PHYSICS AND COMPUTER ARCHITECTURE INFORMED IMPROVEMENTS TO THE IMPLICIT MONTE CARLO METHOD

A Dissertation

by

ALEX ROBERTS LONG

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 G. McClarren
Committee Members,	Jim Morel
	Marvin Adams
	Nancy Amato
Head of Department,	Yassin Hassan

May 2016

Major Subject: Nuclear Engineering

ABSTRACT

The Implicit Monte Carlo (IMC) method has been a standard method for thermal radiative transfer for the past 40 years. In this time, the hydrodynamics methods that are coupled to IMC have evolved and improved, as have the supercomputers used to run large simulations with IMC. Several modern hydrodynamics methods use unstructured non-orthogonal meshes and high-order spatial discretizations. The IMC method has been used primarily with simple Cartesian meshes and always has a first order spatial discretization. Supercomputers are now made up of compute nodes that have a large number of cores. Current IMC parallel methods have significant problems with load imbalance. To utilize many core systems, algorithms must move beyond simple spatial decomposition parallel algorithms. To make IMC better suited for large scale multiphysics simulations in high energy density physics, new spatial discretizations and parallel strategies are needed. Several modifications are made to the IMC method to facilitate running on node-centered, unstructured tetrahedral meshes. These modifications produce results that converge to the expected solution under mesh refinement. A new finite element IMC method is also explored on these meshes, which offer a simulation runtime benefit but does not perform correctly in the diffusion limit. A parallel algorithm that utilizes on-node parallelism and respects memory hierarchies is studied. This method scales almost linearly when using physical cores on a node and benefits from multiple threads per core. A multi-compute node algorithm for domain decomposed IMC that passes mesh data instead of particles is explored as a means to solve load balance issues. This method scales better than the particle passing method on highly scattering problems with short time steps.

DEDICATION

This is dedicated to you, the reader.

ACKNOWLEDGEMENTS

My wife Jane has supported me and pushed me over the entire course of my graduate education. Even when other people close to me doubted, she has always been positive. Jane has also endured living in two small college towns while I've been in graduate school.

My parents Kristi and Ken Long bought me flash cards that with astronomical objects on them when I was about 5 years old. I have no idea why I wanted these, nor do I claim that I was some kind of child prodigy. My parents have always fostered my curiosity and encouraged me to follow my own academic interests. This intellectual support has always been accompanied with love and understanding.

My advisor, Ryan McClarren, has been patient with me and my questions about the wide world of computational physics over the past three years. He has the ability to translate a poorly posed question into the question I should have asked and then deliver a bespoke answer. He has been an exceptional advisor.

All of my friends at Texas A & M made graduate school so much more enjoyable. Coming from a school with a very small transport group, I was surprised to find the A & M transport students were a welcoming, supportive family. Andrew Till has been a great friend and has provided immense technical support. Jim Feguson, Carlolyn McGraw, Don Bruss, Daniel Holladay, Simon Bolding and so many others were and are my brothers in arms in the duty-dance with death that is graduate school.

I've received so much support from the wonderful people at Los Alamos National Laboratory. My mentors Jacob Waltz and John Wohlbier took a chance bringing by bringing me out to the lab full time, for which I am very grateful. Working with the transport team in CCS-2 has been invaluable. Gabe Rockefeller, Allan Wollaber

Matt Cleaveland have provided technical advice and perspective on many occasions. Gabe Rockefeller also provided the idea for the alternate domain decomposition and let me explore it in this work. Kelly Thompson has helped me navigate C++, MPI and the delicate art of linking libraries on HPC machines. The kindness and support of everyone at Los Alamos has made this process enjoyable and enriching.

Joe and Emily Zerr have been great friends to me over the past two years, making the otherwise boring town of Los Alamos a misanthrope's paradise.

Nick Gentile, my mentor from my first summer at LLNL, introduced me to computational physics and has continued to offer guidance and advice in technical and career matters.

NOMENCLATURE

CDF	Cumulative Distribution Function
CFEM	Continuous Finite Element Method
DFEM	Discontinuous Finite Element Method
HEDP	High Energy Density Physics
HPC	High Performance Computing
ICF	Inertial Confinement Fusion
IMC	Implicit Monte Carlo
LTE	Local Thermodynamic Equilibrium
MPI	Message Passing Interface
PDF	Probability Density Function
RDMA	Remote Direct Memory Access
RMA	Remote Memory Access
SIMC	Symbolic Implicit Monte Carlo
TRT	Thermal Radiative Transfer

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

1.1 The Physics of Radiation Transport

What humans perceive as light is made of up photons. Photons travel at 300 million meters per second and have the form of both particles and waves. Photon interactions with matter are the source of many familiar and unfamiliar phenomena. This interaction illuminate rooms, propels solar sails on spacecraft [45], heats the earth and drives supernova explosions. There are many processes in nature that produce photons. *Bound-bound emission* produces photons at discrete energy levels, this comes from the movement of electrons from high energy levels to lower energy levels. *Bound-free transitions* produce photons with a continuous spectra. When a free electron moves to a bound state a photon is emitted with an energy equal to the difference between the free and bound state. *Free-free transitions* also produce photons with a continuous spectra. Free-free transitions occur when free electrons or ions move from a high energy state to a lower energy state. Emission from electrons or ions slowing down or changing direction is called *Bremsstrahlung radiation.* These processes also describe the absorption of photons. An absorbed photon that moves an electron to a higher energy level is a bound-bound absorption. Bound-free absorption occurs when a photon provides the energy for an electron to leave its state. A photon being absorbed by a free electron is free-free absorption. The emission of photons coincides with a decrease in the energy of matter, likewise the absorption of a photon causes an increase in the energy of matter. This interplay between the energy in matter, the energy in radiation, emission and absorption is known as thermal radiative transfer (TRT).

1.1.1 Approximations to the Physics of Radiation Transport

When simulating TRT, several approximations are made to the physics of emission and absorption to make simulations tractable. The first approximation is that the matter is in Local Thermodynamic Equilibrium (LTE). This assumption means the energy of the material is effectively described by a temperature. Instead of simulating the complex interactions of electrons and ions that lead to bound-bound, bound-free and free-free emission, LTE assumes that absorption and emission processes are in equilibrium and the net emission is approximated as being Planckian. In Planckian emission, the emitted photon energy at a given frequency, ν, from a material at a temperature, T is:

$$B(\nu, T) = \frac{2h\nu^3}{c^2} \frac{1}{\left(e^{h\nu/T} - 1\right)}. \tag{1.1}$$

The emission of photons is also influenced by *induced processes*—the probability of a photons entering a new state (frequency, angle and polarization) is a function of the number of photons in that state [50]. Induced processes greatly complicate the solution process in TRT because it makes the scattering term non-linear. For simplicity, induced processes are often neglected in TRT.

The absorption processes are approximated by an *opacity*, which is probability of interaction per unit length. Opacities are used to describe the absorption and scattering interactions of a photon with matter. The absorption opacity approximates the bound-bound, bound-free and free-free interactions with a single value per material, density, temperature and frequency:

$$\sigma_{\mathrm{a}}(\rho, \nu, T) = \sigma_{\mathrm{bb}} + \sigma_{\mathrm{bf}} + \sigma_{\mathrm{ff}}, \tag{1.2}$$

where σ is the opacity of a given type, ρ is the density, ν is the frequency of the interacting photon and T is the temperature of the material. The absorption opacity is generally a strong function of temperature and photon frequency.

There are two types of photon scattering interactions—coherent (Thompson scattering) and incoherent (Compton scattering). A coherent scattering interaction changes the angle of a photon but not its energy. Incoherent interactions change the photon's energy and angle. At high temperatures, photons can gain energy from incoherent interactions. Because Thompson and Compton scattering have different effects on the photon, they cannot be combined into a single opacity value.

1.1.2 Applications of TRT

The total energy emitted by matter is obtained by integrating Eq. (1.1) over all frequencies:

$$\int_0^\infty B(\nu, T)\ d\nu = \frac{acT^4}{4\pi}. \tag{1.3}$$

The emitted energy is proportional to temperature to the fourth power. As the temperature of a system increases, TRT becomes a more important component of heat transfer. At these temperatures, the physical system is difficult to contain. The difficulty containing systems at high temperatures means that building physical experiments is impractical and thus expensive. At temperatures where TRT is important, computer simulation is often the most practical way to study high energy physical systems. These systems fall into the regime called high energy density physics (HEDP). It is defined by the National Academy of Sciences to be systems with a pressure greater than one million atmospheres [12]. Inertial confinement fusion is one example of a problem in HEDP. In inertial confinement, fusion lasers are used to heat and compress fuel to the point of thermonuclear fusion. In one form of inertial confinement fusion called *indirect drive*, lasers are used to heat a gold cylinder surrounding the fuel. The cylinder, called a *hohlraum*, is heated in such a way that it radiates X-rays that uniformly heat the surface of the fuel pellet. There are many HEDP applications in astrophysics—TRT is particularly important in determining *light curves*. In these problems the measured spectrum of light from a supernova is compared to the spectrum produced by a simulation [33] [38] [61]. This problem is used to determine the composition of supernova precursors as well as the dynamics of a supernova explosion. In both the inertial confinement fusion and light curve problems thermal radiative transfer is coupled to other simulated physics, meaning that TRT directly affects the evolution of the physical system.

1.1.3 Equations of TRT

Photons emitted by a material move through a physical system—they can be absorbed, scattered or stream through the surrounding material. The photon energy density in time, space, angle and frequency is described by the Boltzmann transport equation. This transport equation is coupled to a material energy balance equation. Absorbed photons increase the material energy and emitted photons remove energy from the material. The radiative energy balance, generally called the *transport equation*, and the material energy balance form the equations of radiative transfer:

$$\begin{aligned}\frac{1}{c}\frac{\partial I(x,\nu,\Omega,t)}{\partial t} &+ \Omega\cdot\nabla I(x,\nu,\Omega,t) + \sigma_{\mathrm{a}}(x,\nu,T)I(x,\nu,\Omega,t)\\ &= \sigma_{\mathrm{a}}(x,\nu,T)B(\nu,T), \end{aligned} \tag{1.4}$$

Parameter	Description
x	position
Ω	angle
ν	frequency
t	time
$I(x,\nu,\Omega,t)$	Radiation intensity
$U_{\mathrm{m}}(x,T,t)$	Material energy density
$\sigma_{\mathrm{a}}(x,\nu,T)$	Absorption opacity
c_{V}	Heat capacity
$B(\nu,T)$	Planck emission
$S_{\mathrm{m}}(x,t)$	Material energy source
$S_{\mathrm{r}}(x,\Omega,\nu,t)$	Radiation source

Table 1.1: Parameters in the TRT equations

$$\frac{\partial U_{\mathrm{m}}(x,T,t)}{\partial t} = \int_0^\infty \int_0^{4\pi} \sigma_{\mathrm{a}}(x,\nu,T) I(x,\nu,\Omega,t) d\Omega d\nu - \int_0^\infty \sigma_{\mathrm{a}}(x,\nu,T) B(\nu,T) d\nu + S_{\mathrm{m}}(x,t). \tag{1.5}$$

The equations are presented without the scattering terms. The material energy density, U_m, is related to the material temperature by the heat capacity, c_{V}:

$$U_{\mathrm{m}} = \int_0^T c_{\mathrm{V}}(T)\ dT \tag{1.6}$$

The parameters of Eqs. (1.4) to (1.6) are shown in Table (1.1).

The material and radiative energy balance equations are coupled to each other by the emission term, $\sigma_{\mathrm{a}} B$, and the intensity I. The emission term is proportional to temperature to the fourth power, so the TRT equations are nonlinear in temperature. The TRT equations are linear in intensity, I.

This work is focused on solving the TRT equations with the Implicit Monte Carlo method in a manner that is accurate and efficient.

1.1.4 Thesis Outline

The remainder of this dissertation is organized as follows:

Section 2 includes a derivation of the Implicit Monte Carlo method for Thermal Radiative Transfer for multigroup and grey transport. The details of implementing the IMC method are outlined. The implementation of IMC leads to a discussion on the factors that affect simulation runtime. A brief history of progress in IMC is discussed as well as the current state of IMC research.

Section 3 is about alternate spatial discretizations for IMC designed to be used with point-centered hydrodynamics schemes. The consequences of applying a spatial discretization to the IMC equations are explained. A node-centered method is derived for low-memory cases. A method is analyzed that is similar to corner balance with piecewise constant data. Finally, a discontinuous finite element method is derived and the implementation details are explained. The results for each method are compared to standard IMC methods on several test problems. This work builds upon the author's previous work [37].

Section 4 contains specific sampling methods needed to implement the IMC method on tetrahedra and corners. This includes efficient techniques for sampling from uniform distributions on tetrahedra and corners. Implementing IMC also requires a method for sampling from linear distributions, two methods for doing this on tetrahedra and corners are also in Section 4.

In Section 5, the implicit capture method of variance reduction is explored in the context of the finite element method. Several methods are proposed for handling implicit capture with linear basis functions and the performance of each method is compared on a variety of opacity and shape function cases.

Section 6 examines algorithms for parallelizing IMC. The computational requirements of running HEDP problems are detailed, which informs the need for efficient parallel algorithms. A parallel algorithm for on-node, shared memory IMC is explained and analyzed on two different architectures. A multi-node, domain-decomposed method is proposed that allows particles to request data from other parallel processes. Two different implementations for this method are compared on several test cases and then compared to the standard domain-decomposed IMC method.

Section 7 is a summary of the results. Deficiencies and benefits are discussed and future work for spatial discretizations and parallel methods is recommended.

2. THE IMPLICIT MONTE CARLO METHOD

2.1 Methods of Solving the TRT Equations

There are a number of methods for solving the TRT equations that employ different linearizations and discretizations. These methods fall into four general categories: diffusion theory, discrete ordinates, spherical harmonics, and Monte Carlo. The Monte Carlo method for particle transport was developed at Los Alamos National Laboratory in the 1940's [44] and proved to be an effective way to solve neutron criticality problems [31] [24]. The Monte Carlo method was applied to nonlinear, time-dependent TRT by Fleck in a 1963 paper [20]. Fleck applied a simple linearization by assuming that the temperature was constant within a time step. This linearization required very small timestep sizes for numerical stability and also did not hold an equilibrium state. Fleck and Cummings vastly improved upon the simple linearization with what is generally referred to as the Implicit Monte Carlo (IMC) method [20] [1]

Before the IMC equations are derived, a few terms need to be introduced that are used in the derivation. A useful quantity with TRT equations is the equilibrium radiation energy density:

$$U_{\mathrm{r}}(T) = aT^4 \tag{2.1}$$

The emission term in the transport equation can be written in terms of the equilibrium radiation energy density, this involves the Planck weighted opacity:

[1] The IMC method is also referred to as the Fleck and Cummings method.

$$\sigma_{\mathrm{a,p}} = \frac{\int\limits_0^\infty \sigma_{\mathrm{a}}(\nu,T)B(\nu,T)\ d\nu}{\int\limits_0^\infty B(\nu,T)\ d\nu} = \frac{\int\limits_0^\infty \sigma_{\mathrm{a}}(\nu,T)B(\nu,T)\ d\nu}{cU_{\mathrm{r}}(T)}, \tag{2.2}$$

This is used to replace the emission term with an emission spectrum multiplied by a source:

$$\sigma_{\mathrm{a}}(\nu,T)B(\nu,T) = \sigma_{\mathrm{a}}(\nu,T)B(\nu,T)\frac{\int\limits_0^\infty \sigma_{\mathrm{a}}(\nu,T)B(\nu,T)\ d\nu}{\int\limits_0^\infty \sigma_{\mathrm{a}}(\nu,T)B(\nu,T)\ d\nu} = \chi(\nu,T)c\sigma_{\mathrm{a,p}}(T)U_{\mathrm{r}}(T), \tag{2.3}$$

where χ is the emission spectrum:

$$\chi(\nu,T) = \frac{\sigma_{\mathrm{a}}(\nu,T)B(\nu,T)}{\int\limits_0^\infty \sigma_{\mathrm{a}}(\nu,T)B(\nu,T)\ d\nu}. \tag{2.4}$$

Equation (2.4) now has the form of a probability density function (PDF) in frequency. When a PDF in some variable x is integrated over a range (a,b), it yields the probability of $a < x < b$. A PDF is equal to unity when integrated over the entire domain of the variable it represents. For example, when Eq. (2.4) is integrated over the entire frequency range, $\nu \in (0,\infty)$, it evaluates to 1.0. The opacity is usually given as discrete data, meaning the Planck weighted opacity and the emission spectrum can be evaluated numerically. In the material energy balance, the emission term is integrated over all frequencies and $\chi(\nu,T)$ becomes unity:

$$\int\limits_0^\infty \sigma_{\mathrm{a}}(\nu,T)B(\nu,T)\ d\nu = \int\limits_0^\infty \chi(\nu,T)c\sigma_{\mathrm{a,p}}(T)U_{\mathrm{r}}(T)\ d\nu = c\sigma_{\mathrm{a,p}}(T)U_{\mathrm{r}}(T), \tag{2.5}$$

The TRT equations written in terms of U_{r} are:

$$\frac{1}{c}\frac{\partial I(x,\nu,\Omega,t)}{\partial t}+\Omega\cdot\nabla I(x,\nu,\Omega,t)+\sigma_{\mathrm{a}}(x,\nu,T)I(x,\nu,\Omega,t)=\chi(\nu,T)c\sigma_{\mathrm{a,p}}(T)U_{\mathrm{r}}(T) \tag{2.6}$$

$$\frac{\partial U_{\mathrm{m}}(x,T,t)}{\partial t}=\int\limits_0^\infty\int\limits_0^{4\pi}\sigma_{\mathrm{a}}(x,\nu,T)I(x,\nu,\Omega,t)d\Omega d\nu-c\sigma_{\mathrm{a,p}}(T)U_{\mathrm{r}}(T). \tag{2.7}$$

A relationship between the change in equilibrium radiation energy density to the material energy density is needed to write the material energy balance as a time derivative of U_r:

$$\beta=\frac{\partial U_{\mathrm{r}}}{\partial T}\frac{\partial T}{\partial U_{\mathrm{m}}}=\frac{4aT^3}{c_{\mathrm{V}}} \tag{2.8}$$

Such that:

$$\frac{\partial U_{\mathrm{m}}}{\partial t}=\frac{\partial U_{\mathrm{m}}}{\partial U_{\mathrm{r}}}\frac{\partial U_{\mathrm{r}}}{\partial t}=\frac{1}{\beta}\frac{\partial U_{\mathrm{r}}}{\partial t} \tag{2.9}$$

The material energy balance equation becomes:

$$\frac{1}{\beta}\frac{\partial U_{\mathrm{r}}}{\partial t}=\int\limits_0^\infty\int\limits_0^{4\pi}\sigma_{\mathrm{a}}(x,\nu,T)I(x,\nu,\Omega,t)\;d\Omega\;d\nu-c\sigma_{\mathrm{a,p}}(x,T)U_{\mathrm{r}}(T)+S_{\mathrm{m}}(x,t). \tag{2.10}$$

2.2 Derivation of the IMC Equations

No approximations have been made to the TRT equations in forming Eq. (2.10). The derivation of the IMC equations begins by integrating the material energy balance (in terms of equilibrium radiation energy density) over a timestep, Δt:

$$\int_{t^n}^{t^{n+1}} \frac{1}{\beta}\frac{\partial U_{\mathrm{r}}}{\partial t} = \int_0^{\infty}\int_0^{4\pi}\int_{t^n}^{t^{n+1}} \sigma_{\mathrm{a}}(x,\nu,T)I(x,\nu,\Omega,t)\ dt\ d\Omega\ d\nu$$

$$- \int_{t^n}^{t^{n+1}} c\sigma_{\mathrm{a,p}}(x,T)U_{\mathrm{r}}(T)\ dt + \int_{t^n}^{t^{n+1}} S_{\mathrm{m}}(x,t)\ dt. \tag{2.11}$$

To perform this integration, two approximations are made:

- The opacity is assumed to be constant over a timestep and held at the beginning of the timestep, t^n
- The β term is assumed to be constant over a timestep and held at the beginning of the timestep, t^n. This term includes temperature and heat capacity.

