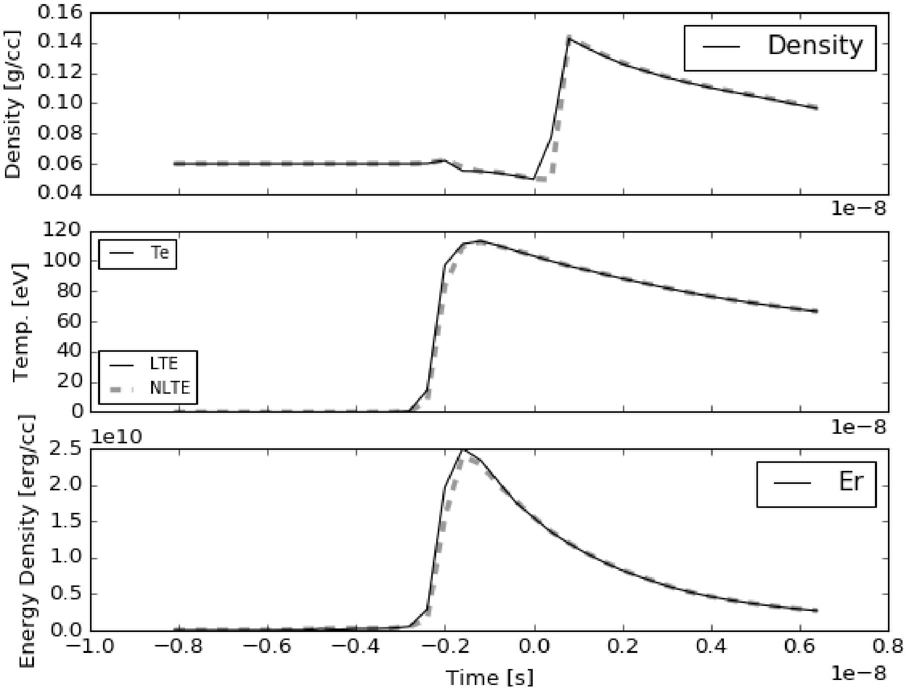


Figure 6.17: This figure shows density (top), material temperature (middle), and radiation energy denisty (bottom) as a function of time near the hohlraum wall at an axial location of 0.4cm. NLTE results shown is dashed line while LTE results shown in dotted line.

The result seen in figure 6.17 is mirrored in figure 6.18.