The equations have now been partially discretized in time:

$$\frac{1}{\beta^n}\int_{t^n}^{t^{n+1}} \frac{\partial U_{\mathrm{r}}}{\partial t} = \int_0^{\infty}\int_0^{4\pi} \sigma_{\mathrm{a}}^n(x,\nu,T)\int_{t^n}^{t^{n+1}} I(x,\nu,\Omega,t)\ dt\ d\Omega\ d\nu$$

$$-c\sigma_{\mathrm{a,p}}^n(T)\int_{t^n}^{t^{n+1}} U_{\mathrm{r}}(T)\ dt + \int_{t^n}^{t^{n+1}} S_{\mathrm{m}}(x,t)\ dt. \tag{2.12}$$

At this point the material energy balance could be solved for U_r with an integrating factor. Following that route leads to the Carter-Forest method [8] and the adaptive material coupling developed by McClarren [41]. To derive the IMC method, the integration is carried out and time average quantities are substituted:

$$\overline{U_{\mathrm{r}}} = \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} U_{\mathrm{r}}\ dt$$

$$\overline{I} = \frac{1}{\Delta t} \int\limits_{t^n}^{t^{n+1}} I \; dt$$

$$\overline{S_{\mathrm{m}}} = \frac{1}{\Delta t} \int\limits_{t^n}^{t^{n+1}} S_{\mathrm{m}} \; dt$$

$$\frac{1}{\beta^n} \frac{U_{\mathrm{r}}^n - U_{\mathrm{r}}^{n+1}}{\Delta t} = \int\limits_0^{\infty} \int\limits_0^{4\pi} \sigma_{\mathrm{a}}^n(x,\nu,T)\overline{I}(x,\nu,\Omega) \; d\Omega d\nu - c\sigma_{\mathrm{a,p}}^n(x,T)\overline{U_{\mathrm{r}}}(T) + \overline{S_{\mathrm{m}}}(x). \quad (2.13)$$

The IMC method is derived by making the approximation:

$$\overline{U_{\mathrm{r}}} = \alpha U_{\mathrm{r}}^{n+1} + (1-\alpha)U_{\mathrm{r}}^n, \quad (2.14)$$

where α is the implicitness parameter. Eq. (2.14) can be rearranged to:

$$\frac{\overline{U_{\mathrm{r}}} - U_{\mathrm{r}}^n}{\alpha} = U_{\mathrm{r}}^{n+1} - U_{\mathrm{r}}^n. \quad (2.15)$$

This is then substituted into Eq. (2.13):

$$\frac{1}{\beta^n} \frac{1}{\alpha} \frac{\overline{U_{\mathrm{r}}} - U_{\mathrm{r}}^n}{\Delta t} = \int\limits_0^{\infty} \int\limits_0^{4\pi} \sigma_{\mathrm{a}}^n(x,\nu,T)\overline{I}(x,\nu,\Omega) \; d\Omega \; d\nu - c\sigma_{\mathrm{a,p}}^n\overline{U_{\mathrm{r}}} + \overline{S_{\mathrm{m}}}(x). \quad (2.16)$$

The equations are then solved for the $\overline{U_{\mathrm{r}}}$:

$$\overline{U_{\mathrm{r}}} = \left(\frac{\alpha \Delta t \beta^n}{1 + \alpha c \sigma_{\mathrm{a,p}}^n \Delta t \beta^n}\right) \left(\int\limits_0^{\infty} \int\limits_0^{4\pi} \sigma_{\mathrm{a}}^n(x,\nu,T)\overline{I}(x,\nu,\Omega) \; d\Omega \; d\nu + U_{\mathrm{r}}^n + \alpha\beta^n \Delta t \overline{S_{\mathrm{m}}}(x) \right). \quad (2.17)$$

This expression for $\overline{U_r}$ can be simplified with the introduction of the Fleck factor, f:

$$f = \frac{1}{1 + \alpha c \sigma^n_{\mathrm{a,p}} \Delta t \beta^n}, \tag{2.18}$$

$$1 - f = \frac{\alpha c \sigma^n_{\mathrm{a}} \Delta t \beta^n}{1 + \alpha c \sigma^n_{\mathrm{a,p}} \Delta t \beta^n}, \tag{2.19}$$

This simplification yields:

$$\overline{U_{\mathrm{r}}} = \frac{(1-f)}{c\sigma_{\mathrm{a,p}}} \left(\int\limits_0^\infty \int\limits_0^{4\pi} \sigma^n_{\mathrm{a}}(x,\nu,T) \overline{I}(x,\nu,\Omega)\; d\Omega\; d\nu + \overline{S_{\mathrm{m}}}(x) \right) + f U^n_{\mathrm{r}}. \tag{2.20}$$

At this point we consider the radiative energy balance of the TRT equations. To accommodate a Monte Carlo solution procedure, an expression is needed for the radiation energy that is continuous in time and linear in temperature. This requires another approximation, one that has been described before as “dubious” [57]. The time averaged equilibrium radiation energy density, $\overline{U_{\mathrm{r}}}$ is approximated as being the instantaneous value. The same approximation is made for the intensity and the source:

$$\overline{U_{\mathrm{r}}} \approx U_{\mathrm{r}}(t),$$

$$\overline{I} \approx I(t),$$

$$\overline{S_{\mathrm{m}}} \approx S_{\mathrm{m}}(t),$$

These expressions are substituted into Eq. (2.20) to form an expression for the radiation source in the transport equation that is explicit in temperature and continuous in the intensity:

$$U_{\mathrm{r}}(t) = \frac{(1-f)}{c\sigma_{\mathrm{a,p}}}\left(\int_0^{\infty}\int_0^{4\pi} \sigma_{\mathrm{a}}^n(x,\nu,T)I(x,\nu,\Omega,t)\ d\Omega d\nu + S_{\mathrm{m}}(x,t)\right) + fU_{\mathrm{r}}^n. \tag{2.21}$$

The transport equation can now be written with $U_{\mathrm{r}}(t)$, which is the final form of the IMC radiative energy balance:

$$\begin{aligned}\frac{1}{c}\frac{\partial I(x,\nu,\Omega,t)}{\partial t} &+ \Omega\cdot\nabla I(x,\nu,\Omega,t) + \sigma_{\mathrm{a}}^n(x,\nu,T)I(x,\nu,\Omega,t)\\ = \chi c\sigma_{\mathrm{a,p}} fU_r^n + \chi(\nu)(1-f)&\left(\int_0^{\infty}\int_0^{4\pi} \sigma_{\mathrm{a}}^n(x,\nu,T)I(x,\nu,\Omega,t)\ d\Omega d\nu + S_{\mathrm{m}}(x,t)\right)\end{aligned} \tag{2.22}$$

Returning to the material energy balance, the linear, time dependent U_r defined in Eq. (2.21) is substituted into the time dependent material energy balance in Eq. (2.7) for $\overline{U_r}$. This yields:

$$\frac{\partial U_{\mathrm{m}}(x,T,t)}{\partial t} = \int_0^{\infty}\int_0^{4\pi} f\sigma_{\mathrm{a}}(x,\nu,T)I(x,\nu,\Omega,t)\ d\Omega d\nu - fc\sigma_{\mathrm{a,p}}(T)U_{\mathrm{r}}^n + fS_{\mathrm{m}}(x,t). \tag{2.23}$$

The material energy balance equation can now be integrated over a timestep. Again, the opacity is assumed to be constant and held at the beginning of timestep value. The approximation for β is not necessary—that approximation was made to obtain an expression for $\overline{U_{\mathrm{r}}}$. Because we have an explicit emission term the material energy balance can remain in terms of U_{m}. The integration yields the final form of the IMC material energy balance:

$$\frac{U_{\mathrm{m}}^{n+1}(x,T) - U_{\mathrm{m}}^{n}(x,T)}{\Delta t} = \frac{1}{\Delta t} \int\limits_0^{\infty} \int\limits_0^{\infty} \int\limits_0^{4\pi} f \sigma_{\mathrm{a}}^{n}(x,\nu,T) I(x,\nu,\Omega,t) \; d\Omega d\nu dt$$

$$-fc\sigma_{\mathrm{a,p}}(T) U_{\mathrm{r}}^{n} + f\overline{S_{\mathrm{m}}}(x,t). \tag{2.24}$$

The TRT system of equations has now been linearized and made into a lower triangular system. The system is solved by determining the intensity with Monte Carlo transport. The intensity can then be used to solve the material energy balance for U_{m}^{n+1}.

The discretization has an odd feature which can sometimes be a tripping point when implementing the method—the Fleck factor in the IMC equations is derived by assuming that β, and thus the heat capacity, c_{V}, are held at the beginning of timestep values but this approximation is not used when the expression for U_{r} is substituted back in to the material energy balance. If the heat capacity is constant, the temperature at t^{n+1} is obtained by simply dividing the material energy by the heat capacity. If the heat capacity is a function of temperature, the temperature at n^{n+1} must be calculated by inverting the heat capacity integral in Eq. (1.6), which is done by solving for T^{n+1} in:

$$U_{\mathrm{m}}^{n+1} = \int\limits_0^{T^{n+1}} c_{\mathrm{V}}(T) \; dT. \tag{2.25}$$

To solve the time-dependent IMC equations, boundary conditions are needed for the temperature and intensity:

$$I(\mathbf{\Gamma}, \overline{\Omega}, \nu, t) = I_b(\mathbf{\Gamma}, \overline{\Omega}, \nu, t) \tag{2.26}$$

$$I(\mathbf{x},\Omega,\nu,0) = I_0(\mathbf{x},\Omega,\nu) \tag{2.27}$$

$$T(\mathbf{x},\Omega,\nu,0) = T_0(\mathbf{x},\overline{\Omega},\nu) \tag{2.28}$$

Where I_b is the incoming intensity, Γ corresponds to points on the boundary and $\overline{\Omega}$ are the incoming angles ($\Omega \cdot \mathbf{n} < 0$). I_0 and T_0 are the initial intensity and initial material temperature. If the problem begins at equilibrium, the initial intensity is given by:

$$\int_0^\infty I_0(x,\Omega,\nu,T)\ d\nu = \frac{1}{4\pi} acT_0^4 \tag{2.29}$$

Where the spectrum of I_0 is the Planckian spectrum[2] of Eq. (1.1).

2.3 Multigroup Equations

Although the Monte Carlo method allows for a continuous particle frequency, the opacity data is usually given as a constant value for a range of frequencies. This approach of using discrete values over a frequency range is known as *multigroup*. The number of discrete energy groups and their range should resolve the physical behavior of the problem. For example, if a material has a sharp absorption peak at 1.0 keV, their should be enough groups to differentiate that from a high scattering opacity at 0.9 keV. To form the multigroup IMC equations, the radiative energy balance is integrated over each discrete energy range to form a system of coupled equations. The radiative energy balance for group g (with space, angle and time dependence suppressed for clarity) is:

[2] The spectrum of the initial radiation is not the emission spectrum of Eq. (2.4), this can be seen by looking at the steady state, infinite medium solution of the Eq. (1.4)

$$\frac{1}{c}\frac{\partial I_g}{\partial t} + \Omega\cdot\nabla I_g + \sigma^n_{a,g} I_g = \chi(g,T) f \sigma^n_{\mathrm{a,p}} c U^n_{\mathrm{r}} + \frac{\chi(g,T)}{4\pi}\sum_{g'=1}^{G}\int_0^{4\pi}(1-f)\sigma^n_{a,g'} I_{g'}\ d\Omega. \quad (2.30)$$

where I_g is the group integrated intensity for group g, $\sigma_{a,g}$ is the average absorption opacity of group g, and G is the total number of groups. The emission spectrum, χ, becomes a function of the group index, g:

$$\chi(g,T) = \frac{\int_{g-1/2}^{g+1/2} \sigma_a B\ d\nu}{\int_0^{\infty} \sigma_a B\ d\nu} \approx \frac{\sigma_{a,r,g}\int_{g-1/2}^{g+1/2} B\ d\nu}{\int_0^{\infty} \sigma_{a,r} B\ d\nu}. \quad (2.31)$$

Where $\sigma_{a,r,g}$ is the Rosseland weighted opacity for group g.

The equation for group g is coupled to the other discrete groups by the effective scattering term, which is a summation over all groups. The multigroup material energy balance is formed by integrating Eq. (2.24) over each frequency group and summing the result:

$$\frac{U^{n+1}_{\mathrm{m}}(x,T) - U^n_{\mathrm{m}}(x,T)}{\Delta t} = \sum_{g'=1}^{G}\int_0^{4\pi} f\sigma^n_{a,g'} I_{g'}\ d\Omega\ d\nu - f\sigma^n_{\mathrm{a,p}} c U^n_{\mathrm{r}} \quad (2.32)$$

Both the grey and the multigroup TRT equations are formed by integrating over frequency, this introduces an additional approximation which leads to a discussion about how to weight opacities. When integrating over a frequency range, the following approximation is made to the absorption term:

$$\int_{g-1/2}^{g+1/2} \sigma_a(\nu,T) I(x,\nu,\Omega,t)\ d\nu \approx \sigma_{a,g}(T)\int_{g-1/2}^{g+1/2} I(x,\nu,\Omega,t)\ d\nu = \sigma_{a,g}(T) I_g \quad (2.33)$$

An approximation would not be necessary if the opacity could be weighted with the actual intensity:

$$\sigma_{a,g} = \frac{\int\limits_{g-1/2}^{g+1/2} \sigma_a(\nu,T) I(x,\nu,\Omega,t)\ d\nu}{\int\limits_{g-1/2}^{g+1/2} I(x,\nu,\Omega,t)\ d\nu} \tag{2.34}$$

Since the intensity is an unknown, the opacity must be weighted with a different function that approximates the shape of the intensity. *Planck weighting* uses the Planckian to weight the group opacities:

$$\sigma_{a,g,p} = \frac{\int\limits_{g-1/2}^{g+1/2} \sigma_a(\nu,T) B(\nu,T)\ d\nu}{\int\limits_{g-1/2}^{g+1/2} B(\nu,T)\ d\nu} \tag{2.35}$$

At equilibrium, the intensity is Planckian and this weighting is correct. An asymptotic analysis of the equilibrium diffusion limit shows that an alternate weighting known as *Rosseland weighting* is necessary to preserve the equilibrium diffusion limit [34]:

$$\sigma_{a,g,r} = \frac{\int\limits_{g-1/2}^{g+1/2} \frac{\partial B(\nu,T)}{\partial T)}\ d\nu}{\int\limits_{g-1/2}^{g+1/2} \frac{1}{\sigma_{\mathrm{a}}}(\nu,T)\frac{\partial B(\nu,T)}{\partial T}\ d\nu} \tag{2.36}$$

2.4 Properties of the IMC Equations

The IMC method has been shown to be unconditionally stable for all timestep sizes [3] in gray infinite-medium and one-dimensional cases [46] [59]. The IMC method

[3] The implicitness parameter, α, must be greater than 0.5 to guarantee unconditional stability

also maintains equilibrium conditions.

The linearization in IMC leads to a reemission term, which is effectively a scattering term—photons are redistributed in angle and energy after they interact. Effective scattering can be interpreted as a photon reemission. The amount of effective scattering depends on the Fleck factor. The effective scattering has ramifications related to simulation runtime that are discussed below.

2.4.1 Implementation of the IMC Equations

To solve the IMC equations using a computer involves four large steps:

1. Some number of user defined particles are created to represent the energy in the problem. Position in a cell, angle, frequency and emission time are all determined with random numbers.

2. Each particle history is simulated: particles move at the speed of light and the probability of a scattering interaction is determined with random numbers. To improve the quality of the solution, the implicit capture method is used, which treats the particle as continuously being absorbed into the material as it moves between events (see Section 4). Particles with energy below a user defined cutoff value are treated with a rouletting procedure. The life of a particle is further detailed in Fig. (2.1).

3. The photons that reached the end of the timestep without leaving the problem or being rouletted are called *census particles*. The emission time is set to the beginning of the next timestep and these census particles form the initial radiation condition of the next timestep.

4. The material energy is updated using the absorbed energy from the Monte Carlo transport step. The temperature is determined with Eq. (2.25) and the

emission energy, opacity, and Fleck factor are all updated and the timestep is incremented.

A key point of the IMC method is that the frequency of a particle is not the energy of that particle. Instead, a particle can be thought of as a group of photons all with frequency ν' and a total energy of E'. To clarify this relationship, the energy represented by an IMC particle is usually referred to as its *energy weight* and a simulated group of photons is called a particle.

2.4.2 Computational Cost

In Monte Carlo methods for radiation transport the radiation intensity is in essence continuous in both space and angle (a continuous representation is not stored, but is used to update a spatially discrete temperature). This property has lead to IMC being considered a standard solution of the TRT equations [6]. This continuous representation of the radiation intensity comes with a very large caveat—the solution is only a continuous representation, and thus the correct solution for the radiation intensity, in the limit of an infinite number of simulated particles. The computational cost of IMC is generally proportional to the number of simulated particles used to obtain the solution, although the cost of simulating a particle is dependent on the local dependent variables. Because the solution is determined with pseudorandom numbers, the solution also exhibits variance. The variance scales inversely as the square root of the number of particles, meaning a 100 fold increase in simulated particles yields a 10 fold reduction in error. The IMC method is thus limited because it uses a finite number of particles to represent seven dimensional phase space.

In addition to the cost of accurately sampling phase space and reducing variance in the solution, (features of all Monte Carlo methods) the IMC method has several additional computational challenges. The Fleck factor introduces "effective scatter-

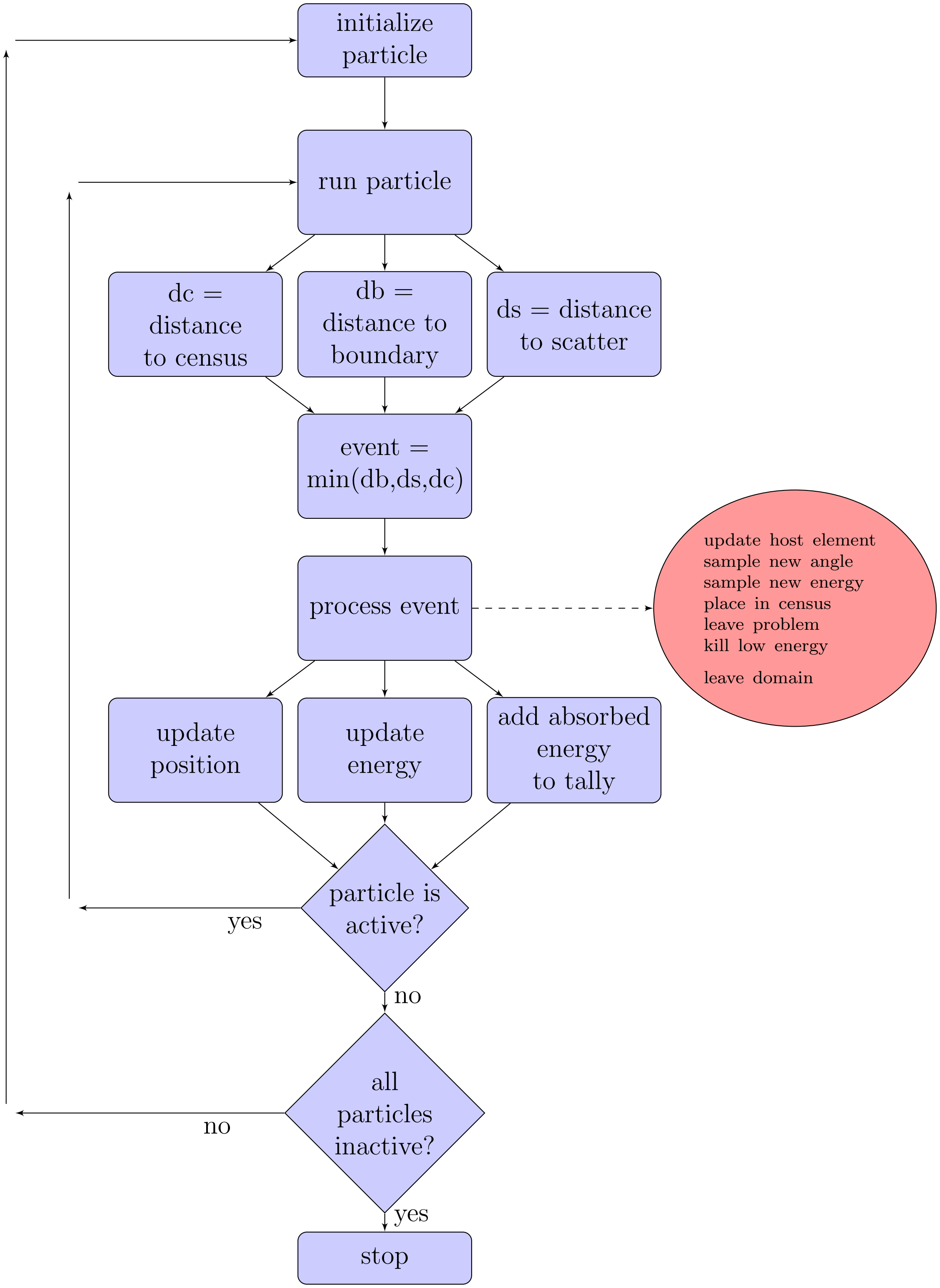


Figure 2.1: The main loop over IMC particles, similar to chart in CCS-2:12-55(U)

ing" to the TRT radiative energy balance. In areas where the effective scattering ratio is high, particles can have many scattering events before reaching the end of the timestep. There are several ways of dealing with this cost, all of them involve treating the particle with diffusion theory in areas where effective scattering is large [21] [15] [23] [11]. Another computational challenge in thermal radiative transfer problems is the size of the discrete mesh. Unlike source-detector Monte Carlo simulations, TRT problems are time-dependent and require capturing spatial gradients. This requires a mesh that can resolve the features of the solution. This kind of mesh is not usually necessary in source-detector problems common in neutronics and gamma-ray simulations. In those problems, large combinatorial geometries are used. The need for a mesh that can resolve features of the solution has two consequences for Monte Carlo TRT simulations:

- Memory access is more likely to be indirect
- Multiple computer nodes are needed to represent large meshes

The first consequence is a difficulty because IMC is *memory intensive.* This means that a significant amount of time during simulation is spent loading data instead of performing arithmetic operations on that data. This problem is compounded by the fact that particles do not access data in regular patterns, as is common in hydrodynamics, S$_N$ transport and diffusion algorithms. In IMC, a particle can stream across the entire mesh if the timestep is large and the problem is optically thin. The degree to which IMC is memory bound is a subject of this work.

The second item requires some elucidation on the idea that Monte Carlo transport is "embarrassingly parallel." This oft-spoken shibboleth comes from the fact that Monte Carlo particle histories are independent (this is true for particles within a timestep for IMC and for all particles in steady-state neutronics problems), which allows computers operating in parallel to divide the particles between them. This kind of decomposition only works when the spatial domain and physical data can be stored locally by each parallel process, this mode of operation is called *domain replication*. If the mesh and physical data are too large to be replicated on each parallel process, the mesh (usually larger than the physical data) must be decomposed into smaller components that are stored on each process. This mode of parallel execution is called *domain decomposition*. In domain decomposed execution, parallel processes must communicate with other processors in order to compete the simulation. This communication significantly increases runtime and adds complexity to a Monte Carlo simulation, thus breaking the “embarrassingly parallel” norm. The runtime and complexity associated with domain decomposition is addressed in Section 6 of this work.

2.5 Current State of IMC Research and Monte Carlo TRT

Many improvements have been made to the IMC method over the past 40 years. IMC has been shown to hold a “diffusion-like” equation in the equilibrium diffusion limit of its temporal discretization [14] and the incorrect diffusion limit with its spatial discretization, which can be corrected by properly accounting for the spatial shape of the emission [13] [52]. Much work has been done to improve the computational performance of diffusive regions, starting with the “random-walk” method of Fleck and Canfield [21] and then later Gentile and Densmore used a diffusion discretization in thick regions [23] [15] [11]. Several other Monte Carlo radiation transport codes have sought to correct both the linearatization and discretization deficiencies of the IMC method. The Symbolic Implicit Monte Carlo Method (SIMC) of Nkoua [47] and Brooks [3] does not involve a linearization of the emission term, instead a matrix is formed and solved that relates the emission of one cell to the absorption in every other cell. This method involves the solution of an $N \times N$ matrix, where N is the number of unknowns. This matrix is positive and could be dense if the problem is optically thin. The SIMC method still uses the implicit Euler approximation in time, making it temporally first order. The Carter-Forest method [8] applies a linearization similar to IMC but the equations are integrated over a timestep instead of being temporally discretized.[4]

The IMC method is still considered the standard for Monte Carlo TRT simulations. Although the SIMC method is implicit and would thus avoid maximum principle violations, the computational and memory cost of SIMC have kept it from being widely implemented. The Carter-Forest has smaller magnitude discretization errors but it introduces time-dependent emission and opacity. This makes the implementation more complex than IMC. The Carter-Forest still has an $\mathrm{O}\left(\Delta t\right)$ error due to the approximation that the opacity and β are constant throughout a timestep [10]. For these reasons and simple inertia, the IMC method is still used as the standard method for solving the TRT equations with a Monte Carlo method.

[4] The linearizations and temporal discretizations of IMC, Carter Forest and SIMC are examined in the equilibrium diffusion limit by Densmore [14]

3. ADVANCED SPATIAL DISCRETIZATION

3.1 Applying a Spatial Discretization to the IMC Equations

The IMC equations in Eq. (2.22) and Eq. (2.24) have not yet been discretized in space. The IMC radiative energy balance in Eq. (2.22) has been discretized in time and additionally, frequency, in Eq. (2.30). Because Monte Carlo particles can take on any time within the timestep, the intensity, I, in the radiative energy balance and material energy balance is continuous in time in the limit of an infinite number of particles. The Monte Carlo particles can also take on any value in space and angle meaning that the intensity I is also continuous in space and angle in the limit of an infinite number of particles and does not need to be discretized. The continuous nature of the intensity is the reason that Monte Carlo is considered a standard solution to the transport equation—it does not suffer from the ray effects of the S_N method or the wave effects of the P_N method [7]. All other dependent variables in the IMC equations need to be discretized in space in order to solve the IMC equations. This is done by imposing a spatial grid or mesh on the problem. Currently, the IMC method uses piecewise-constant unknowns, meaning there is one temperature, opacity and Fleck factor value for each cell on the mesh.

There are several consequences of using piecewise-constant unknowns:

- The error term is first order in space: $\epsilon = \mathrm{O}\big(\Delta x\big)$
- Because the variance in quantities of interest (e.g. temperature, momentum) is related to the number of tally events, increasing the number of mesh cells reduces the number of tally events in a given cell if the number of particles is not increased

- Particle absorption is continuous, but the emission is based on discrete, piecewise-constant, dependent variables. Thus absorption can occur at one side of a cell and on the next timestep emission occurs uniformly in the cell. This means energy propagates too quickly leading to an error sometimes referred to as *teleportation error*, but this diffusive behavior could also just be called spatial discretization error.

To address the first two errors, the spatial resolution and the number of particles can be increased. The third error can be improved by imposing a spatial shape on to the emission. This is done by using additional information from the simulation to form the spatial shape. Originally it was done by using the temperature of adjacent cells to form a linear function [21]. This procedure is often called *source tilting* [42]. Other forms of tilting use functional tallies [10] or particle histories [30]. Densmore has shown that tilting improves the behavior of IMC in the diffusion limit [13]. These tilting methods all serve in reducing the spatial discretization error of IMC without increasing the number of cells.

In coupled multiphysics simulations, the discrete unknowns must be mapped between different physics algorithms. The simplest way to do this is to collocate the unknowns between different physics simulations. For HEDP and astrophysics problems, TRT simulations are most often coupled to hydrodynamics, this is commonly referred to as *radiation hydrodynamics*. The equations of radiation hydrodynamics are the Euler equations with additional sources:

$$\frac{\partial}{\partial t}\rho = \nabla \cdot (\rho \mathbf{u}) \tag{3.1}$$

$$\frac{\partial}{\partial t}(\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \nabla p = -\mathbf{S_{rp}} \tag{3.2}$$

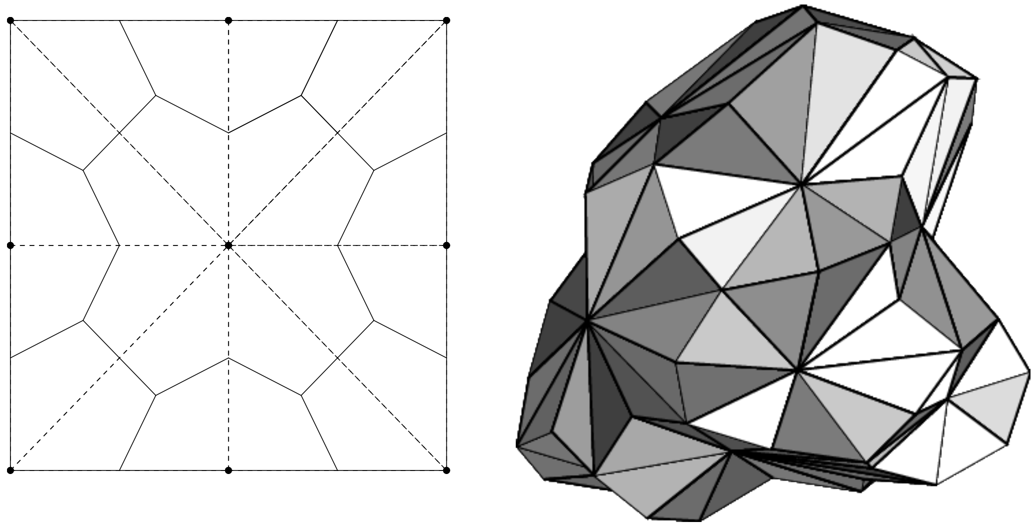
$$\frac{\partial}{\partial t}\left(\frac{1}{2}\rho u^2 + \rho e\right) + \nabla \cdot \left(\left(\frac{1}{2}\rho u^2 + \rho e + p\right)\mathbf{u}\right) = -S_{re} \tag{3.3}$$

The radiation source to the momentum equation, S_{rp}, and the radiation source to the energy equation, $\mathbf{S_{re}}$, are defined by Pomraning [50]. This work does not explore radiation hydrodynamics explicitly, instead the IMC is examined in the context of using the same mesh and spatial discretization as modern hydrodynamics so that it can be effectively coupled to those hydrodynamics methods.

Currently, all of the Monte Carlo methods for TRT are first order accurate in space and make no attempt to use a high-order spatial representation of the temperature. Work has been done exploring a discontinuous finite element discretization with IMC in idealized conditions as a means of reducing the diffusive teleportation error [60].

Modern algorithms for other physics, especially hydrodynamics, have embraced high-order spatial discretizations [17] [9] [56]. High-order methods are viewed as a path forward for modern architectures because they involve more computational work per unknown [18]. This reduces the number of memory operations per floating point operation. The use of high-order spatial discretization in hydrodynamics means that there is a disparity between the spatial accuracy of Monte Carlo radiation transport methods and modern methods for hydrodynamics. How to treat a high-order spatial discretization of IMC is a focal point of this work. We examine how IMC can best be coupled to a hydrodynamics code called CHICOMA, which uses a continuous linear finite-element discretization to solve the ALE form of the Euler equations [56].

The hydrodynamics method in CHICOMA has node-centered unknowns, meaning that the control volumes is represented by the dual mesh. This presents a problem for IMC methods—the unknowns in standard IMC implementations are at cell-centers. The CHICOMA code uses an unstructured tetrahedral mesh. The control volumes formed by the dual mesh are arbitrarily sided polygons and are generally not convex.[1] The dual mesh of a triangular 2D mesh is shown in Fig. (3.1a). An example of dual mesh cell formed by an unstructured tetrahedral mesh is shown in Fig. (3.1b).



(a) Example of the dual mesh (solid line) formed by a 2D triangular mesh

(b) A dual-mesh cell formed by an unstructured tetrahedral mesh

Figure 3.1: Dual mesh cells in 2D and 3D (reprinted with permission from [37])

The IMC method requires tracking particles as they move through, and interact with the background material. The interaction probability is defined by the opacity, which is constant within a mesh cell in the standard IMC method. Tracking a particle as it streams through a polyhedral mesh is computationally expensive because the intersection of each surface with the particle must be calculated to determine which surface is crossed. Sampling particle emission locations on this mesh would also be very computationally expensive: particles would have to be sampled uniformly on a large encompassing volume and then rejected if they were outside the dual mesh

[1] Concave mesh cells are not a problem in Monte Carlo transport, in S_N transport they introduce an additional dependency in the sweep algorithm—the intensity in the cell depends upon itself

cell. Sampling emission locations based on a temperature field would require an additional rejection step.

In this work we examine three IMC variations for unstructured tetrahedral meshes that avoid the difficulty of tracking and sampling directly on the dual mesh. Two of them use piecewise- constant data and the third attempts to use the linear temperature field of the CHICOMA hydrodynamics method:

- A node-centered IMC method where all unknowns are located at the nodes and the unknowns are treated as piecewise-constant
- A corner-centered IMC method where unknowns are defined on the corners of the tetrahedra and the node centers are updated using corner values
- A linear discontinuous finite element method for the material energy that treats emission, opacity and the scattering opacity as piecewise-linear

3.2 Node-Centered IMC Variant

Instead of implementing the algorithm on the dual mesh, the energy is decomposed on to the *corners* of the tetrahedra that make up the dual cells (corners are also used as a sub-cell in S_N methods for arbitrary polyhedra [2]). As an example, for the 2D mesh in Fig. (3.1a), the dual mesh cell in the center of contains 8 corners. Decomposing the energy like this allows particle tracking to be done on the tetrahedral mesh, where there are always four potential surfaces to cross instead of an unknown number. Sampling uniform emission locations on the tetrahedral mesh is much simpler, as is sampling emission locations using a temperature field. Efficient methods for both uniform and temperature field sampling are given in Section 4.

The unknowns in the node-centered method are shown for an analogous 2D mesh in Fig. (3.2). One feature of this method that is unique to the other discretizations considered in this work is that spatial cells on the boundary can have the same volume and different surface area. This is the case for cells T_i and T_k in Fig. (3.2)—they are both formed from the dual mesh of equally sized triangular cells, but T_k has twice the surface area of T_i.

Materials are defined on the tetrahedra. This means that a dual-mesh cell can contain multiple materials. This is handled by storing a unique temperature for every material at the node, which means there are multiple control volumes in within the dual mesh cell and the temperature is discontinuous at material interfaces. This is shown for an analogous 2D triangular mesh in Fig. (3.3).

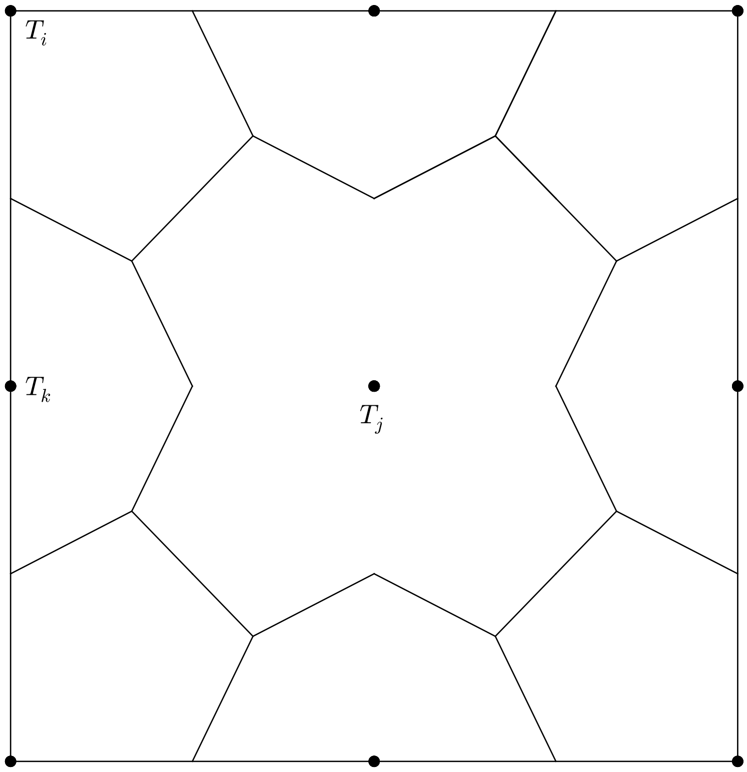


Figure 3.2: Locations of the unknowns in the node-centered IMC method for a 2D triangular mesh

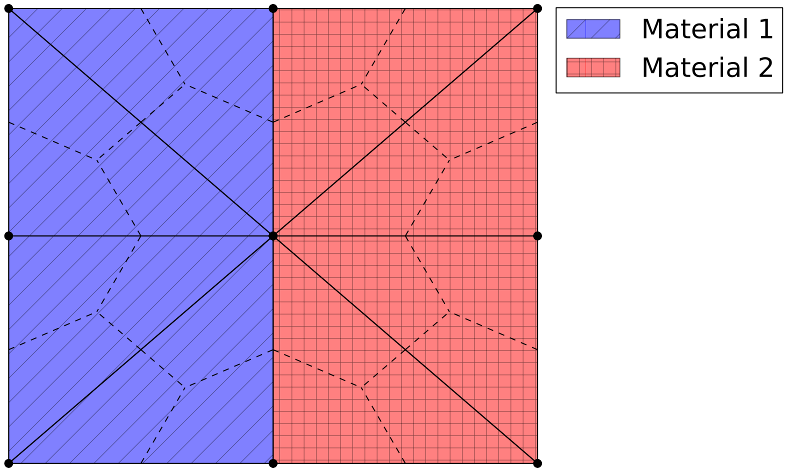


Figure 3.3: Two materials in a 2D triangular mesh with a material interface within the center dual mesh cell (reprinted with permission from [37])

The shift from implementing IMC directly on the dual volumes to tetrahedra and corners presents two drawbacks:

- Decomposing energy equally between the corners is akin to assuming that the emission is uniform within a cell. This limits the efficacy of tilting algorithms. This problem is addressed with a method called *emission upwinding*.
- An additional event must be accounted for in particle tracking: dual volume crossing within tetrahedra. This is necessary because a tetrahedra contains four different nodes and potentially four different opacities and Fleck factors. This adds computational work.

The second drawback mentioned above related to decomposing the energy on the tetrahedra adds a large amount of computational work. During the simulation, a particle determines its next event by calculating the distance to each potential event and then selecting the event with the smallest distance, or in other words, the next temporal event. Transporting the particle on the tetrahedra instead of the dual mesh requires tracking to the dual interface. This adds another event into the basic transport loop. The hydrodynamics algorithm used in the CHICOMA code uses a continuous linear finite element spatial discretization. If IMC could also use this spatial discretization, the opacity could be treated as linearly continuous within a tetrahedra. This would eliminate the need to track a particle to the interface of a dual mesh cell. Instead, a continuous opacity could be used to determine the distance to scatter and energy would be absorbed according to the linear basis functions. This is another potential benefit to finite element methods in IMC.

Uniformly dividing the energy between corners of the dual mesh cell essentially eliminates the benefits of tilting. Tilting can still be used after energy is decomposed onto the corners, but this only affects the emission shape within a corner. To show the effects of energy decomposition, we examine emission in the center cell of Fig. (3.2), which is centered at $(0.5\text{ cm}, 0.5\text{ cm})$. We look at the case of emitted particle locations in the presence of a linear temperature field that varies in x with $T(0) = 1.0$ and $T(1.0) = 0.01$. The specifics of how to generate samples with a PDF defined by a linear field are addressed in Section 4. As the reference case, we sample directly on the dual mesh cell and use rejection sampling based on the temperature field. Ideally, this is what would be done in the IMC routine but this is very expensive when there are an indeterminate number of tetrahedra that make up a dual mesh cell. The results for the reference case are shown in Fig. (3.4). If energy is uniformly divided between the corners, the emission is uniform over each corner and thus the whole dual mesh cell, as seen in Fig. (3.5). If the linear temperature field is used to bias emission after the energy is divided, the average particle location does not change significantly, as Fig. (3.6) shows. The average particle locations are reported in Table (3.1).

To effectively sample emission particles based on the temperature field and regain the benefits of tilting, a method for biasing the amount of energy decomposed on to each corner is needed. To do this we assume that the shape of the emission is piecewise-linear. The emission within a tetrahedra can then be written as a function of the emission at the nodes using barycentric coordinates:

$$E(\lambda) = \sum_{i=1}^{4} E_i \lambda_i \tag{3.4}$$

Where λ_i is the barycentric coordinate for node i and E_i is the emission at node i (the nodes of the tetrahedra are centers of dual-mesh cells). To determine the amount energy that should be emitted in a given corner of a dual mesh cell, we integrate Eq. (3.4) over a corner of the tetrahedra, which yields:

$$\overline{E}_c = \int\limits_{\mathbf{x}_c} \sum_{i=1}^{4} E_i \lambda_i = \frac{1}{4} V_t \left(\frac{75}{144} E_c + \sum_{i=1, i \neq c}^{4} \frac{23}{144} E_i \right) \tag{3.5}$$

Where c is the corner that belongs to the dual mesh cell of interest, $\mathbf{x}_c$ is the spatial domain of the corner, V_t is the volume of the tetrahedron and $\overline{E}_c$ is the integral of the linear emission function over the corner c. The fraction of emission energy that corner c receives is:

$$\tilde{E}_c = \frac{\overline{E_c}}{\sum\limits_{i=1}^{N} \overline{E_i}} \tag{3.6}$$

Where N is the total number of corners in the dual mesh cell, $\overline{E_i}$ is the integral, Eq. (3.5), over the corner i, and $\tilde{E}_c$ is the fraction of the total emission in the dual-mesh cell that corner c receives. We now have a method of decomposing the

Method	Average Particle Position (cm)	
	x	y
Dual Sampling with Tilting	0.445	0.500
Uniformly Decomposed	0.500	0.500
Uniformly Decomposed with Tilting	0.490	0.500
Emission Upwinding Decomposition	0.454	0.500
Emission Upwinding with Tilting	0.445	0.500

Table 3.1: Average Particle positions in the dual mesh cell centered at $(0.5, 0.5)$, with 5.0×10^5 points for various methods

energy onto the corners that takes into account the shape of the emission field. We call this method *emission upwinding* because it biases emission in the direction of an energy gradient. Emission upwinding does not advect energy across dual-mesh cell boundaries, it only changes how energy is decomposed onto the corners of a dual-mesh cell.

To test emission upwinding on a triangular mesh, we integrate linear emission over a corner of a triangle, which yields:

$$\overline{E}_c = \int\limits_{\mathbf{x}_c} \sum_{i=1}^{3} E_i \lambda_i = \frac{1}{3} A_t \left(\frac{22}{36} E_c + \sum_{i=1, i \neq c}^{3} \frac{7}{36} E_i \right), \tag{3.7}$$

where A_t is the area of the triangle. Equation (3.6) is then used in the same manner to determine the amount of energy each corner receives. Particle locations with emission upwinding for a triangular dual mesh cell are shown in Fig. (3.7). Using tilting with emission upwinding is shown in Fig. (3.8). Emission upwinding with tilt gives an average particle location that is within 0.1% of the reference solution, as shown in Table (3.1). Thus emission upwinding proves to be an effective way of decomposing energy and when it is combined with tilting it appears to produce the same effect as sampling with tilting directly on the dual mesh cell.

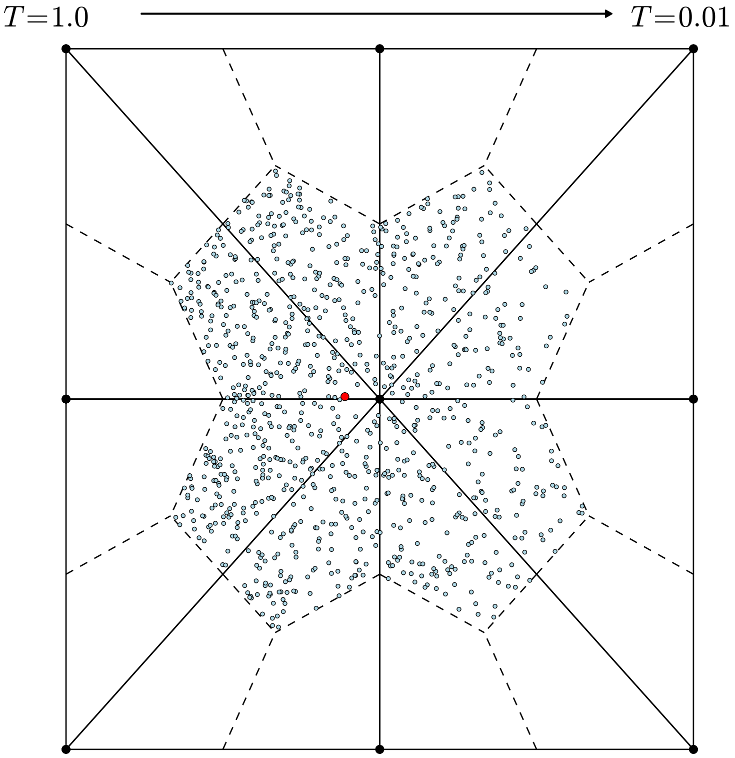


Figure 3.4: One thousand sampled emission particle locations using a linear temperature field on the dual mesh cell, the average location is the large red point

Another concern with the node-centered method is that individual dual-mesh cells on a boundary can have different surface area to volume ratios. This presents problems when a boundary source is present and the problem is optically thick—in this case all energy absorption takes place on the boundary surface and there is very little energy movement between adjacent cells because the problem is optically thick. Because the surface areas to volume ratios are different and the energy absorbed is proportional the surface area, two mesh cell that are normal the boundary source will have two different material temperatures. This creates an asymmetry in the solution. This asymmetry is expected in optically thick Marshak wave problems. Although

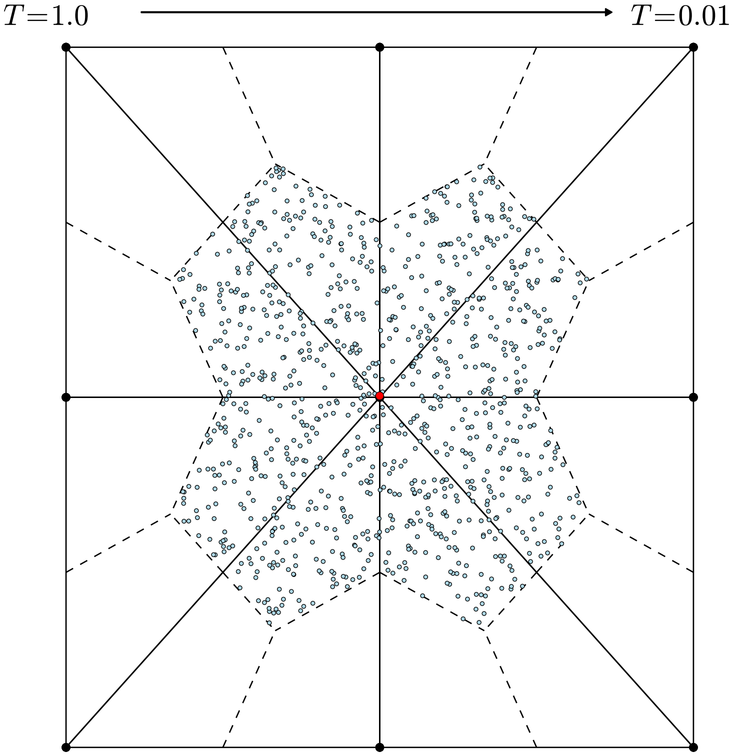


Figure 3.5: One thousand sampled emission particle locations after energy is uniformly decomposed from the dual mesh cell on to the corners, the average location is the large red point

Marshak wave problems are generally simulated in 1D, the CHICOMA code uses a 3D mesh[2], thus an asymmetry is expected in the solution until the problem heats up enough to allow radiation to flow between mesh cells.

3.3 Corner-Centered IMC Variant

The second method is based on using corners as control volumes. A corner of a tetrahedron is formed by drawing a line between the barycenter of the tetrahedron

[2] The CHICOMA code was designed from the start to be 3D. The idea being that is is easier for a method to prove its utility in 3D and then move down in dimensionality than to prove its utility in 1D and then move up.