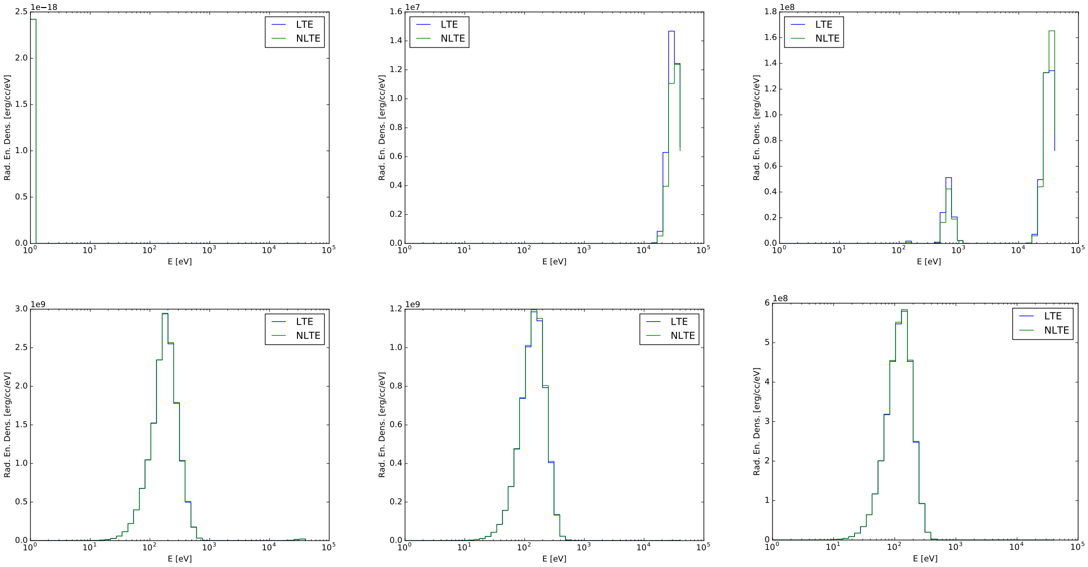


Figure 6.18: This array of figures shows group-wise spectral energy density near the hohlraum wall at an axial location of 0.4cm and has the same layout as figure 6.8.

These axial locations correspond to changes in foam density within the hohlraum. As the temperature front propagates through the system, differences between the LTE and NLTE simulations decrease. This is supported by the data and only minor changes are observed, even in the spectra, at the 0.4 cm axial location, which is the start of the experimentally relevant portion of the simulated domain. This experiment was designed to be in LTE and the data shown support this notion.

7. CONCLUSIONS

Vast improvements in computing technology over the past 2 decades have brought about new opportunities in the realm of scientific computing including topics relevant to the HEDP community. These improvements are realized both through increased geometrical and spatio-temporal resolution as well as through incorporating higher fidelity physical models. This work detailed the incorporation of higher fidelity physics by relaxing the assumption of LTE. This was accomplished through the creation of the `inlinlte` library and demonstrated in the `xRAGE` RH code as well as the `spectra` post-processor. This library showed marked improvements over the prior implementation in `xRAGE` beyond an order of magnitude.

In an RH simulation, the NLTE solver is used only for a single material of interest and only when the temperature exceeds a cutoff value. Traditional domain decomposition methods will yield an imbalance of NLTE work if the simulation contains localized regions that exceed the cutoff temperature. `inlinlte` utilizes both inter-node and intra-node load balancing, which increases hardware utilization and improves performance. A scalable inter-node load balancing algorithm was demonstrated. On-node parallelism was obtained using the `Kokkos` library which provides performance portability through hierarchical parallel patterns, dynamic scheduling, and architecture dependent data structure layouts. Dynamic process mapping is accomplished via `libquo` and allows for efficient MPI + OpenMP usage independent of the parallel nature of the host code. Increased performance for the study of three demonstration problems: a supernova (SN), a 1D problem meant to capture dynamics experienced in inertial confinement fusion (ICF) hohlraums, and a modified version of the blastwave diagnostic experiment performed at Sandia National

Laboratory's (SNL) Z machine. In the supernova modeling, explosive dynamics were modeled using the xRAGE radiation hydrodynamics (RH) code [9] developed at the Los Alamos National Laboratory (LANL). The simulation was run with and without NLTE physics. The results of that RH simulation were post-processed using LANL's suite of light curve and spectra codes [8]. The post-processor was able to leverage `inlinlte`'s monochromatic opacity capabilities to calculate opacities inline in order to obtain simulated spectral luminosities, a metric used by observers to characterize astrophysical events. The use of NLTE physics, both in the RH simulation and in the post-processor, lead to qualitative differences in the resultant spectral luminosities, despite only modest differences in the RH simulation. However, the presence of thresholds with sharp barriers, such as ionization, indicates that deviations from LTE can lead to qualitative differences in relevant quantities when modeling supernovae ejecta. In targeting the HEDP regime, a 1D problem meant to mimic the dynamics possible in a hohlraum, a set of parameters were varied to study their effect on a quantity of interest (QOI), in particular, the time in which 2 ablation fronts collided, which is analogous to the closing of the laser entrance hole (LEH) in ICF. A total 54 configurations were used and run with different assumptions: LTE and NLTE using two different atomic models for iron. This was reduced to 4 configurations that were run under LTE and NLTE with four atomic models. Analysis indicates that the collision time QOI was relatively insensitive to the 3 most complex atomic models used, indicating some detail is needed in the treatment of auto-ionizing states. For non-integrated quantities like the radiation spectral energy density, minor differences were observed. Depending on the purpose of the simulation, relative importance of atomic model complexity will vary.