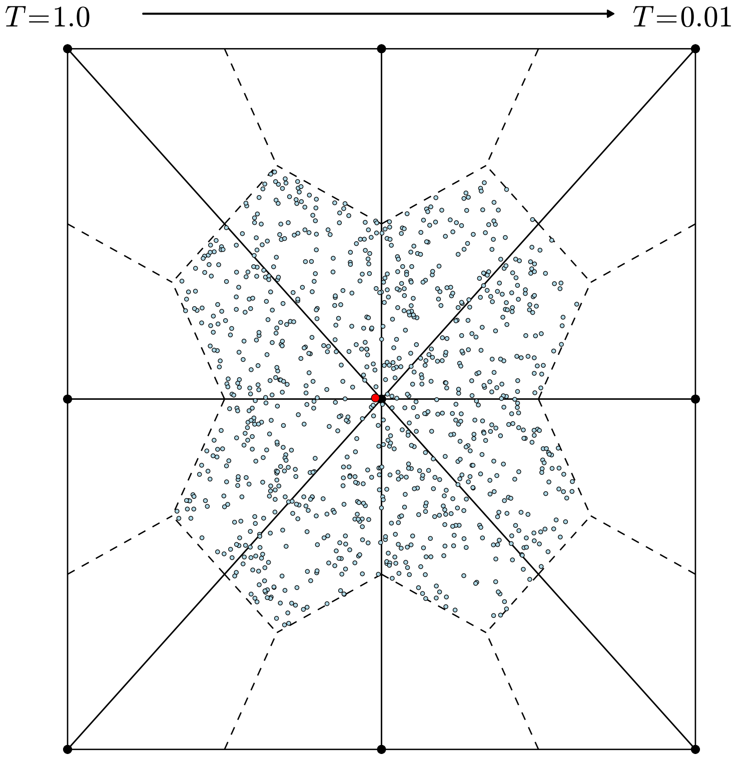


Figure 3.6: One thousand sampled emission particle locations given energy is uniformly decomposed from the dual mesh cell on to the corners and then biased with the linear temperature field, the average location is the large red point

and the barycenter of each of the four faces. This divides a tetrahedron into four equal volume corners. A corner within a tetrahedron is shown in Fig. (3.9). Corners are used as the control volumes in S_N radiative transfer in the corner balance method [2]. Using corners as a control volume defines multiple, independent temperatures within a larger dual mesh cell—this approach does not have the diffusive behavior that comes from dividing the energy of the node onto the corners. The corner-centered method applied to a 2D triangular dual mesh is shown in Fig. (3.10). In Fig. (3.10), T_k is the temperature of a corner that belong to the dual-mesh cell $\tilde{T}_n$. T_k is used to update

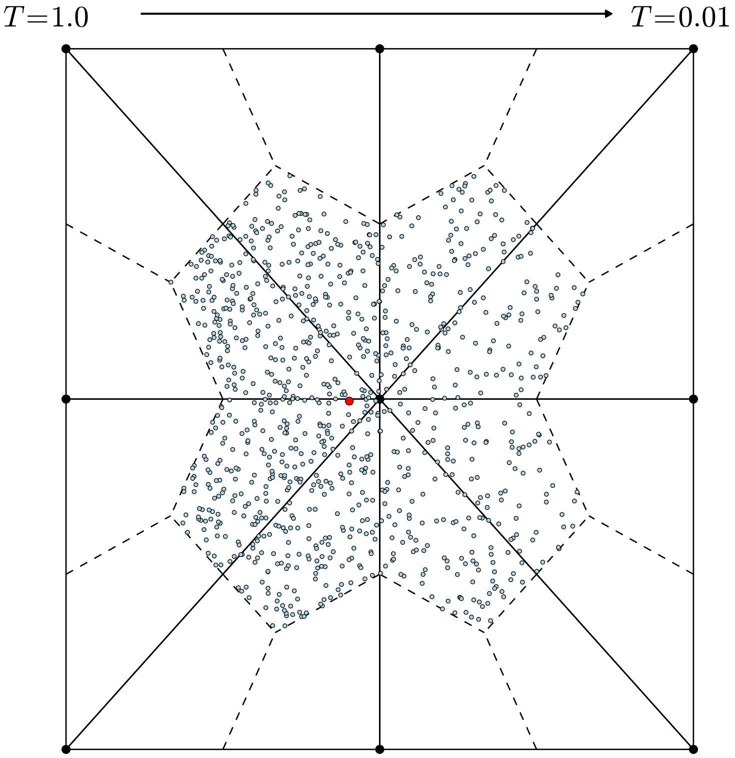


Figure 3.7: One thousand sampled emission particle locations given emission upwinding is used to decompose the energy on to the corners, the average location is the large red point

$\tilde{T}_n$ at the end of the IMC timestep. Emission upwinding is not necessary in the corner-centered IMC method, instead the corner temperatures naturally determine the amount of emission energy in each corner. Within the IMC method, the node temperatures are only used to construct an emission shape, which is used to tilt within corners.

The corner-centered method has a higher memory cost than the node-centered method because each corner holds an opacity, Fleck factor and temperature value. This results in about eight times more memory for storing unknowns than the node-

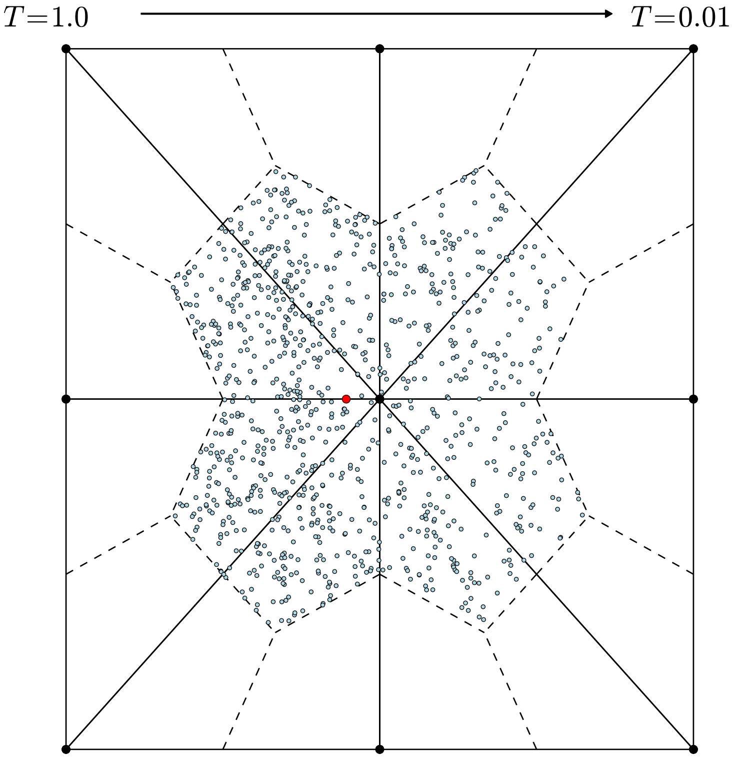


Figure 3.8: One thousand sampled emission particle locations given emission upwinding is used to decompose the energy on to the corners and then biased with the linear temperature field, the average location is the large red point

centered method. The memory footprint for unknown storage is generally much smaller than storing the Monte Carlo census particles so the additional memory required by the corner-centered method is generally not prohibitive. The node-centered method and the corner-centered both track particles in the same manner—in both cases particles must be tracked to the interface of a corner within a tetrahedra. In the node-centered method a corner boundary is an interface between dual mesh cells.

Using the node-centered temperatures in the emission tilt provides a continuous

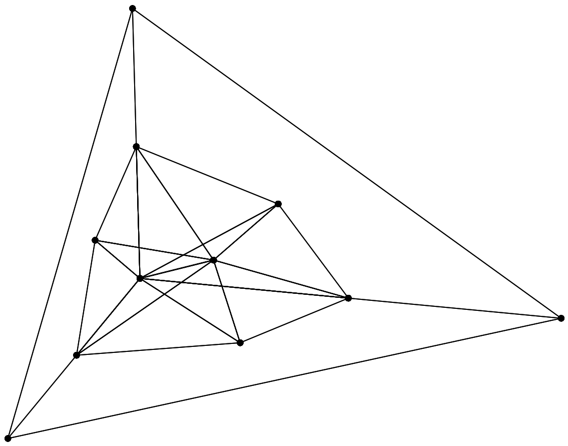


Figure 3.9: A corner control volume within a tetrahedra

emission shape between corners. This is in contrast to the piecewise-constant IMC method in a Cartesian mesh. With a Cartesian mesh, the emission shape is obtained from two adjacent temperatures for each dimension within the cell. This yields a continuous emission shape within the cell but it is discontinuous between cells. Smedley-Stevenson and McClarren have identified a continuous emission shape as having the appropriate behavior in the equilibrium diffusion limit [52].

The corner-based method has not been used in a hydrodynamics simulation with CHICOMA, but the coupling is not as straightforward as the node-centered IMC method. A hydrodynamics code will move energy and mass and change the temperature at the nodes. The temperatures at the corners must be updated in a manner that conserves the new energy in the dual mesh cell and preserves the relative shape of the temperature. This can be done by changing the corner temperatures by the same relative amount as the node temperature was changed by the hydrodynamics step. This method of updating the corner temperatures after the hydrodyanmics

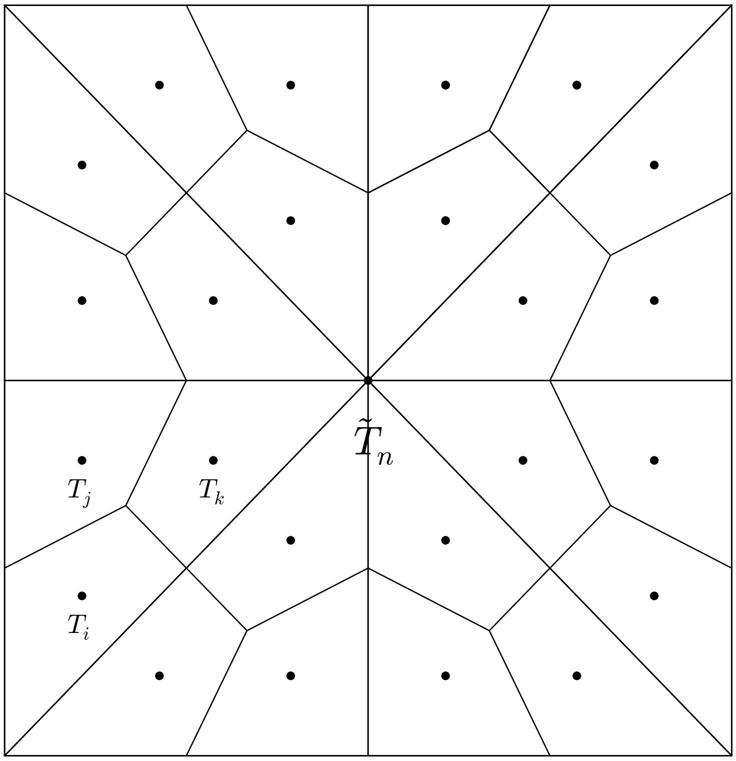


Figure 3.10: Locations of the unknowns in the corner-centered IMC method for a 2D triangular mesh

step is used by other corner-based TRT methods.

In the corner-centered method, the surface area to volume ratio is constant for cells on the boundary. This means that the the temperature is the same for each corner on the boundary after the first timestep. Unfortunately, when the problem is optically thick, energy can only travel through corners that are hot and thus optically thin. This means that a corner must be adjacent to a hot, thin corner to receive energy. Because corners are arranged in a 3D configuration, the solution is not 1D, as it should be. This can be seen by comparing the wavefront in an optically thick Marhsak wave problem and an optically thin Su-Olson problem, as shown in the results below.

3.4 Finite Element IMC

The Finite Element method was first used to solve the neutron transport equation by Reed and Hill in the early 1970's [51]. Finite Element methods were applied to the S_N form of the radiative transfer equations by Larsen and Morel in the 1980's and by Adams in the early 1990's [1].

In the finite element method, unknowns are projected on to a trial function space. Discontinuous finite elements are used for the unknowns. Using discontinuous finite elements is the standard method for finite element methods in TRT. Although the hydrodynamics in CHICOMA uses a continuous finite element method (CFEM) unknown space, the discontinuous finite element method (DFEM) solution in the TRT could be updated after the hydrodynamics step in a similar manner as the corner-centered IMC method described above. The DFEM method is often used to transform spatial derivatives in partial differential equations (PDE) into a set of 3algebraic equations that can be solved with a linear solver. In the IMC method, the Monte Carlo method solves the radiative energy balance equation accurately, including the spatial derivative. Thus we do not use the finite element method to solve the radiation energy balance equation. In applying the finite element method to IMC, we are more interested in representing the material energy with linear trial functions as a means of reducing diffusivity. To form the DFEM IMC equations we first multiply the material energy balance by a test function and integrate over the spatial domain:

$$\int\limits_{\mathbf{X}} \frac{dU^u_{\mathrm{m}}(x)N^v(x)}{dt}\ dx = \int\limits_0^{\infty}\int\limits_0^{4\pi}\int\limits_{\mathbf{X}} N^v(x) f\sigma_a^n I(x,\Omega,\nu)\ dx\ d\Omega\ d\nu - \int\limits_{\mathbf{x}} N^v(x)E(x)\ dx. \tag{3.8}$$

Where N^v are the test functions in the space of V and U^u_{m} are the trial functions in the space of U. Trial functions that satisfy Eq. (3.8) yield the weak form of the material energy balance. In the Galerkin method, the test and trial functions are in the same space. In this work, we assume that the the test and trial spaces are linear. To spatially discretize Eq. (3.8), we represent the domain with some number of linear functions, $U_{\mathrm{m}} \in V$ and $N^v \in V$. Each U_{m} must satisfy Eq. (3.8) for each N^v. This is done by defining N^v in a spatial cell i such that:

$$N_i^v = \begin{cases} N_i^v(x) & \text{if } x \in \mathbf{x}_i \\ 0 & \text{otherwise} \end{cases} \tag{3.9}$$

The linear functions, U^u_{m}, representing the solution contain a coefficient multiplied by a linear function defined in a spatial cell i:

$$U^u_{\mathrm{m},i}(x) = U_{\mathrm{m},i} N^u(x). \tag{3.10}$$

The linear trial functions N^u have the same properties as the test functions:

$$N_i^u = \begin{cases} N_i^u(x) & \text{if } x \in \mathbf{x}_i \\ 0 & \text{otherwise} \end{cases} \tag{3.11}$$

To satisfy the weak form, each trial function for an unknown at node i must satisfy Eq. (3.8) for each test function N^v, because the test and trial functions have been defined to only have extent within the cell, the inner produce of N_i^v and N_j^u is nonzero only within a spatial cell containing node i and node j. The locations of the unknowns in DFEM IMC for an unstructured 2D triangular mesh, are shown in Fig. (3.11). Using a tetrahedral mesh means that a material balance is formed for each N_i^v with integrals from all four material energies in the cell. This forms a local 4×4 matrix, which can be inverted to find $U_{\mathrm{m},i}$ for each node i in the tetrahedron:

$$\sum_{i=1}^{4}\sum_{j=1}^{4}\frac{\partial U_{\mathrm{m},j}}{\partial t}\int\limits_{\mathbf{X_c}} N_j^u(x)N_i^v(x)\ dx = \sum_{i=1}^{4}\int\limits_{0}^{\infty}\int\limits_{0}^{4\pi}\int\limits_{\mathbf{X}} N_i^v(x)f\sigma_a^n I(x,\Omega,\nu)\ dx\ d\Omega\ d\nu$$

$$-\int\limits_{\mathbf{x}} N_i^v(x)E(x)\ dx. \qquad (3.12)$$

To take advantage of the linear material energy density, we assume that the emission, E, absorption opacity, σ_{a}, and effective scattering opacity $f\sigma_{\mathrm{a}}$ are also linear within a tetrahedral cell:

$$\sigma_a(\mathbf{x}) = \sum_{i=1}^{k} N^i(\mathbf{x})f_i\sigma_{a,i}, \qquad (3.13)$$

$$\sigma_s = (1-f)\sigma_a(\mathbf{x}) = \sum_{i=1}^{k} N^i(\mathbf{x})(1-f_i)\sigma_{a,i}, \qquad (3.14)$$

$$E(\mathbf{x}) = c\sum_{i=1}^{k} N^i(\mathbf{x})f_i\sigma_{a,i}U_{r,i}, \qquad (3.15)$$

Because the emission is also linear within a cell, integrating it against the test function N^v yields the same mass matrix as N^u. Carrying out the integration yields the DFEM IMC material energy balance:

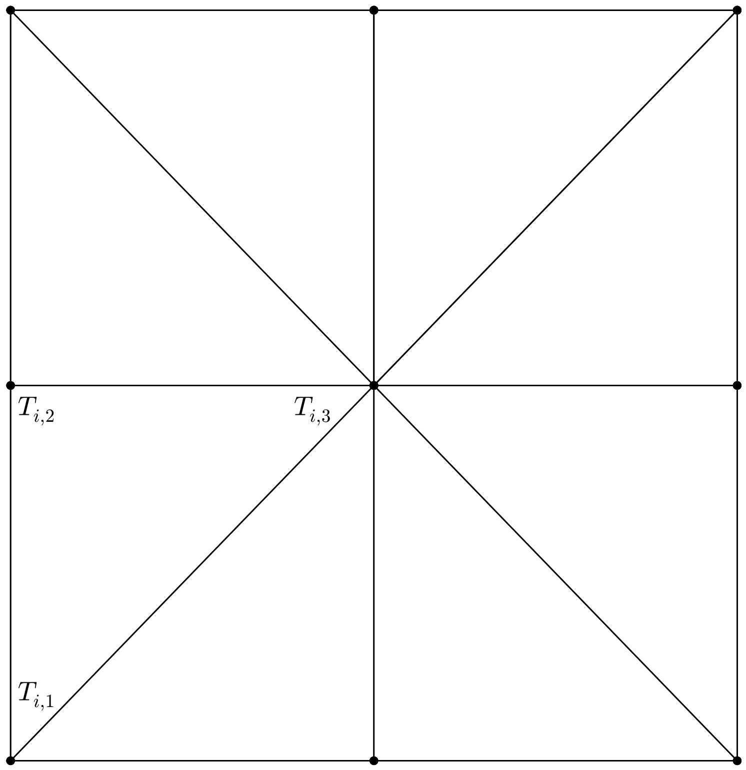


Figure 3.11: Locations of the unknowns in the DFEM IMC method for a 2D triangular mesh

$$\mathbf{M}_{\mathrm{t}} = \frac{\partial \mathbf{U}_{\mathrm{m}}}{\partial t} = \sum_{i=1}^{4} \int_{0}^{\infty} \int_{0}^{4\pi} \int_{\mathbf{X}} N_i^v(x) f \sigma_a^n I(x, \Omega, \nu)\ dx\ d\Omega\ d\nu - M_t \mathbf{E}\ dx. \tag{3.16}$$

Where $\mathbf{M}_{\mathrm{t}}$ is the mass matrix for for a linear tetrahedron:

$$\mathbf{M}_{\mathrm{t}} = V_{\mathrm{t}} \begin{pmatrix} 1/10 & 1/20 & 1/20 & 1/20 \\ 1/20 & 1/10 & 1/20 & 1/20 \\ 1/20 & 1/20 & 1/10 & 1/20 \\ 1/20 & 1/20 & 1/20 & 1/10 \end{pmatrix} \tag{3.17}$$

The integration cannot be carried out for the test function multiplied by the intensity I because the shape of the intensity is not linear. Instead, the IMC transport solution is modified to solve for the spatial moment of the intensity and all test functions within a cell. This adds additional complexity to the particle transport that is addressed in Section 5. The linear opacity shape also adds complexity to the particle transport—the particle now experiences a changing scattering opacity as it streams. Brown and Martin [5] described the process for calculating the distance to an event in continuously changing media, that method is used here for scattering events.

In addition to a linear opacity, it may be necessary to use an average opacity value for the element. If a constant opacity is used, there are several options to get the average opacity for an element, the first is to take the average opacity of the nodes:

$$\overline{\sigma}^n = \frac{1}{k} \sum_{n=1}^{k} \sigma_i^n. \tag{3.18}$$

Where k is the number of nodes in an element. The opacity at the average temperature could also be used:

$$\overline{\sigma}^n = \sigma^n(\overline{T}) \rightarrow \overline{T} = \frac{1}{k} \sum_{n=1}^{k} T_i \tag{3.19}$$

Or the average T^4 can be used, as is done when selecting the opacity at a face in TRT diffusion:

$$\overline{\sigma^n} = \sigma^n(\overline{T}) \rightarrow \overline{T} = \left(\frac{1}{k} \sum_{n=1}^{k} T_i^4 \right)^{1/4} \tag{3.20}$$

For each element, the moment of the test functions and the absorption is determined from the transport step:

$$\langle \sigma^n I \rangle_i = \int\limits_0^{\infty} \int\limits_{4\pi} \int\limits_{\mathbf{x}} N^i \sigma^n I \; d\mathbf{x}_i \; d\Omega \; d\nu \tag{3.21}$$

Where $\langle \sigma_i I \rangle_i$ is the moment of the absorption and the basis function for node i. The absorption within a node can be projected onto a linear space as well. The absorption then becomes a sum of the coefficients for each node multiplied by the linear basis function for that node:

$$(\sigma^n I)(\mathbf{x}) = \sum_{i=1}^{k} (\sigma^n I)_i N^i(\mathbf{x}) \tag{3.22}$$

The linear reconstruction must satisfy the integral equations for each moment:

$$\langle \sigma^n I \rangle_i = \int\limits_{\mathbf{x}} \sum_{j=1}^{k} (\sigma^n I)_j N^j(\mathbf{x}) \tag{3.23}$$

In tetrahedra, a 4 by 4 matrix is formed that relates the moments of the absorption to the linear coefficients. If the continuous finite element discretization is used with the assumption that the absorption is linear, Eq. (3.22)) is multiplied by the basis function for node i and integrated over the element to get the absorption term in the TRT material energy balance:

$$\int\limits_{\mathbf{x}} N^i N^j \frac{dU_m}{dt} \; dx = \int\limits_{\mathbf{x}} \int\limits_0^{\infty} \int\limits_0^{4\pi} N^j (\sigma_a^n I)(\mathbf{x}) \; d\Omega \; d\nu \; dx - \int\limits_{\mathbf{x}} N^j E \; dx. \tag{3.24}$$

In this case, the absorption term in the material energy balance then becomes the absorption moment $\langle \sigma I \rangle_i$ as is shown in Eq. (3.23)).

The moments of the absorption could be used as the linear coefficient in Eq. (3.22) (this is equivalent to lumping the solution in Eq. (3.23)). If this is done, the absorption moment of all nodes within an element would contribute to the energy absorbed at a node. Implicit capture with basis functions and non-constant opacity is explained in Section 5.

3.5 Numerical Analysis

3.5.1 Diffusion Limit Analysis of the Node-Centered Method

The diffusion limit behavior of the node-centered IMC method at equilibrium is analyzed here. We begin with the equilibrium, 1D, transport solution:

$$\frac{1}{c}\frac{\partial \psi(x,\mu,t)}{\partial t} + \mu \frac{\partial \psi(x,\mu)}{\partial x} + \sigma(x)\psi(x,\mu) = \frac{\sigma(x)}{2}\left(fB(x) + (1-f)\phi(x)\right), \qquad (3.25)$$

where,

$$\phi(x) = \int_{-1}^{1} \psi(x,\mu)\, d\mu, \qquad (3.26)$$

and $\psi(x,\mu)$ is the Monte Carlo solution for the angular flux—in the analysis it is taken to be analytic.

The node-centered IMC method has a piecewise-linear emission profile. For the discrete node-centered IMC method, the 1D unknowns are shown in Fig. (3.12). The emission profile between cell centers is given by:

$$B(x) = B_i \frac{x_{i+1} - x}{\Delta x_i} + B_{i+1} \frac{x - x_i}{\Delta x_i}, \quad x \in (x_i, x_{i+1}), \tag{3.27}$$

Thus there are two piecewise-linear functions within a cell that are continuous and have the value of B_i at the cell center. The coefficients of the emission shape are determined by integrating the flux over a cell.

$$B_i = \left\{ \begin{array}{ll} \int\limits_{x_1}^{x_{1+1/2}} \frac{\sigma(x)\phi(x)}{\sigma_1 \Delta x_1} \, dx, & i = 1 \\ \int\limits_{x_{i-1/2}}^{x_{i+1/2}} \frac{\sigma(x)\phi(x)}{\sigma_i \Delta x_i} \, dx, & i = 2, ..., I-1 \\ \int\limits_{x_{I-1/2}}^{x_I} \frac{\sigma(x)\phi(x)}{\sigma_I \Delta x_I} \, dx, & i = I \end{array} \right\} \tag{3.28}$$

The scaling and methodology of this analysis generally follows that of Larsen [34]. In the standard discrete diffusion limit analysis, the optical depth of a mesh cell is scaled according to:

$$\sigma_i \Delta x \to \frac{\sigma_i \Delta x_i}{\epsilon} \tag{3.29}$$

where $\sigma_i \Delta_i = \mathrm{O}\big(1\big)$ and $\epsilon \to 0$.

The solution is assumed to be separable into boundary layer and interior components:

$$\psi(x, \mu) = \psi_b(x, \mu) + \psi_n(x, \mu), \tag{3.30}$$

where the boundary solution and interior solution both satisfy the equilibrium transport solution:

$$\frac{1}{c}\frac{\partial \psi_{\mathrm{b}}(x,\mu,t)}{\partial t} + \mu\frac{\partial \psi_{\mathrm{b}}(x,\mu)}{\partial x} + \sigma(x)\psi_{\mathrm{b}}(x,\mu) = \frac{\sigma(x)}{2}\left(fB_{\mathrm{b}}(x) + (1-f)\phi_{\mathrm{b}}(x)\right), \tag{3.31a}$$

$$\frac{1}{c}\frac{\partial \psi_{\mathrm{n}}(x,\mu,t)}{\partial t} + \mu\frac{\partial \psi_{\mathrm{n}}(x,\mu)}{\partial x} + \sigma(x)\psi_{\mathrm{n}}(x,\mu) = \frac{\sigma(x)}{2}\left(fB_{\mathrm{n}}(x) + (1-f)\phi_{\mathrm{n}}(x)\right). \tag{3.31b}$$

where the boundary solution also satisfies:

$$\psi_{\mathrm{b}}(x_{1/2},\mu) = \psi_{\mathrm{inc}}(\mu) - \psi_{\mathrm{n}}(x_{1/2},\mu), \qquad \mu > 0, \tag{3.32a}$$

$$\lim_{x\to\infty}\psi_{\mathrm{b}}(x,\mu) = 0. \tag{3.32b}$$

The boundary at x_I has similar conditions.

We focus on the implications of the interior solution with the scaling from Eq. (3.29). The transport equation then becomes:

$$\frac{\epsilon}{c}\frac{\partial \psi(x,\mu,t)}{\partial t} + \mu\frac{\partial \psi_{\mathrm{n}}(x,\mu)}{\partial x} + \frac{\sigma(x)}{\epsilon}\psi_{\mathrm{n}}(x,\mu) = \frac{1}{\epsilon}\frac{\sigma(x)}{2}\left(fB_{\mathrm{n}}(x) + (1-f)\phi_{\mathrm{n}}(x)\right). \tag{3.33}$$

The coefficients in $B_{\mathrm{n}}(x)$ given in Eq. (3.28) have an opacity in the numerator and denominator and thus all ϵ terms cancel inside the coefficients. Expanding ψ_{n}, ϕ_{n} and B_{n} with a power series in ϵ yields:

$$\psi_{\mathrm{n}}(x,\mu) = \psi_{\mathrm{n}}^{(0)}(x,\mu) + \epsilon\psi_{\mathrm{n}}^{(1)}(x,\mu) + \epsilon^2\psi_{\mathrm{n}}^{(2)}(x,\mu) + ... \tag{3.34a}$$

$$\phi_{\mathrm{n}}(x,\mu) = \phi_{\mathrm{n}}^{(0)}(x,\mu) + \epsilon\phi_{\mathrm{n}}^{(1)}(x,\mu) + \epsilon^2\phi_{\mathrm{n}}^{(2)}(x,\mu) + ... \tag{3.34b}$$

$$B_{\mathrm{n}}(x) = B_{\mathrm{n}}^{(0)}(x,\mu) + \epsilon B_{\mathrm{n}}^{(1)}(x,\mu) + \epsilon^2 B_{\mathrm{n}}^{(2)}(x,\mu) + ... \tag{3.34c}$$

Multiplying Eq. (3.51) by ϵ produces the $\mathrm{O}\big(1\big)$ solution:

$$\psi_{\mathrm{n}}^{(0)}(x,\mu) = \frac{1}{2}\left(fB_{\mathrm{n}}^{(0)}(x) + (1-f)\phi_{\mathrm{n}}^{(0)}(x)\right) \tag{3.35}$$

Integrating over angle:

$$\phi_{\mathrm{n}}^{(0)}(x) = fB_{\mathrm{n}}^{(0)}(x) + (1-f)\phi_{\mathrm{n}}^{(0)}(x), \tag{3.36}$$

which simplifies to

$$\phi_{\mathrm{n}}^{(0)}(x) = B_{\mathrm{n}}^{(0)} \tag{3.37}$$

And from Eq. (3.51),

$$\psi_{\mathrm{n}}^{(0)}(x,\mu) = \frac{1}{2}B_{\mathrm{n}}^{(0)}(x) = \frac{1}{2}\phi_{\mathrm{n}}^{(0)}(x). \tag{3.38}$$

The emission function, $B_{\mathrm{n}}(x)$, is piecewise-linear, meaning that the leading order intensity is also piecewise-linear and isotropic.

Because the system is assumed to be in equilibrium, the intensity should produce coefficients for $B(x)$ that in turn produce the same intensity. We now examine the coefficients of $B_{\mathrm{n}}^{(0)}$ ($B_{\mathrm{n},i}^{(0)}$ and $B_{\mathrm{n},i+1}^{(0)}$), given that the leading order intensity is piecewise linear:

$$B_{\mathrm{n},i}^{(0)} = \int\limits_{x_{i-1/2}}^{i+1/2} \frac{\sigma(x)\phi_{\mathrm{n}}^{(0)}(x)}{\sigma_i \Delta x_i}\, dx, \tag{3.39}$$

Substituting the piecewise linear $\phi_{\mathrm{n}}^{(0)}$ into Eq. (3.39). There are two piecewise-linear functions over $x \in (x_{i-1/2}, x_{i+1/2})$ so the integral is broken into two parts:

$$\begin{aligned} B_{\mathrm{n},i}^{(0)} = \frac{1}{\sigma_i \Delta x_i} \int\limits_{x_{i-1/2}}^{x_i} \sigma(x) \left(B_{\mathrm{n},i}^{(0)} \frac{x_{i+1} - x}{\Delta x_i} + B_{\mathrm{n},i-1}^{(0)} \frac{x - x_i}{\Delta x_i} \right)\, dx + \\ \frac{1}{\sigma_i \Delta x_i} \int\limits_{x_i}^{x_{i+1/2}} \sigma(x) \left(B_{\mathrm{n},i}^{(0)} \frac{x_{i+1} - x}{\Delta x_i} + B_{\mathrm{n},i+1}^{(0)} \frac{x - x_i}{\Delta x_i} \right)\, dx. \end{aligned} \tag{3.40}$$

Carrying out the integration:

$$B_{\mathrm{n},i}^{(0)} = \frac{1}{\sigma_i \Delta x_i} \left(\frac{(\sigma \Delta x)_i}{2} \left(\frac{1}{4} B_{\mathrm{n},i-1}^{(0)} + \frac{3}{4} B_{\mathrm{n},i}^{(0)} \right) + \frac{(\sigma \Delta x)_i}{2} \left(\frac{1}{4} B_{\mathrm{n},i+1}^{(0)} + \frac{3}{4} B_{\mathrm{n},i}^{(0)} \right) \right) \tag{3.41}$$

Simplifying:

$$0 = B_{\mathrm{n},i-1}^{(0)} - B_{\mathrm{n},i}^{(0)} + B_{\mathrm{n},i+1}^{(0)} - B_{\mathrm{n},i}^{(0)} \tag{3.42}$$

which can be written as:

$$\frac{1}{3\sigma_i} \frac{B_{\mathrm{n},i+1}^{(0)} - B_{\mathrm{n},i}^{(0)}}{\Delta x_i / 2} - \frac{1}{3\sigma_i} \frac{B_{\mathrm{n},i}^{(0)} - B_{\mathrm{n},i-1}^{(0)}}{\Delta x_i / 2} = 0 \tag{3.43}$$

This result is incorrect and implies that a linear intensity will not reproduce the equilibrium emission coefficients unless the intensity is constant everywhere in the problem. This problem stems from a mismatch in the emission and absorption shapes.

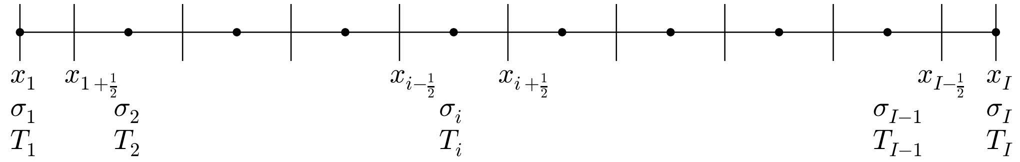


Figure 3.12: 1D view of node-centered method showing the location of the unknowns

3.5.2 Absorption Based Emission Analysis

To obtain the correct diffusion limit we attempt to force the emission and absorption to have the same shape. This leads to a different way of calculating the emission shape for the node-centered method. This new method, which we call "absorption based emission," is analyzed here. This method uses sub-cell absorption tallies to calculate a linear emission profile for the next timestep. The location of the unknowns is again shown in Fig. (3.12). Cell centers are at integer values and faces are at $i + 1/2$ and $i - 1/2$. The emission is treated as discontinuous at cell-centers and continuous across cell faces, meaning the emission profile is described by a left and right value for each cell center:

$$B(x) = B_{i,\mathrm{R}} \frac{x_{i+1} - x}{x_{i+1} - x_i} + B_{i+1,\mathrm{L}} \frac{x - x_i}{x_{i+1} - x_i}, \quad x \in (x_i, x_{i+1}), \tag{3.44}$$

where B is the emission shape and $B_{i,\mathrm{R}}$ and $B_{i+1,\mathrm{L}}$ are linear coefficients.

The coefficients are determined by solving for a linear scalar intensity that reproduces the sub-cell absorption integral. We consider the equilibrium case, and thus there is no need to differentiate the time of absorption from emission:

$$\int_{x_i}^{x_{i+1/2}} \sigma(x)\left(B_{i,\mathrm{R}}\frac{x_{i+1}-x}{x_{i+1}-x_i}+B_{i+1,\mathrm{L}}\frac{x-x_i}{x_{i+1}-x_i}\right)\,dx = \int_{x_i}^{x_{i+1/2}} \sigma(x)\phi(x)\,dx, \tag{3.45a}$$

$$\int_{x_{i+1/2}}^{x_{i+1}} \sigma(x)\left(B_{i,\mathrm{R}}\frac{x_{i+1}-x}{x_{i+1}-x_i}+B_{i+1,\mathrm{L}}\frac{x-x_i}{x_{i+1}-x_i}\right)\,dx = \int_{x_{i+1/2}}^{x_{i+1}} \sigma(x)\phi(x)\,dx. \tag{3.45b}$$

The purpose of this analysis is to determine if using the emission in Eq. (3.44) with coefficients from Eq. (3.45) introduces spurious terms that cause the solution to diverge from an equilibrium solution, as was shown in the node-centered method analysis above.

Using a constant opacity within a cell, as is done in IMC, yields:

$$\sigma_i \int_{x_i}^{x_{i+1/2}} B_{i,\mathrm{R}}\frac{x_{i+1}-x}{x_{i+1}-x_i}+B_{i+1,\mathrm{L}}\frac{x-x_i}{x_{i+1}-x_i}\,dx = \sigma_i \int_{x_i}^{x_{i+1/2}} \phi(x)\,dx, \tag{3.46a}$$

$$\sigma_{i+1} \int_{x_{i+1/2}}^{x_{i+1}} B_{i,\mathrm{R}}\frac{x_{i+1}-x}{x_{i+1}-x_i}+B_{i+1,\mathrm{L}}\frac{x-x_i}{x_{i+1}-x_i}\,dx = \sigma_{i+1} \int_{x_{i+1/2}}^{x_{i+1}} \phi(x)\,dx. \tag{3.46b}$$

Canceling opacity and carrying out these integrals gives the matrix form of Eq. (3.46):

$$\frac{x_{i+1}-x_i}{2}\begin{pmatrix}3/4 & 1/4\\ 1/4 & 3/4\end{pmatrix}\begin{pmatrix}B_{i,\mathrm{R}}\\ B_{i+1,\mathrm{L}}\end{pmatrix} = \begin{pmatrix}\int_{x_i}^{x_{i+1/2}} \phi(x)\,dx\\ \int_{x_{i+1/2}}^{x_{i+1}} \phi(x)\,dx\end{pmatrix}. \tag{3.47}$$

This method is examined in 1D diffusion limit with the additional assumption that the material and radiation are in equilibrium. In 1D, the IMC transport equation is:

$$\frac{1}{c}\frac{\partial \psi(x,\mu,t)}{\partial t}+\mu\frac{\partial \psi(x,\mu,t)}{\partial x}+\sigma(x)\psi(x,\mu,t)=\frac{\sigma(x)}{2}\left(fB(x)+(1-f)\phi(x,t)\right), \quad (3.48)$$

where the intensity is assumed to be taken as analytic. We consider the interior solution with the standard scaling applied, which yields:

$$\frac{\epsilon}{c}\frac{\partial \psi(x,\mu,t)}{\partial t}+\mu\frac{\partial \psi_{\mathrm{n}}(x,\mu)}{\partial x}+\frac{\sigma(x)}{\epsilon}\psi_{\mathrm{n}}(x,\mu)=\frac{1}{\epsilon}\frac{\sigma(x)}{2}\left(fB_{\mathrm{n}}(x)+(1-f)\phi_{\mathrm{n}}(x)\right). \quad (3.49)$$

Because $B_{\mathrm{n}}(x)$ is a piecewise linear function, we look at the scaling of the coefficients. In 1D, the coefficients of $B_{\mathrm{n}}(x)$ are $B_{\mathrm{n},i,R}$ and $B_{\mathrm{n},i+1,L}$. An expression for each coefficient can be seen in the solution of Eq. (3.47), which yields:

$$B^{(0)}_{\mathrm{n},i,R}=\phi^{(0)}_{n,i,R}=\frac{1}{x_{i+1}-x_i}\left(3\int\limits_{x_i}^{x_{i+1/2}}\phi_n(x)\;dx-\int\limits_{x_{i+1/2}}^{x_{i+1}}\phi_n(x)\;dx\right), \quad (3.50)$$

There are no opacity terms in the coefficients of B_{n} and thus no scaling is applied. Multiplying Eq. (3.51) by ϵ, substituting the expansions for ψ_{n}, ϕ_{n} and B_{n}, and grouping $\mathrm{O}\big(1\big)$ terms:

$$\psi^{(0)}_{\mathrm{n}}(x,\mu)=\frac{1}{2}\left(fB^{(0)}_{\mathrm{n}}(x)+(1-f)\phi^{(0)}_{\mathrm{n}}(x)\right). \quad (3.51)$$

Integrating over angle:

$$\phi^{(0)}_{\mathrm{n}}(x)=fB^{(0)}_{\mathrm{n}}(x)+(1-f)\phi^{(0)}_{\mathrm{n}}(x), \quad (3.52)$$

which simplifies to

$$\phi_{\mathrm{n}}^{(0)}(x) = B_{\mathrm{n}}^{(0)}. \tag{3.53}$$

And from Eq. (3.51),

$$\psi_{\mathrm{n}}^{(0)}(x,\mu) = \frac{1}{2}B_{\mathrm{n}}^{(0)}(x) = \frac{1}{2}\phi_{\mathrm{n}}^{(0)}(x). \tag{3.54}$$

The emission function, $B_{\mathrm{n}}(x)$, is piecewise-linear, meaning that the leading order intensity is also piecewise-linear and isotropic. Because leading order radiation intensity is isotropic, the leading order flux is zero:

$$F_{\mathrm{n}}^{(0)} = \int_{-1}^{1} \mu\psi_{\mathrm{n}}^{(0)}(x,\mu)\ d\mu = \int_{-1}^{1} \mu\frac{\phi_{\mathrm{n}}^{(0)}(x)}{2}\ d\mu = 0. \tag{3.55}$$

The radiation pressure with an isotropic intensity is:

$$P_{\mathrm{n}}^{(0)} = \int_{-1}^{1} \mu^2\psi_{\mathrm{n}}^{(0)}(x,\mu)\ d\mu = \int_{-1}^{1} \mu^2\frac{\phi_{\mathrm{n}}^{(0)}(x)}{2}\ d\mu = \frac{1}{3}\phi_{\mathrm{n}}^{(0)}. \tag{3.56}$$

We can now examine if the leading order intensity generates an emission profile that reproduces the equilibrium solution. The linear coefficients in $B_{\mathrm{n}}^{(0)}$ can be calculated. Examining the coefficient $B_{\mathrm{n},i,R}^{(0)}$:

$$B_{\mathrm{n},i,R}^{(0)} = \phi_{\mathrm{n},i,R}^{(0)}. \tag{3.57}$$

Solving Eq. (3.47) for $\phi_{n,i,R}^{(0)}$ and substituting into Eq. (3.57) yields an expression for $B_{\mathrm{n},i,R}^{(0)}$ in terms of the left and right absorption integrals:

$$B^{(0)}_{\mathrm{n},i,R} = \phi^{(0)}_{n,i,R} = \frac{1}{x_{i+1} - x_i} \left(3 \int\limits_{x_i}^{x_{i+1/2}} \phi_n(x)\ dx - \int\limits_{x_{i+1/2}}^{x_{i+1}} \phi_n(x)\ dx \right). \tag{3.58}$$

The interior intensity in the absorption integrals is replaced with the piecewise linear intensity, given by $B_{\mathrm{n}}(x)$:

$$B^{(0)}_{\mathrm{n},i,R} = \frac{1}{x_{i+1} - x_i} \left(3 \int\limits_{x_i}^{x_{i+1/2}} B_{\mathrm{n}}^{(i,R)} \frac{x_{i+1} - x}{x_{i+1} - x_i} + B_{\mathrm{n}}^{(i+1,L)} \frac{x - x_i}{x_{i+1} - x_i}\ dx \right)$$
$$- \frac{1}{x_{i+1} - x_i} \left(\int\limits_{x_{i+1/2}}^{x_{i+1}} B_{\mathrm{n}}^{(i,R)} \frac{x_{i+1} - x}{x_{i+1} - x_i} + B_{\mathrm{n}}^{(i+1,L)} \frac{x - x_i}{x_{i+1} - x_i}\ dx \right), \tag{3.59}$$

The equations now only contain the linear coefficients for the emission. Integrating and simplifying:

$$B^{(0)}_{\mathrm{n},i,R} = \frac{1}{x_{i+1} - x_i} (\frac{3(x_{i+1} - x_i)}{2} \left(\frac{3}{4} B^{(0)}_{\mathrm{n},i,R} + \frac{1}{4} B^{(0)}_{\mathrm{n},i+1,L} \right)$$
$$- \frac{(x_{i+1} - x_i)}{2} \left(\frac{1}{4} B^{(0)}_{\mathrm{n},i,R} + \frac{3}{4} B^{(0)}_{\mathrm{n},i+1,L} \right)),$$

$$B^{(0)}_{\mathrm{n},i,R} = \frac{9}{8} B^{(0)}_{\mathrm{n},i,R} + \frac{3}{8} B^{(0)}_{\mathrm{n},i+1,L} - \frac{1}{8} B^{(0)}_{\mathrm{n},i,R} - \frac{3}{8} B^{(0)}_{\mathrm{n},i+1,L},$$

$$B^{(0)}_{\mathrm{n},i,R} = B^{(0)}_{\mathrm{n},i,R}.$$

This result implies that the leading order intensity in the diffusion limit will hold an equilibrium solution by reproducing the same linear coefficients used in the emission. No spurious terms are introduced into the essentially steady -state equation by using

Eq. (3.44) with coefficients from Eq. (3.45).