The blastwave diagnostic simulation was conducted and comparisons between LTE and NLTE were made, resulting in some differences associated with the early dynamics of simulation. More detailed comparisons corroborate importance of NLTE modeling near hohlraum walls, but also support that LTE is a reasonable assumption at late times, in accordance with experimental design.

REFERENCES

[1] Joseph Abdallah, Jr. and Manolo E. Sherrill. The reduced detailed configuration accounting (RDCA) model for NLTE plasma calculations. *High Energy Density Physics*, 4(3–4):124 – 130, 2008.

[2] Joseph Abdallah, Jr., Manolo E. Sherrill, D. P. Kilcrease, C. J. Fontes, H. L. Zhang, and J. Oelgoetz. The reduced detailed configuration accounting (RDCA) model for NLTE plasma spectral calculations. *High Energy Density Physics*, 5(3):204 – 207, 2009.

[3] M. Busquet. Radiation-dependent ionization model for laser-created plasmas. *Physics of Fluids B: Plasma Physics*, 5(11):4191–4206, 1993.

[4] R Paul Drake. *High-Energy-Density Physics*. Springer-Verlag Berlin Heidelberg, 1 edition, 2006.

[5] H. Carter Edwards and Daniel A. Ibanez. Kokkos' task dag capabilities. Technical Report SAND2017-10464, Sandia National Laboratories, Albuquerque, New Mexico 87185, September 2017.

[6] H. Carter Edwards, Christian R. Trott, and Daniel Sunderland. Kokkos: Enabling manycore performance portability through polymorphic memory access patterns. *Journal of Parallel and Distributed Computing*, 74(12):3202 – 3216, 2014. Domain-Specific Languages and High-Level Frameworks for High-Performance Computing.

[7] C J Fontes, H L Zhang, J Abdallah Jr, R E H Clark, D P Kilcrease, J Colgan, R T Cunningham, P Hakel, N H Magee, and M E Sherrill. The Los Alamos suite

of relativistic atomic physics codes. *Journal of Physics B: Atomic, Molecular and Optical Physics*, 48(14):144014, 2015.

[8] Lucille H. Frey, Wesley Even, Daniel J. Whalen, Chris L. Fryer, Aimee L. Hungerford, Christopher J. Fontes, and James Colgan. The los alamos supernova light-curve project: Computational methods. *The Astrophysical Journal Supplement Series*, 204(2):16, 2013.

[9] Michael Gittings, Robert Weaver, Michael Clover, Thomas Betlach, Nelson Byrne, Robert Coker, Edward Dendy, Robert Hueckstaedt, Kim New, W Rob Oakes, Dale Ranta, and Ryan Stefan. The RAGE radiation-hydrodynamic code. *Computational Science & Discovery*, 1(1):015005, 2008.

[10] S. K. Gutiérrez, K. Davis, D. C. Arnold, R. S. Baker, R. W. Robey, P. McCormick, D. Holladay, J. A. Dahl, R. J. Zerr, F. Weik, and C. Junghans. Accommodating thread-level heterogeneity in coupled parallel applications. In *2017 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 469–478, May 2017.

[11] Ibrahim N Hajj and Stig Skelboe. A multilevel parallel solver for block tridiagonal and banded linear systems. *Parallel Computing*, 15(1):21 – 45, 1990.

[12] Michael A. Heroux, Roscoe A. Bartlett, Vicki E. Howle, Robert J. Hoekstra, Jonathan J. Hu, Tamara G. Kolda, Richard B. Lehoucq, Kevin R. Long, Roger P. Pawlowski, Eric T. Phipps, Andrew G. Salinger, Heidi K. Thornquist, Ray S. Tuminaro, James M. Willenbring, Alan Williams, and Kendall S. Stanley. An overview of the trilinos project. *ACM Trans. Math. Softw.*, 31(3):397–423, 2005.