We now examine the $\mathrm{O}\big(\epsilon\big)$ and $\mathrm{O}\big(\epsilon^2\big)$ equations to look for correct behavior in the diffusion limit. The first order $(\mathrm{O}\big(\epsilon\big))$ transport equation is:

$$\mu\frac{\partial\psi_{\mathrm{n}}^{(0)}(x,\mu)}{\partial x}+\sigma(x)\psi_{\mathrm{n}}^{(1)}(x,\mu)=\frac{\sigma(x)}{2}\left(fB_{\mathrm{n}}^{(1)}(x)+(1-f)\phi_{\mathrm{n}}^{(1)}(x)\right). \tag{3.60}$$

Taking the first angular moment of Eq. (3.60) and simplifying yields Fick's law for the first order flux:

$$\frac{\partial P_{\mathrm{n}}^{(0)}(x)}{\partial x}+\sigma(x)F_{\mathrm{n}}^{(1)}(x,\mu)=0,$$

$$\frac{1}{3\sigma(x)}\frac{\partial\phi_{\mathrm{n}}^{(0)}(x)}{\partial x}+F_{\mathrm{n}}^{(1)}(x,\mu)=0,$$

$$F_{\mathrm{n}}^{(1)}(x,\mu)=-\frac{1}{3\sigma(x)}\frac{\partial\phi_{\mathrm{n}}^{(0)}(x)}{\partial x}. \tag{3.61}$$

The second order transport equation is:

$$\begin{aligned}\frac{1}{c}\frac{\partial\psi_{\mathrm{n}}^{(0)}(x,\mu,t)}{\partial t}&+\mu\frac{\partial\psi_{\mathrm{n}}^{(1)}(x,\mu,t)}{\partial x}+\sigma(x)\psi_{\mathrm{n}}^{(2)}(x,\mu,t)\\&=\frac{\sigma(x)}{2}\left(fB_{\mathrm{n}}^{(2)}(x,t)+(1-f)\phi_{\mathrm{n}}^{(2)}(x,t)\right).\end{aligned} \tag{3.62}$$

Suppressing time arguments and taking the zeroth angular moment of Eq. (3.62):

$$\frac{1}{c}\frac{\partial\phi_{\mathrm{n}}^{(0)}(x)}{\partial t}+\frac{\partial F_{\mathrm{n}}^{(1)}(x)}{\partial x}+\sigma(x)\phi_{\mathrm{n}}^{(2)}(x)=\sigma(x)\left(fB_{\mathrm{n}}^{(2)}(x)+(1-f)\phi_{\mathrm{n}}^{(2)}(x)\right). \tag{3.63}$$

To eliminate the emission and scattering terms, the second order transport equation is combined with the second order, stead-state material energy balance, which is:

$$c_{\mathrm{V}}\frac{\partial T_{\mathrm{n}}^{(0)}(x)}{\partial t} = f\sigma(x)\phi_{\mathrm{n}}^{(2)}(x) - f\sigma(x)B_{\mathrm{n}}^{(2)}(x). \tag{3.64}$$

Combing Eq. (3.63) and Eq. (3.64) yields:

$$c_{\mathrm{V}}\frac{\partial T_{\mathrm{n}}^{(0)}(x)}{\partial t} + \frac{1}{c}\frac{\partial \phi_{\mathrm{n}}^{(0)}(x)}{\partial t} + \frac{\partial F_{\mathrm{n}}^{(1)}(x)}{\partial x} = 0. \tag{3.65}$$

Using Eq. (3.61) and simplifying:

$$c_{\mathrm{V}}\frac{\partial T_{\mathrm{n}}^{(0)}(x)}{\partial t} + \frac{1}{c}\frac{\partial \phi_{\mathrm{n}}^{(0)}(x)}{\partial t} - \frac{\partial}{\partial x}\frac{1}{3\sigma(x)}\frac{\partial \phi_{\mathrm{n}}^{(0)}(x)}{\partial x} = 0,$$

$$c_{\mathrm{V}}\frac{\partial T_{\mathrm{n}}^{(0)}(x)}{\partial t} + \frac{1}{c}\frac{\partial \phi_{\mathrm{n}}^{(0)}(x)}{\partial t} - \frac{\partial}{\partial x}\frac{1}{3\sigma(x)}\frac{\partial B_{\mathrm{n}}^{(0)}(x)}{\partial x} = 0. \tag{3.66}$$

Eq. (3.66) is then integrated between node edges: $x \in (x_{i-1/2}, x_{i+1/2})$:

$$\int\limits_{x_{i-1/2}}^{x_{i+1/2}} c_{\mathrm{V}}\frac{\partial T_{\mathrm{n}}^{(0)}(x)}{\partial t} + \frac{1}{c}\frac{\partial \phi_{\mathrm{n}}^{(0)}(x)}{\partial t}\, dx - \frac{1}{3\sigma_i}\left(\left.\frac{\partial B_{\mathrm{n}}^{(0)}(x)}{\partial x}\right|_{x=x_{i+1/2}} - \left.\frac{\partial B_{\mathrm{n}}^{(0)}(x)}{\partial x}\right|_{x=x_{i-1/2}}\right) = 0. \tag{3.67}$$

The opacity is constant within a cell and thus becomes σ_i. The spatial derivatives can be evaluated by substituting the linear emission function for $B_{\mathrm{n}}^{(0)}(x)$ in Eq. (3.44):

$$\int\limits_{x_{i-1/2}}^{x_{i+1/2}} c_{\mathrm{V}}\frac{\partial T_{\mathrm{n}}^{(0)}(x)}{\partial t} + \frac{1}{c}\frac{\partial \phi_{\mathrm{n}}^{(0)}(x)}{\partial t}\, dx -$$

$$\frac{1}{3\sigma_i}\left(\left(\frac{B_{\mathrm{n},i+1,L}^{(0)} - B_{\mathrm{n},i,R}^{(0)}}{x_{i+1} - x_i}\right) - \left(\frac{B_{\mathrm{n},i,L}^{(0)} - B_{\mathrm{n},i-1,R}^{(0)}}{x_i - x_{i-1}}\right)\right) = 0. \tag{3.68}$$

Eq. (3.68) represents a discrete diffusion equation at node edges.

3.5.3 Analysis of Opacity Averaging in DFEM IMC

The DFEM IMC method is analyzed for case where the intensity from the transport solution is a linear function:

$$I(x) = I_L N_L(x) + I_R N_R(x). \tag{3.69}$$

In this analysis, all intensity values are assumed to be angle-integrated quantities. The opacity is treated as linear and assumed to be proportional to the T^{-3}, given by:

$$\sigma_{\mathrm{a}}(x) = \sigma_{\mathrm{a,L}} N_L(x) + \sigma_{\mathrm{a,R}} N_R(x), \tag{3.70}$$

which yields,

$$\sigma_{\mathrm{a,L}} = \frac{\sigma_{\mathrm{a,o}}}{T_L^3}, \tag{3.71}$$

$$\sigma_{\mathrm{a,R}} = \frac{\sigma_{\mathrm{a,o}}}{T_R^3}. \tag{3.72}$$

With a linear opacity and linear intensity, the material energy balance for left and right temperatures becomes:

$$\int\limits_{\mathbf{x}} f\sigma_{\mathrm{a}}(x) N_L(x) I(x)\ dx = \frac{acf\sigma_{\mathrm{a,L}} T_L^4}{2}, \tag{3.73}$$

$$\int\limits_{\mathbf{x}} f\sigma_{\mathrm{a}}(x) N_R(x) I(x)\ dx = \frac{acf\sigma_{\mathrm{a,R}} T_R^4}{2}. \tag{3.74}$$

Carrying out the integration and replacing left and right opacity values with the temperature expression:

$$\frac{I_L}{4T_L^3} + \frac{I_L}{12T_R^3} + \frac{I_R}{12T_L^3} + \frac{I_R}{12T_R^3} = \frac{1}{2}T_L \tag{3.75}$$

$$\frac{I_R}{4T_R^3} + \frac{I_R}{12T_L^3} + \frac{I_L}{12T_L^3} + \frac{I_L}{12T_R^3} = \frac{1}{2}T_R \tag{3.76}$$

This shows that using a linear opacity in the DFEM IMC case is not guaranteed to reproduce the steady-state solution unless the intensity and temperature are flat ($I_L = I_R$ and $T_L = T_R$). The intensity coefficients that do reproduce the steady-state solution are:

$$I_L = \frac{\frac{T_L}{2}\left(\frac{\sigma_L}{12} + \frac{\sigma_R}{4}\right) - \frac{T_R}{2}\left(\frac{\sigma_L}{12} + \frac{\sigma_R}{12}\right)}{-\left(\frac{\sigma_L}{12} + \frac{\sigma_R}{12}\right)^2 + \left(\frac{\sigma_L}{12} + \frac{\sigma_R}{4}\right)\left(\frac{\sigma_L}{4} + \frac{\sigma_R}{12}\right)} \tag{3.77}$$

$$I_R = \frac{-\frac{T_L}{2}\left(\frac{\sigma_L}{12} + \frac{\sigma_R}{12}\right) + \frac{T_R}{2}\left(\frac{\sigma_L}{4} + \frac{\sigma_R}{12}\right)}{-\left(\frac{\sigma_L}{12} + \frac{\sigma_R}{12}\right)^2 + \left(\frac{\sigma_L}{12} + \frac{\sigma_R}{4}\right)\left(\frac{\sigma_L}{4} + \frac{\sigma_R}{12}\right)} \tag{3.78}$$

The standard method of selecting an opacity at a cell face in TRT diffusion is to use an opacity calculated with the average of the material temperature to the fourth power:

$$\sigma(\tilde{T}) = \frac{\sigma_{\mathrm{a,o}}}{\tilde{T}^3} \rightarrow \tilde{T} = \left(\frac{T_i^4 + T_{i+1}^4}{2}\right)^{1/4} \tag{3.79}$$

This opacity averaging improves the results of the DFEM IMC method compared to using a linear opacity within an element. Using Eq. (3.79) in the DFEM method can be justified by requiring that an average opacity reproduce the average emission in a cell:

$$\overline{\sigma_{\mathrm{a}}}\overline{T^4} = \overline{\sigma T^4} \tag{3.80}$$

For a cell in the DFEM IMC method, this becomes:

$$\overline{\sigma_{\mathrm{a}}}\frac{T_L^4 + T_R^4}{2} = \frac{\sigma_{\mathrm{a,L}}T_L^4 + \sigma_{\mathrm{a,R}}T_R^4}{2} = \sigma_{\mathrm{a,o}}\frac{T_L + T_R}{2} \tag{3.81}$$

Solving for $\overline{\sigma_{\mathrm{a}}}$ yields:

$$\overline{\sigma_{\mathrm{a}}} = \sigma_{\mathrm{a,o}}\frac{T_L + T_R}{T_L^4 + T_R^4} \tag{3.82}$$

The average temperature used in an inverse cubic opacity is:

$$\tilde{T} = \left(\frac{T_L^4 + T_R^4}{T_L + T_R}\right)^{1/3} \tag{3.83}$$

Although this is not the exact form of the temperature averaging that is traditionally used, it does suggest that the temperature in an average opacity should be a T^4 average. If the opacity is constant, a simple temperature average will reproduce the average emission in a cell. The results for the Marshak wave problem are given in Fig. (3.13), they are nearly indistinguishable from the results using Eq. (3.79).

3.6 Test Problems

There are several standard test problems in TRT. These test problems are used to verify the method implementation and test the method's performance in the optically thick and thin limits.

The TRT equations are non-linear, and it is therefore difficult to obtain an analytic solution without eliminating many dependent variables. Su and Olson provided a 1D, time-dependent semi-analytic solution to the TRT equations [53]. This solu-

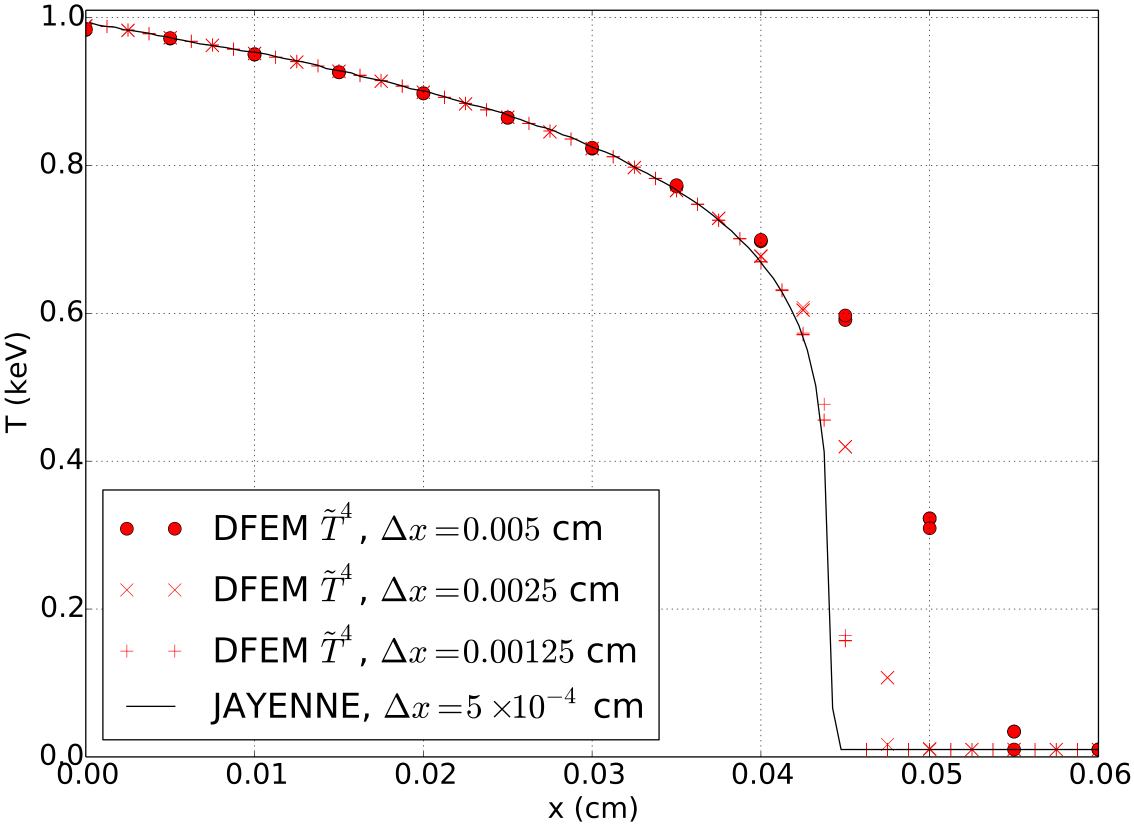


Figure 3.13: DFEM IMC results at 1 ns using the cell averaged temperature in Eq. (3.83 for various spatial resolutions

tion linearizes the TRT equations by using a heat capacity that is proportional to T^3:

$$c_V = \alpha T^3 \tag{3.84}$$

The equations are now linear in T^4. The Su-Olson solution is non-dimensional and the parameters are given in Table (3.2).

The Marshak wave problem was first proposed by Marshak in the late 1950's [40]. This problem uses an absorption opacity that is inversely proportional the cube of the material temperature. A relatively hot boundary source drives a 1D problem.

Parameter		Value
σ_a	$=$	1.0
α	$=$	4.0
c_V	$=$	αT^3
$T_{m,0}$	$=$	0.0
$T_{r,0}$	$=$	0.0
T_s	$=$	1.0
c	$=$	1.0
a	$=$	1.0
ρ	$=$	1.0

Table 3.2: Parameters of the Su-Olson problem

The inner material starts cold and is thus very opaque. As the material heats up, the material becomes thin. This creates a sharp wavefront that propagates through the material. The mean free path in a Marshak wave problem is about 1.0×10^{-8} cm in the cold material. To spatially resolve a mean free path would prohibitively expensive in both computation and memory costs. Thankfully, much work has been done to show that a method can be expected to behave correctly, even if it does not resolve a mean free path, if it holds the *equilibrium diffusion limit* [34]. This means that the spatially discretized TRT equations look like consistent, discretized, diffusion equations when certain terms in the equations are scaled by a parameter representing the optically thick limit. The Marshak wave problem tests the behavior of a method in the equilibrium diffusion limit. The parameters for the Marshak wave test problem used in this work are shown in Table (3.3).

The crooked pipe test problem was proposed by Graziani and LeBlanc [25] as a way to test method performance in optically thin and thick regimes. The crooked pipe test problem is shown in Fig. (3.14)). The temperature is tracked at 5 points in the thin section of the pipe. The energy propagation speed is highly dependent on the resolution of the thick-thin material interface and the method's behavior in the

Parameter		Value
$\sigma_a(T)$	$=$	$\frac{100}{T^3}\frac{\mathrm{cm}^2}{\mathrm{g}}$
c_V	$=$	$0.0081181\ \frac{\mathrm{jk}}{\mathrm{keV cm}^3}$
$T_{m,0}$	$=$	0.001 keV
$T_{r,0}$	$=$	0.001 keV
T_s	$=$	1.0 keV
c	$=$	$299.79\ \frac{\mathrm{cm}}{\mathrm{sh}}$
a	$=$	$0.013720160\ \frac{\mathrm{jk}}{\mathrm{cm}^3\mathrm{keV}^4}$
ρ	$=$	$3.0\ \frac{\mathrm{kg}}{\mathrm{m}^3}$

Table 3.3: Parameters of the Marshak wave problem

diffusion limit. In this work, a source temperature of $T_s = 0.5$ keV is used instead of the $T_s = 0.3$ keV as specified originally because that source temperature is used in the reference solution [23]—IMC results from the KULL code developed at Lawrence Livermore National Laboratory [22]. The parameters for the crooked pipe problem are shown in Table (3.4).

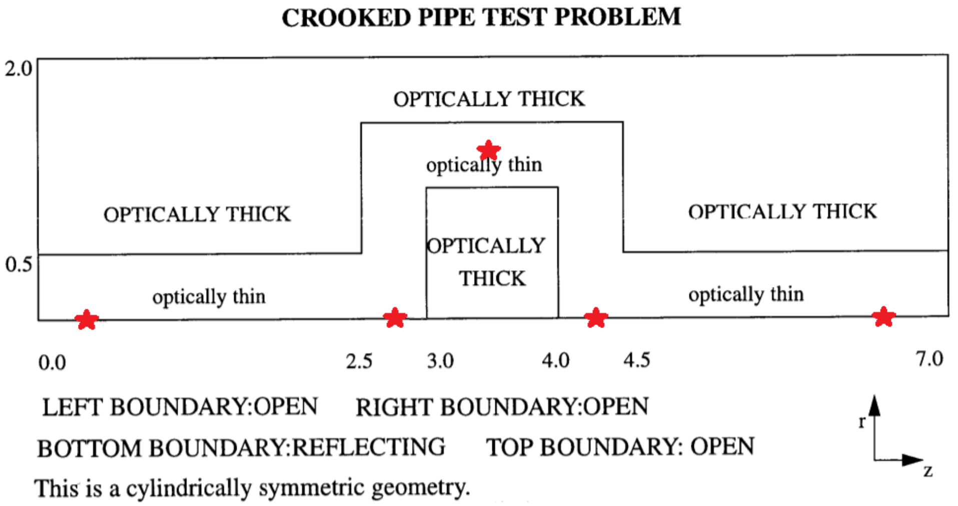


Figure 3.14: The crooked pipe or “tophat” problem from [25]—the points where the solution is tracked are marked with red stars

Parameter		Value
σ_{thin}	$=$	$20.0 \frac{\text{g}}{\text{cm}^2}$
σ_{thick}	$=$	$200.0 \frac{\text{g}}{\text{cm}^2}$
ρ_{thin}	$=$	$0.01 \frac{\text{g}}{\text{cm}^3}$
ρ_{thick}	$=$	$10.0 \frac{\text{g}}{\text{cm}^3}$
c_V	$=$	$0.05 \ \frac{\text{jk}}{\text{keVcm}^3}$
$T_{m,0}$	$=$	0.05 keV
$T_{r,0}$	$=$	0.05 keV
T_s	$=$	0.5 keV
c	$=$	$299.79 \ \frac{\text{cm}}{\text{sh}}$
a	$=$	$0.013720160 \ \frac{\text{jk}}{\text{cm}^3\text{keV}^4}$

Table 3.4: Parameters of the crooked pipe problem

3.7 Numerical Results

The results for the Su-Olson problem are shown in Figs. (3.15) to (3.34). In the Su-Olson problem the node-centered and corner-centered IMC method appear to be 1D, expecting Monte Carlo noise, even though the mesh is 3D. The corner-based method results show the material temperatures at the nodes, unless noted otherwise. The Su-Olson results show that the node-centered, corner-centered and DFEM IMC method have been implemented correctly. This includes the quadrature rule used in the implicit capture in the DFEM method. The runtimes for each method using $\Delta t = 0.01$ and $\Delta x = 0.01$ are shown in Table (3.5). The runtimes show that the corner-centered method is not significantly longer than the node-centered methods. The DFEM IMC method runtime is not significantly longer than the other methods—this confirms that the addition of the quadrature rule in implicit capture is offset by eliminating the need to track to corners within the tetrahedra.

The results for the Marshak wave problem for each of the four methods are shown in Figs. (3.18) to (3.20). The Marshak wave results appear to converge to the 1D, refined JAYENNE results. The corner-based IMC method uses the same mesh as

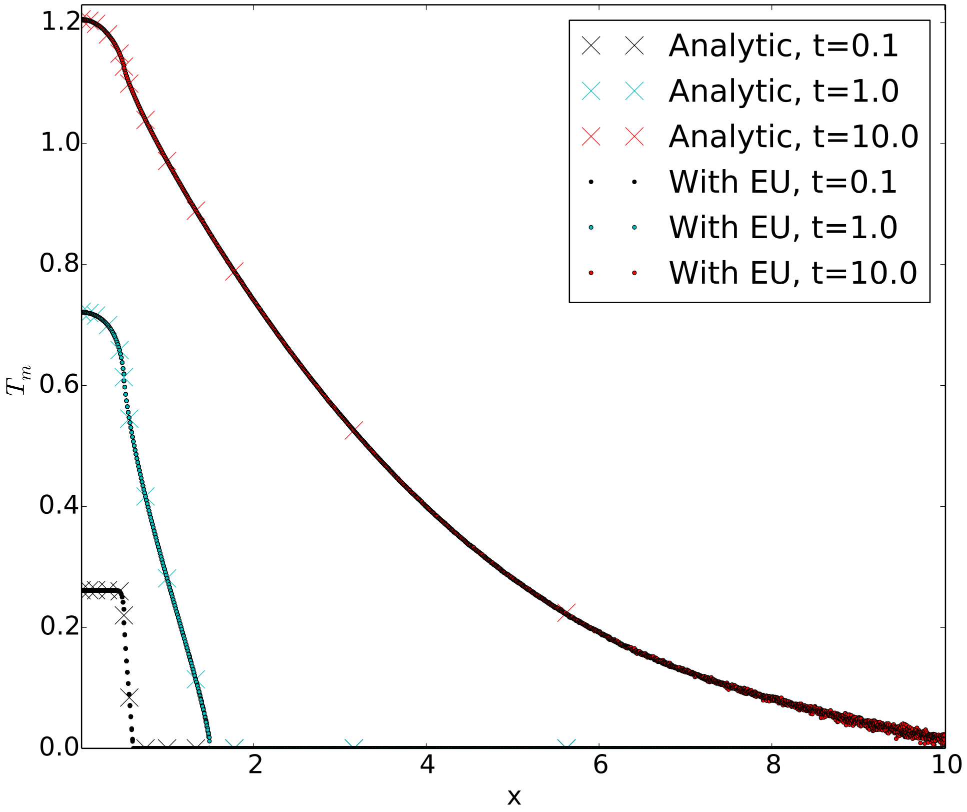


Figure 3.15: The material temperature in the Su-Olson problem with the node-based IMC method without emission upwinding at three times

the node-centered IMC method, but it effectively uses double the resolution in each dimension because it divides the dual mesh cells into corners. This can be seen in the 2D example of the corner-based mesh in Fig. (3.10)). Thus, the corner based method should be compared to a node-centered method using twice as many cells. Even with that caveat, the corner-based method gives results that are closer to the 1D Jayenne results compared to the node-centered method using mesh at twice the resolution, as seen in Fig. (3.21).

The runtimes for each method on the Marshak wave problem for $\Delta t = 0.000125$ sh and $\Delta x = 0.0025$ are shown in Table (3.5). The corner-centered method takes the

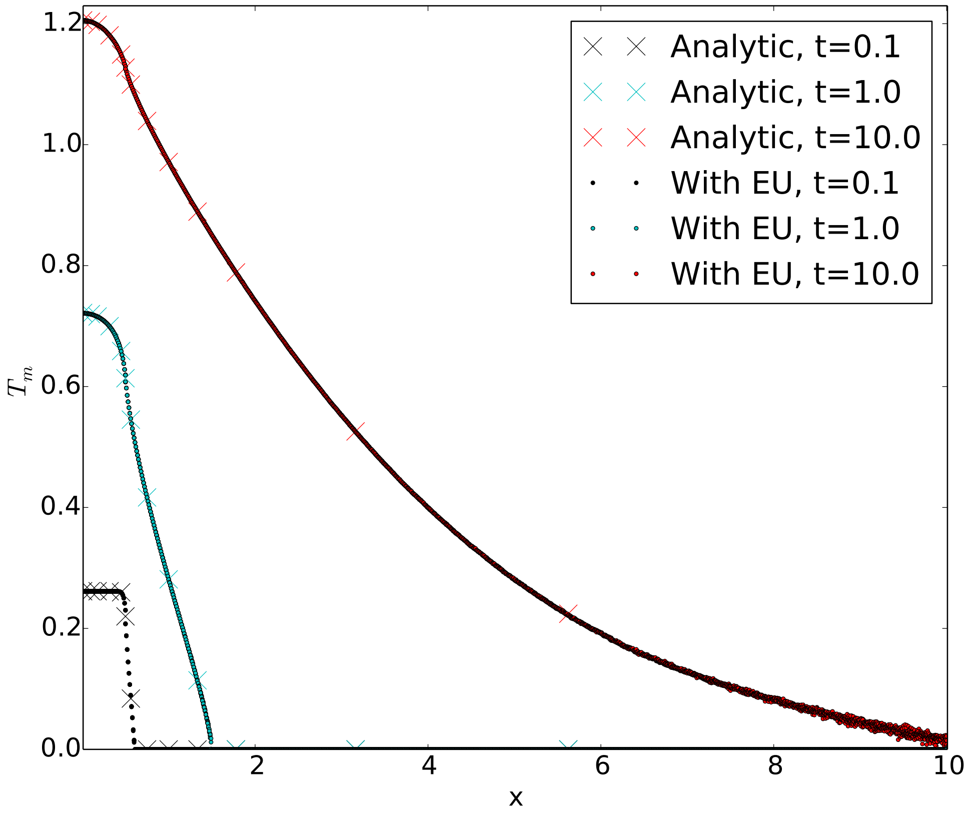


Figure 3.16: The material temperature in the Su-Olson problem with the node-based IMC method with emission upwinding at three times

longest time to complete for the Marshak wave problem of the piecewise-constant IMC methods. Although the tracking cost is the same for the corner-centered method and the node-centered methods, there is additional work from other places in the implementation related to the increased number of unknowns in the corner-centered method. The DFEM IMC method takes about 8% longer than the piecewise-constant methods. Based on the Su-Olson results, the decreased cost of particle tracking seems to be offset by the increased cost in evaluating the implicit capture integral. An increase in runtime for the DFEM IMC method relative to the other methods in the Marshak wave problem is likely caused by particles encountering higher opacity

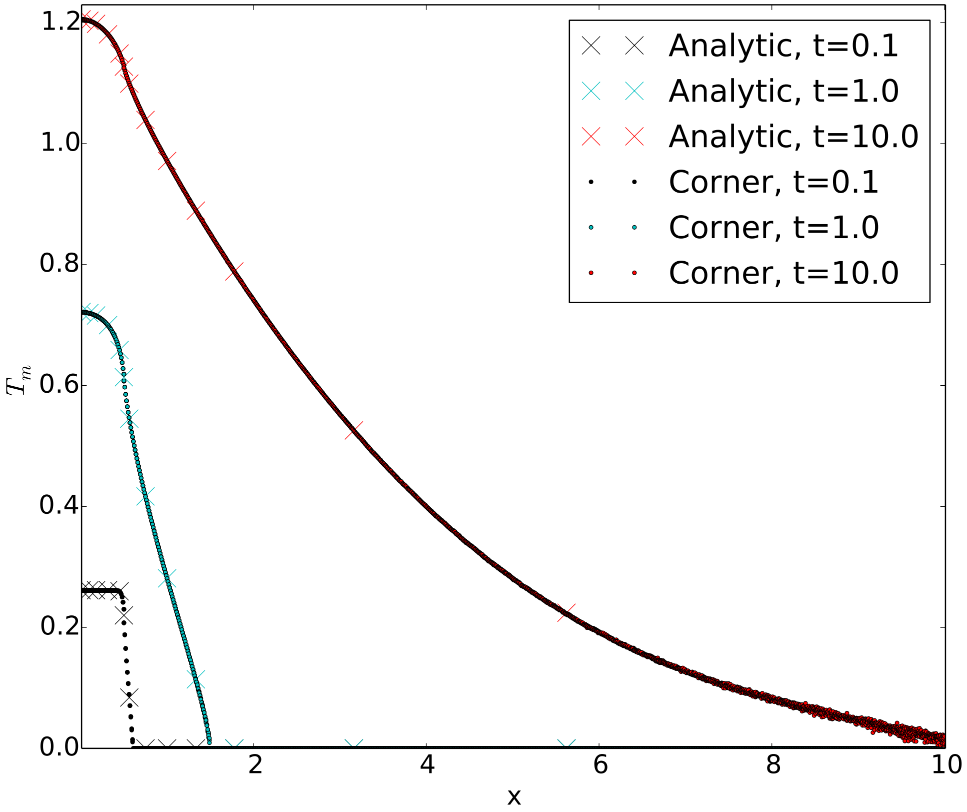


Figure 3.17: The material temperature in the Su-Olson problem with the corner-based IMC method at three times

than the peicewise-constant methods and thus experiencing more scattering events. This is due to the linear opacity representation and is explained further below and illustrated in Fig. (3.32).

Results for the crooked pipe problem for each method are shown in Figs. (3.22) to (3.24). The various methods in CHICOMA are compared to the 2D, RZ KULL results. The crooked pipe problem was run with various resolutions at the thick-thin material interface. The propagation of energy through the problem is sensitive to this material interface resolution, as is seen in the CHICOMA results. In 2D, it is relatively simple to increase the resolution at the material interface, in 3D selectively

Method	Runtime (s)		
	Su-Olson	Marshak Wave ($\Delta x = 0.0025$ cm)	Crooked Pipe ($\Delta x = 0.015$ cm)
Node-Centered (no upwind)	7469	2640	7200.526
Node-Centered (with upwind)	7461	2652	7888.19
Corner-centered	7559	2689	11717.55*
DFEM	7515	2864	11165.16*

Table 3.5: Runtimes for different methods on the Su-Olson and Marshak wave problems (Crooked Pipe Δx is the resolution at the material interface) (* these results were run with half as many cores on a slightly faster machine)

increasing the resolution at the material interface is more complicated. With standard meshing tools, the tetrahedra at the material interface can be uniformly refined, as shown in Fig. (3.25). Uniform refinement at the interface increases the number of mesh cells at the interface by a factor of four. These interface cells quickly dominate the number of total cells, as is seen in Table (3.6). This has two negative effects on the simulation:

- Particles cross relatively more cell boundaries and thus experience more discrete events before reaching the end of the timestep, which increases runtime
- To obtain the same relative quality (low variance) of the solution in the additional mesh cells, more particle histories must be used

An alternative meshing solution would be to increase the aspect ratio of the tetrahedra at the material interface. This would provide an increased resolution at the interface without significantly increasing the number of mesh cells. This type of meshing is used to resolve boundary layers when solving Navier-Stokes problems in the aerospace community [49]. This method of meshing an interface is called anisotropic grid stretching. Advanced mesh generation has not been used in CHICOMA simulations, but the simulation results and runtimes highlight the need for mesh generation

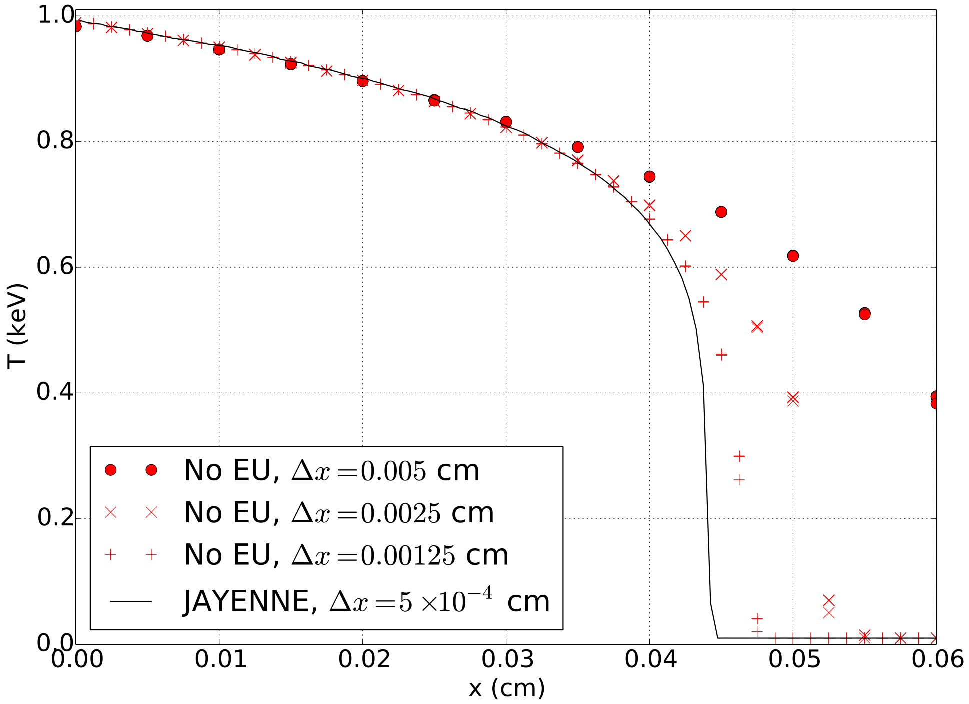


Figure 3.18: The node-centered IMC method without emission upwinding IMC method for the Marshak wave problem at $t = 1.0$ ns at various spatial resolutions

that can automatically capture material interfaces without significantly increasing the cell count.

3.7.1 Absorption Based Emission Results

Implementing the absorption based emission as the emission shape in CHICOMA yields results that behave more like S_N than IMC on Marshak wave type problems. Figs. (3.26) compares the convergence of the absorption based emission to the S_N results from the CAPSAICIN code at Los Alamos National Laboratory. The node-centered IMC method with standard tilting results are shown for comparison in Fig. (3.27). Comparing these figures shows that the fix up method allows less energy into the problem for all cases in a way that is similar to the S_N results. The results of the absorption based emission compared to resolved JAYENNE and CAPSAICIN

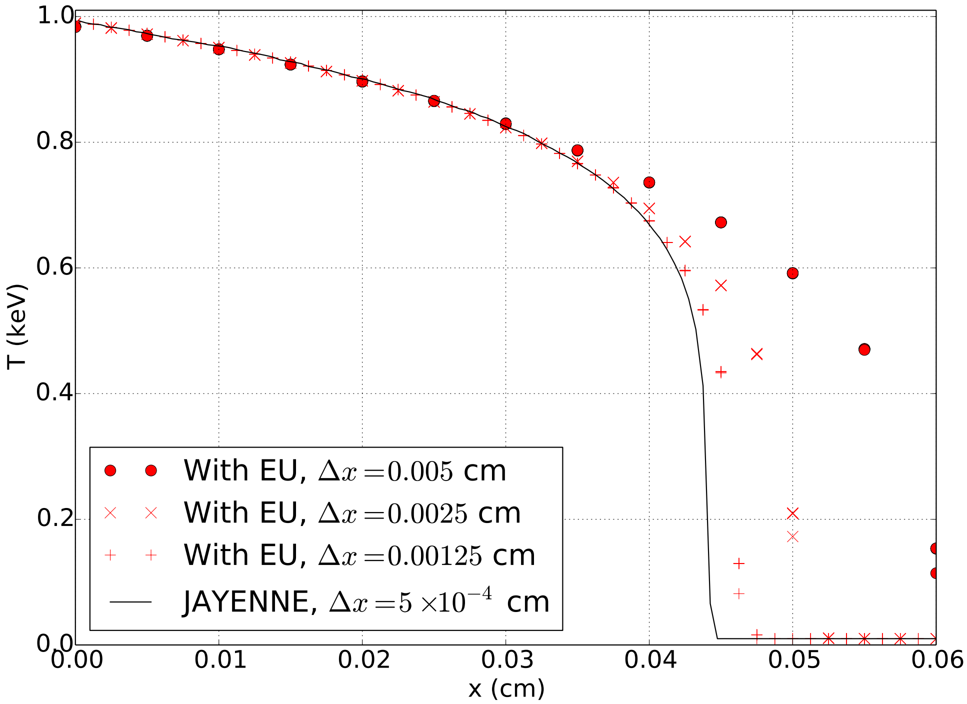


Figure 3.19: The node-centered IMC method with emission upwinding for the Marshak wave problem at $t = 1.0$ ns at various spatial resolutions

Interface Resolution	Unknown Count
$\Delta x = 0.03$	49724
$\Delta x = 0.015$	161875
$\Delta x = 0.0075$	608046
$\Delta x = 0.00375$	2387779

Table 3.6: Unknown count for each level of interface resolution in the crooked pipe problem

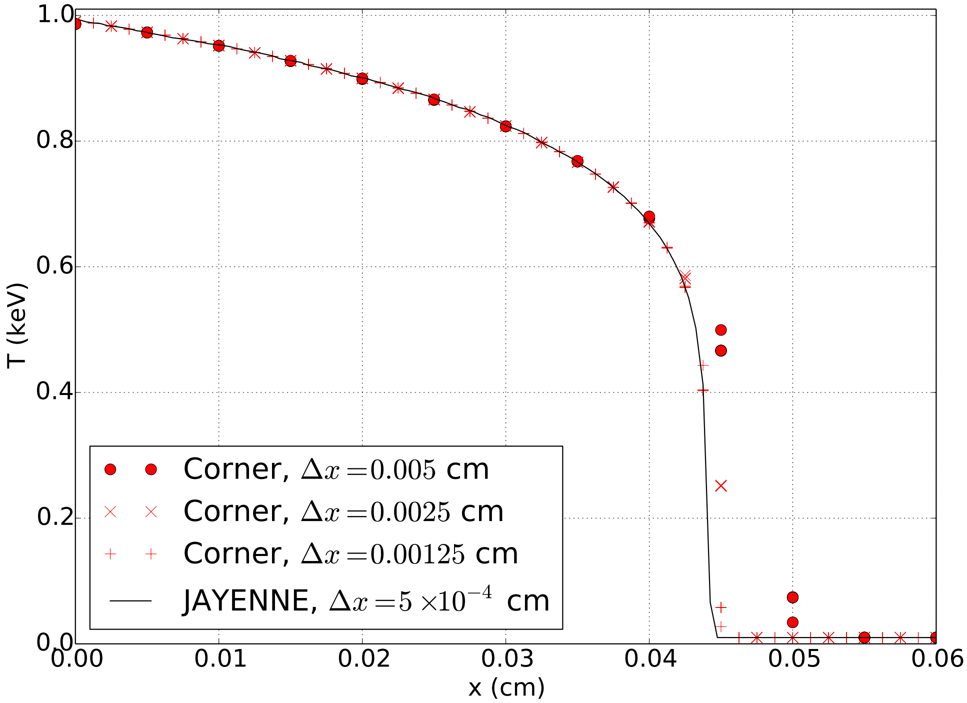


Figure 3.20: The corner-centered IMC method for the Marshak wave problem at $t = 1.0$ ns at various spatial resolutions

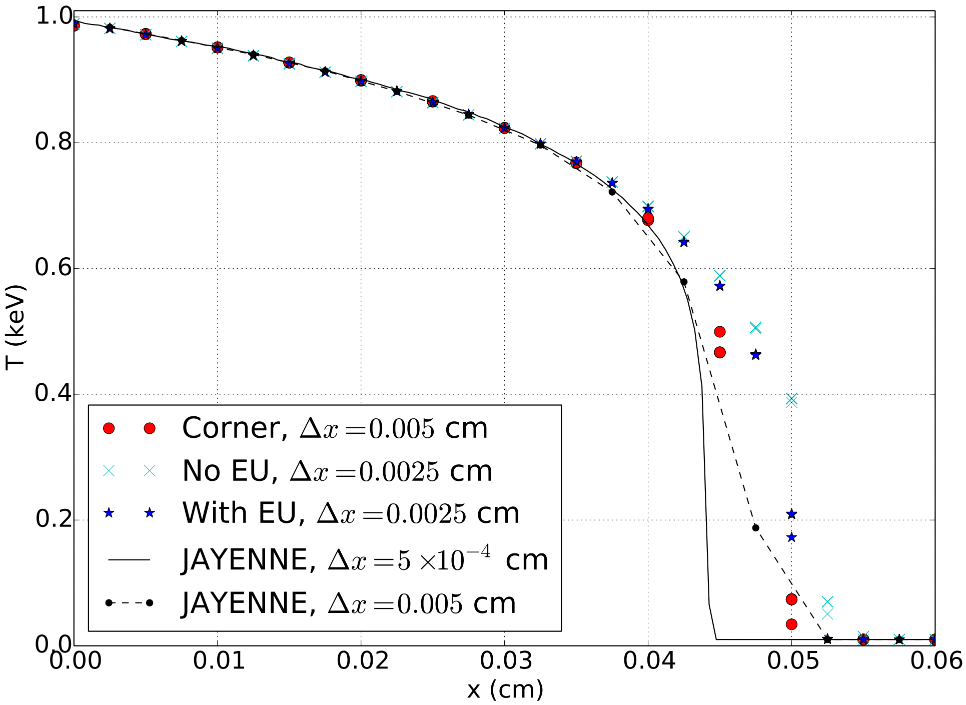


Figure 3.21: A comparison of the node-centered and corner-centered IMC methods for the Marshak wave problem at $t = 1.0$ ns

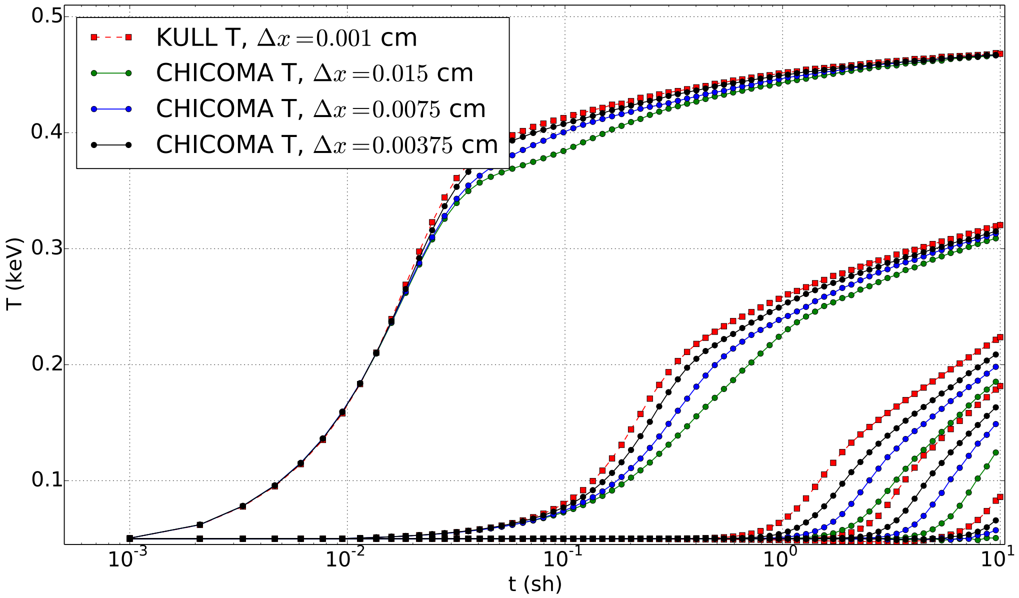


Figure 3.22: Temperature vs time for 5 tracked points at various mesh resolutions in the crooked pipe problem with the node-centered IMC method, KULL IMC is given as a reference solution

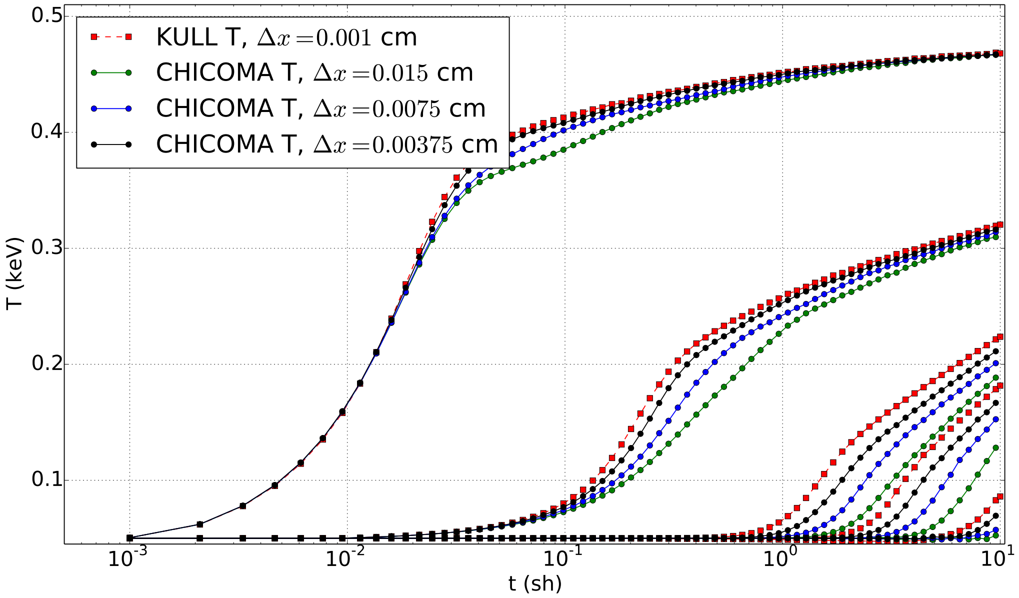


Figure 3.23: Temperature vs time for 5 tracked points at various mesh resolutions in the crooked pipe problem with the node-centered IMC method and emission upwinding, KULL IMC is given as a reference solution