[13] S. P. Hirshman, K. S. Perumalla, V. E. Lynch, and R. Sanchez. Bcyclic: A parallel block tridiagonal matrix cyclic solver. *Journal of Computational Physics*,

229(18):6392 – 6404, 2010.

[14] Thomas E. Tierney IV, Heidi E. Tierney, George C. Idzorek, Robert G. Watt, Robert R. Peterson, Darrell L. Peterson, Christopher L. Fryer, Mike R. Lopez, Michael C. Jones, Daniel Sinars, Gregory A. Rochau, and James E. Bailey. Blast wave energy diagnostic. *Review of Scientific Instruments*, 79(10):10E919, 2008.

[15] N. M. Law, S. R. Kulkarni, R. G. Dekany, E. O. Ofek, R. M. Quimby, P. E. Nugent, J. Surace, C. C. Grillmair, J. S. Bloom, M. M. Kasliwal, L. Bildsten, T. Brown, S. B. Cenko, D. Ciardi, E. Croner, S. G. Djorgovski, J. van Eyken, A. V. Filippenko, D. B. Fox, A. Gal-Yam, D. Hale, N. Hamam, G. Helou, J. Henning, D. A. Howell, J. Jacobsen, R. Laher, S. Mattingly, D. McKenna, A. Pickles, D. Poznanski, G. Rahmer, A. Rau, W. Rosing, M. Shara, R. Smith, D. Starr, M. Sullivan, V. Velur, R. Walters, and J. Zolkower. The Palomar Transient Factory: System Overview, Performance, and First Results. PASP, 121:1395, December 2009.

[16] MJ O'Brien, PS Brantley, KI Joy, and F Gygi. Scalable domain decomposed monte carlo particle transport. 2013.

[17] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 3 edition, 2007.

[18] Yuri Ralchenko, editor. *Modern Methods in Collisional-Radiative Modeling of Plasmas*. 90. Springer International Publishing, New York, 1 edition, 2016.

[19] M. D. Rosen, H.A. Scott, D. E. Hinkel, E. A. Williams, D. A. Callahan, R. P. J. Town, L. Divol, P. A. Michel, W. L. Kruer, L. J. Suter, R. A. London, J. A. Harte, and G. B. Zimmerman. The role of a detailed configuration accounting

(dca) atomic physics package in explaining the energy balance in ignition-scale hohlraums. *High Energy Density Physics*, 7(3):180 – 190, 2011.

[20] Sudip K. Seal, Kalyan S. Perumalla, and Steven P. Hirshman. Revisiting parallel cyclic reduction and parallel prefix-based algorithms for block tridiagonal systems of equations. *Journal of Parallel and Distributed Computing*, 73(2):273 – 280, 2013.

[21] Christopher Stone, Earl Duque, Yao Zhang, David Car, Roger Davis, and John Owens. *GPGPU parallel algorithms for structured-grid CFD codes*. American Institute of Aeronautics and Astronautics, 2017/01/18 2011.

[22] C. Thurl, P. D. Sackett, and P. H. Hauschildt. Resolving stellar atmospheres. I. The Hα line and comparisons to microlensing observations. A&A, 455:315–324, August 2006.

[23] D. J. Whalen, C. L. Fryer, D. E. Holz, A. Heger, S. E. Woosley, M. Stiavelli, W. Even, and L. H. Frey. Seeing the First Supernovae at the Edge of the Universe with JWST. ApJ, 762:L6, January 2013.

[24] Y.B. Zel'dovich and Y.P. Raizer. *Physics of Shock Waves and High-Temperature Hydrodynamic Phenomena*. Dover Books on Physics. Dover Publications, 2012.