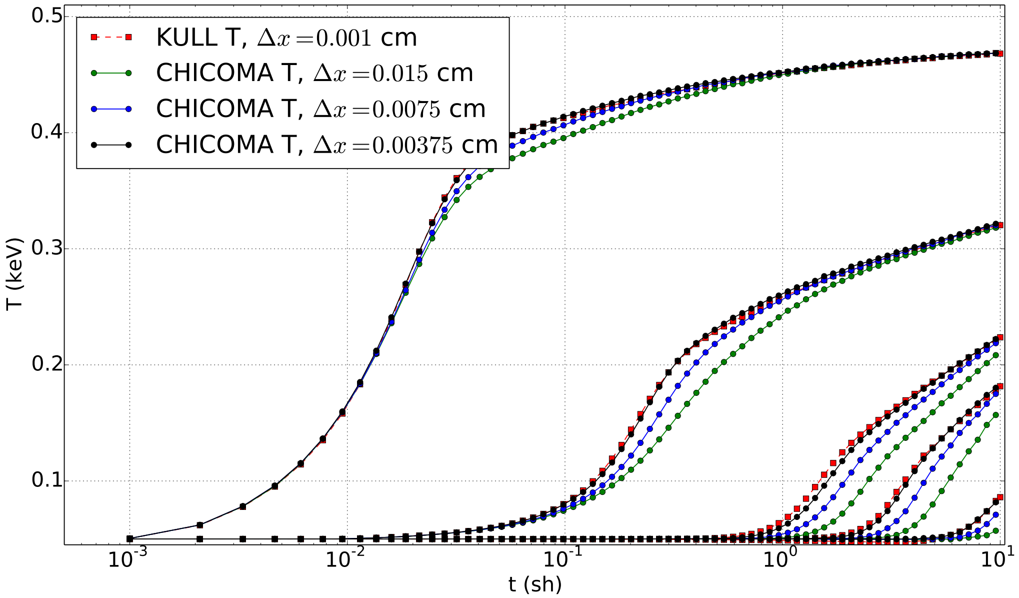


Figure 3.24: Temperature vs time for 5 tracked points at various mesh resolutions in the crooked pipe problem with the corner-centered IMC method, KULL IMC is given as a reference solution

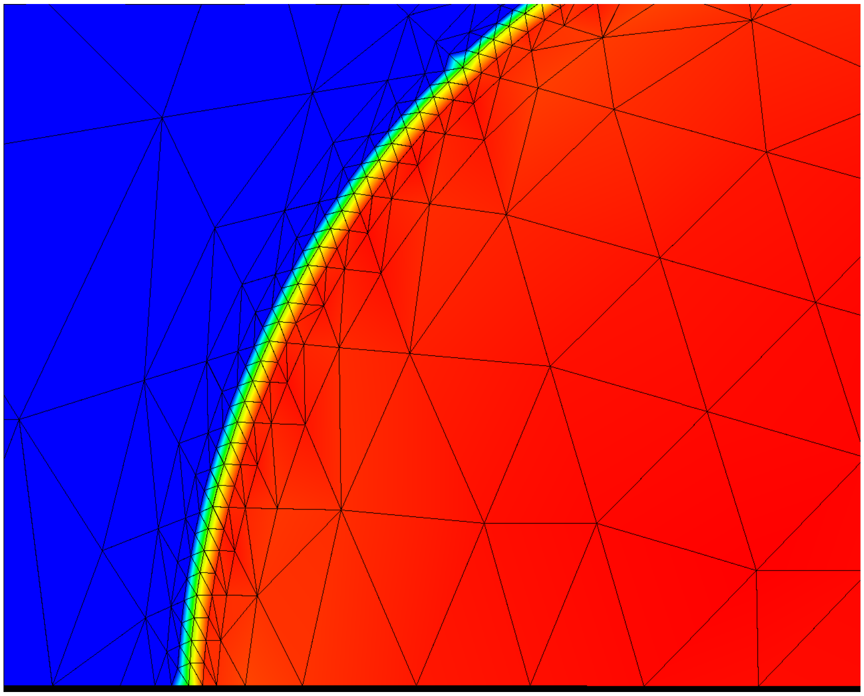


Figure 3.25: Resolution of the material interface in the crooked pipe problem run with the CHICOMA code–the interface is at $r = 0.5$ cm, the location of the smallest mesh cells (reprinted with permission from [37])

Δx_i	Standard					Emission Based Absorption				
	P_1	P_2	P_3	P_4	P_5	P_1	P_2	P_3	P_4	P_5
0.03	0.466	0.304	0.165	0.092	0.049	0.466	0.306	0.169	0.100	0.050
0.015	0.466	0.309	0.188	0.128	0.052	0.467	0.311	0.190	0.133	0.051
0.0075	0.467	0.313	0.200	0.152	0.057	0.467	0.315	0.204	0.155	0.060
0.00375	0.466	0.316	0.211	0.166	0.069	0.467	0.317	0.213	0.170	0.073

Table 3.7: Temperature of the five tracked points for the standard node-centered IMC method and the absorption based emission IMC method at various interface resolutions (Δx_i, in centimeters)

are shown Fig. (3.28). Again, the wavefront for the fix up method is closer to the S$_N$ solution than the resolved IMC solution.

The crooked pipe problem has also been rerun to examine how absorption based emission changes the results. At the same mesh resolution this method gives higher temperatures than the node-centered method. The higher temperatures are closer to the much more resolved RZ results from the KULL IMC code at Lawrence Livermore National Laboratory. This is consistent with the results observed in the Marshak wave problem—less energy penetrates into thick regions, meaning more energy is reflected back into the thin region, producing higher material temperatures at the tracked points. Visually, the differences between the temperatures are difficult to discern so the material temperature at the end of the simulation for each tracked point are instead listed in Table (3.7).

3.7.2 DFEM IMC

To study the DFEM IMC method, we first examine the use of different opacity treatments in a 1D code. This is done to isolate the effects of an unstructured grid from the effects of the DFEM IMC and the opacity treatments. In 1D, the mass matrix, $\mathbf{M}_\mathrm{t}$, is:

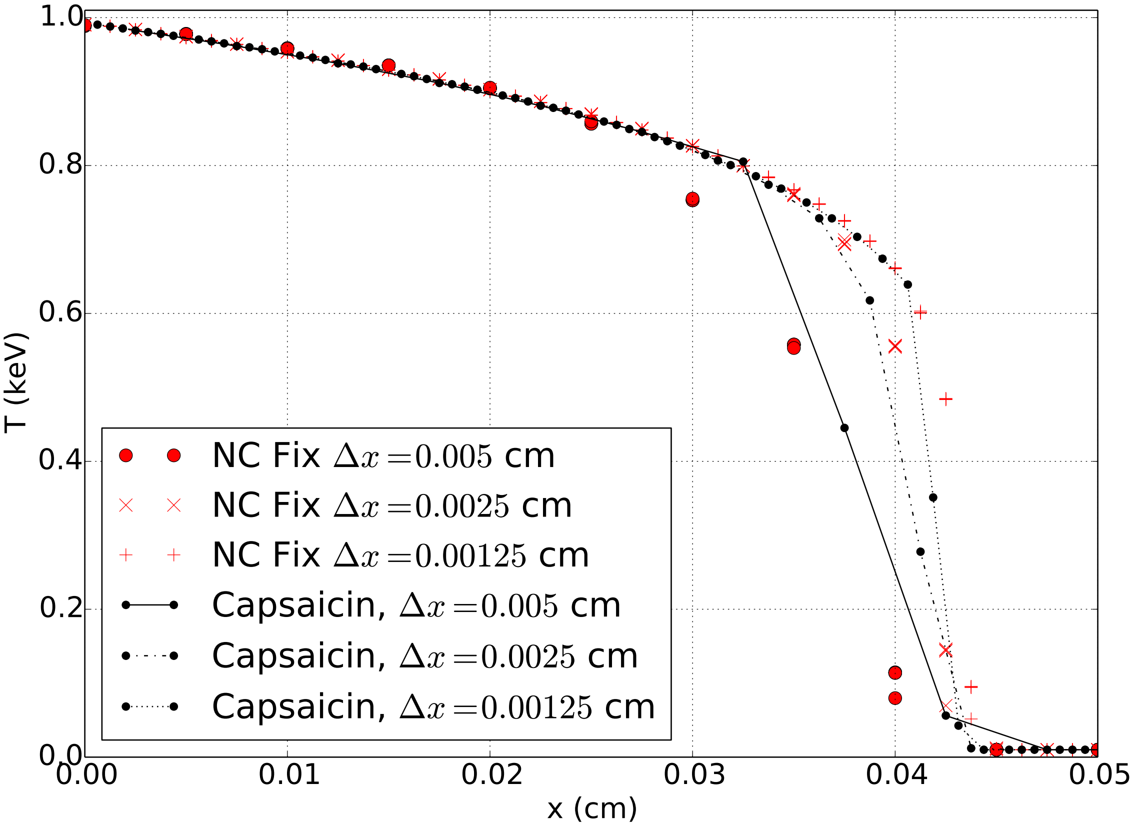


Figure 3.26: Node-centered IMC with absorption based emission and CAPSAICIN S_N results at 1 ns for various mesh resolutions

$$\mathbf{M}_{\mathrm{t}} = x_{\mathrm{c}} \begin{pmatrix} 1/3 & 1/6 \\ 1/6 & 1/3 \end{pmatrix} \tag{3.85}$$

Where x_{c} is the length of the 1D cell. The results for each opacity treatment in the DFEM IMC method on the Marshak wave problem are compared to the standard IMC method in the JAYENNE code. The case where opacity is simply averaged, as in Eq. (3.79), yields negative solutions on the Marhsak wave problem and is therefore not included in the results. Each method requires lumping the mass matrix in front of the time derivative and in the emission term to ensure positivity—this is due to

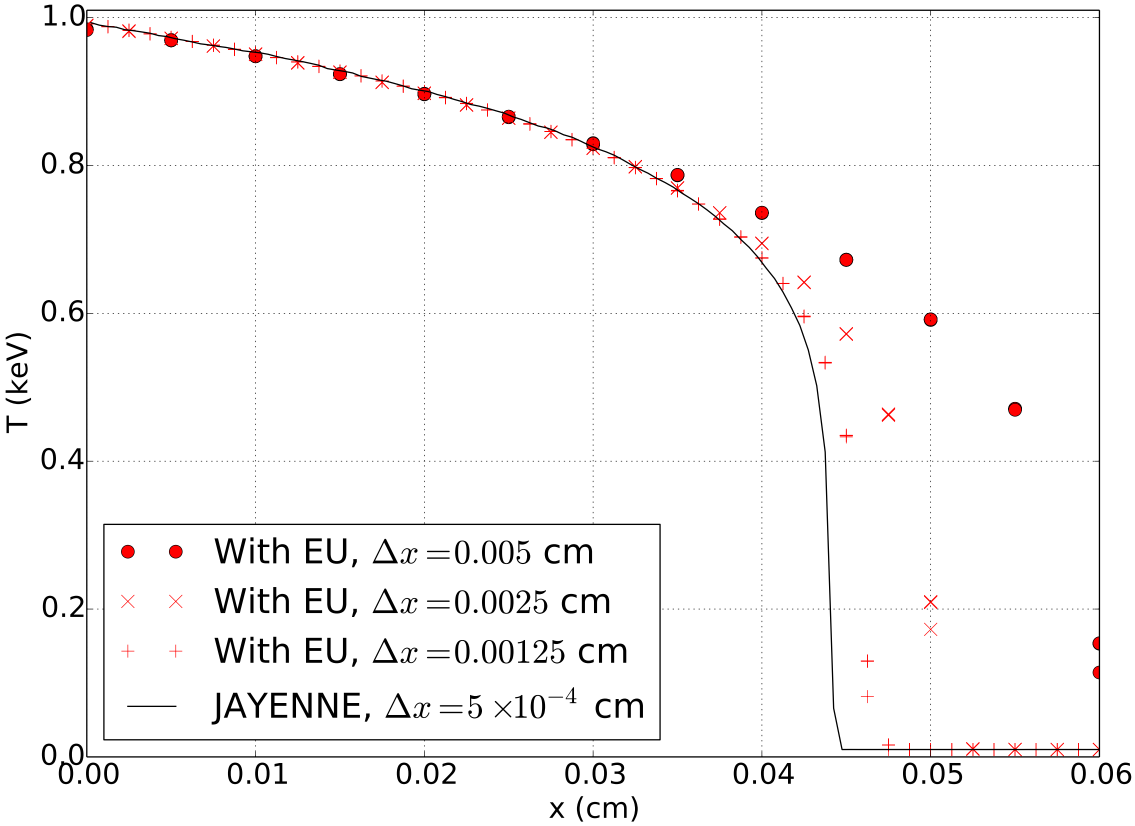


Figure 3.27: Node-centered IMC results at 1 ns for various mesh resolutions

the fact that the temperature at the wavefront cannot be fit with a linear function. The DFEM IMC solution is shown with the full discontinuous solution and the cell average solution. Figures (3.29) to (3.30) shows the results for each case. A feature of using piecewise linear opacity not seen in the late time results is that the material temperature can be larger than the source temperature incident on the problem. This effect is known as a *maximum principle violation* [58] and it is a well known occurrence in IMC methods. It is essentially non-monotonic behavior of the solution about some equilibrium condition and appears when too much energy is absorbed during a timestep[3]. This occurs in the linear opacity case of DFEM IMC because

[3] this energy absorption can be controlled by reducing the timestep size or increasing the size of

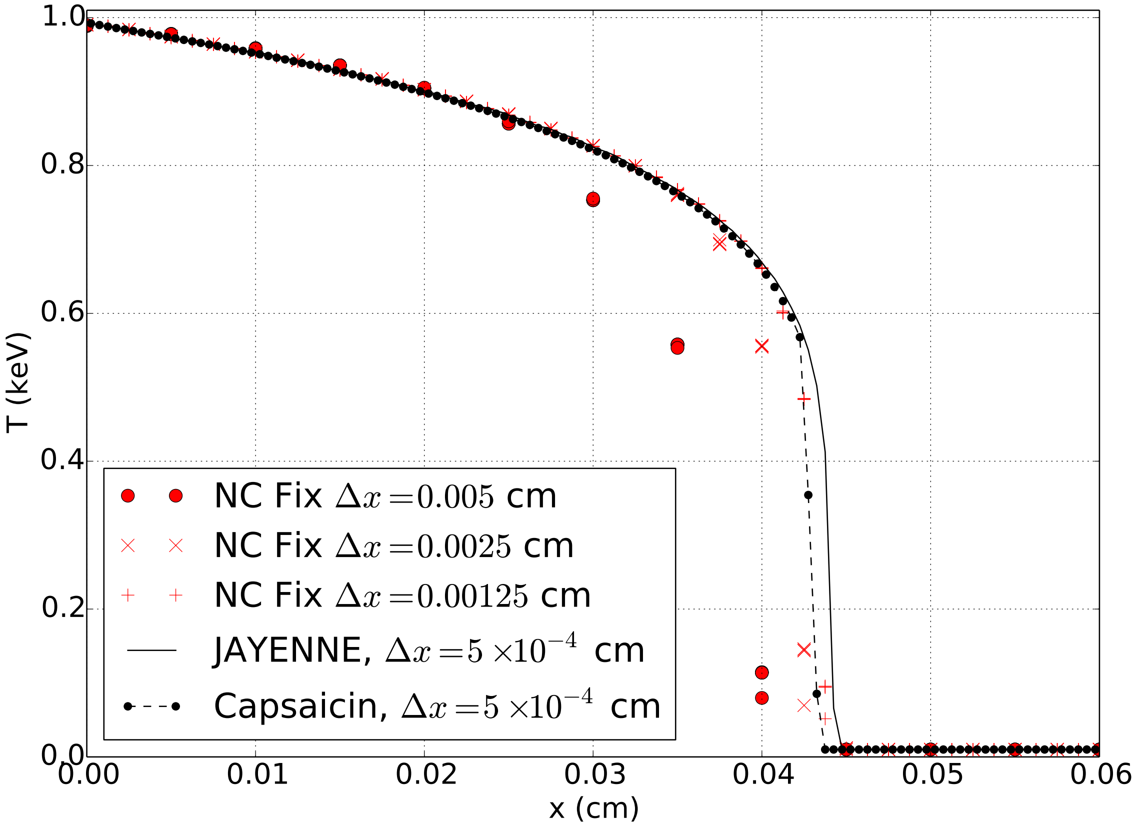


Figure 3.28: Node-centered IMC results at 1 ns compared to refined results from CAPSAICIN and JAYENNE

applying a linear shape to the opacity yields a larger opacity than the standard IMC method for the same number of unknowns and thus more energy is absorbed compared to the standard IMC method. This situation is illustrated in Fig. (3.32). Figure (3.33) shows the peaks in the temperature solution for DFEM linear opacity and how these oscillations are damped out over time.

The results from the linear test case show that linear opacity case converges to a value that is ahead of the piecewise constant IMC method solution. This could be due to the overheating that occurs at the early timesteps. The opacity averaging

spatial cells

methods both converge to the piecewise constant IMC solution, but they do not converge at an accelerated rate—for the same level of grid spacing there are twice as many unknowns in the DFEM method so in reality both of these cases are worse than the piecewise constant method. These results indicate that more work is necessary on developing a DFEM IMC solution that is both robust and accurate. Alternative means of lumping the mass matrix may improve the quality and robustness of the solution, as is done by Maginot et al [39].

The results of the 1D DFEM IMC code eliminate the use of simple averaged opacity. The DFEM IMC method could still prove advantageous to the IMC implementation in CHICOMA because using a constant or linear opacity within a tetrahedral cell eliminates the need to track to the corner boundary. The DFEM IMC method with linear opacity was tested for the Su-Olson and Marshak wave problems, the results are shown in Figs. (3.34) to (3.35). The constant opacity with an average temperature is shown in Fig. (3.36). The constant opacity with the average temperature to the fourth is shown in Fig. (3.37). The Marshak wave results appear similar to the 1D case: the wavefront in the full linear opacity DFEM IMC method converges to a position ahead of the standard IMC wavefront. The averaged opacity results both converge to the refined, standard IMC wavefront. Although it is not clearly visible in the graph, the constant opacity with T^4 averaging has a slightly higher temperature for all points compared to the T averaged opacity. This is expected because a T^4 average is higher, yielding an overall lower opacity. The performance of these opacity averaged DFEM IMC results indicates that an IMC method could be used on the CHICOMA mesh that does not need to track to the interface of the dual mesh cells.

3.8 Discussion

The corner-centered IMC method appears to be the best IMC method for the unstructured tetrahedral mesh with node-based unknowns. It produces more accurate results compared to the node-centered method in all cases. The DFEM IMC method shows promise when used with average opacity—this method converges to the same solution as the corner-centered method and does not require tracking to the dual mesh cell interface. A DFEM IMC that uses the linear temperature shape for opacity and emission and that has improved convergence behavior has not been observed in the results and requires more development.

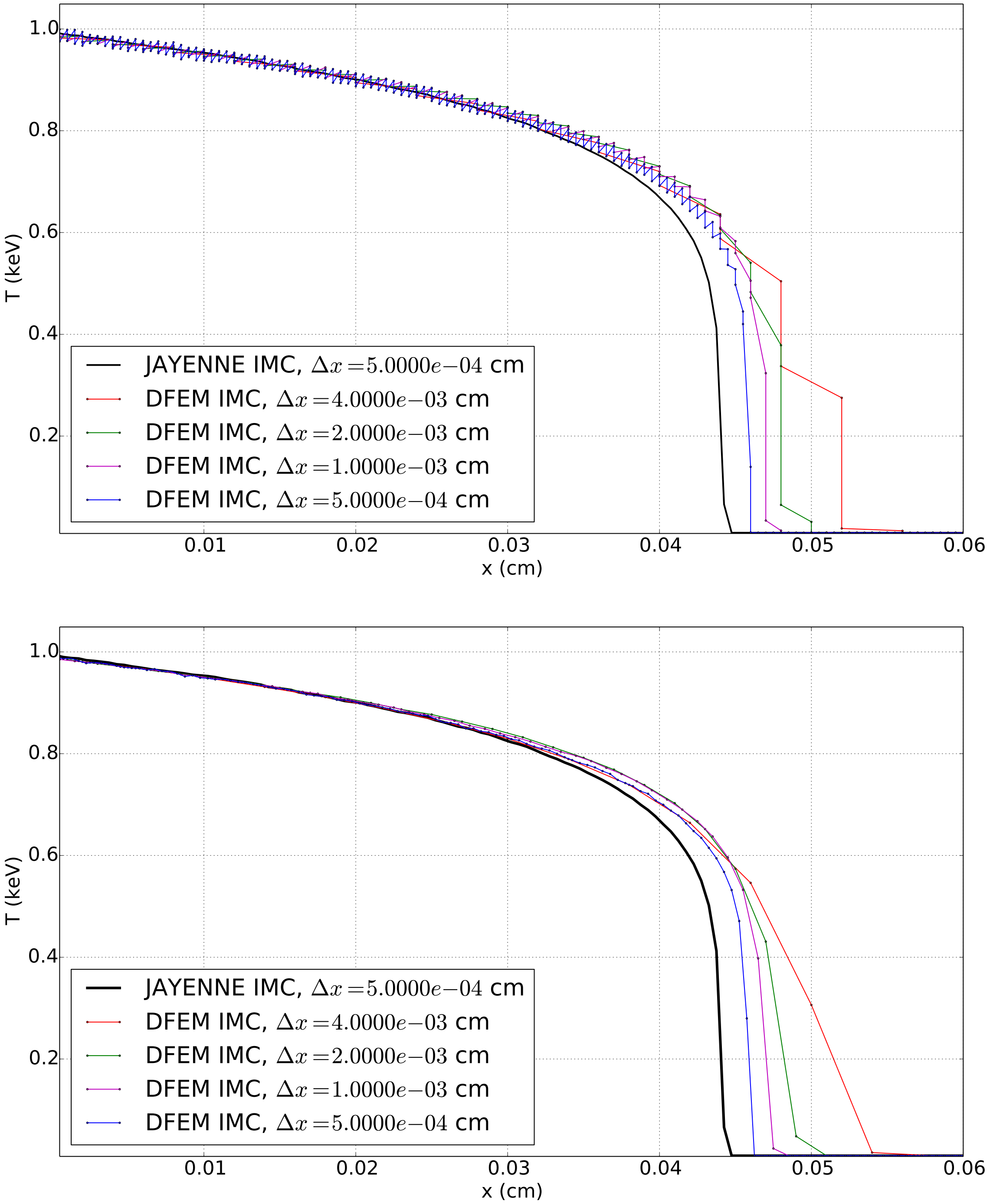


Figure 3.29: Discontinuous and average temperatures for the 1D DFEM IMC method with linear opacity within a mesh cell for the Marshak wave problem at $t = 1.0$ ns compared to the JAYENNE solution

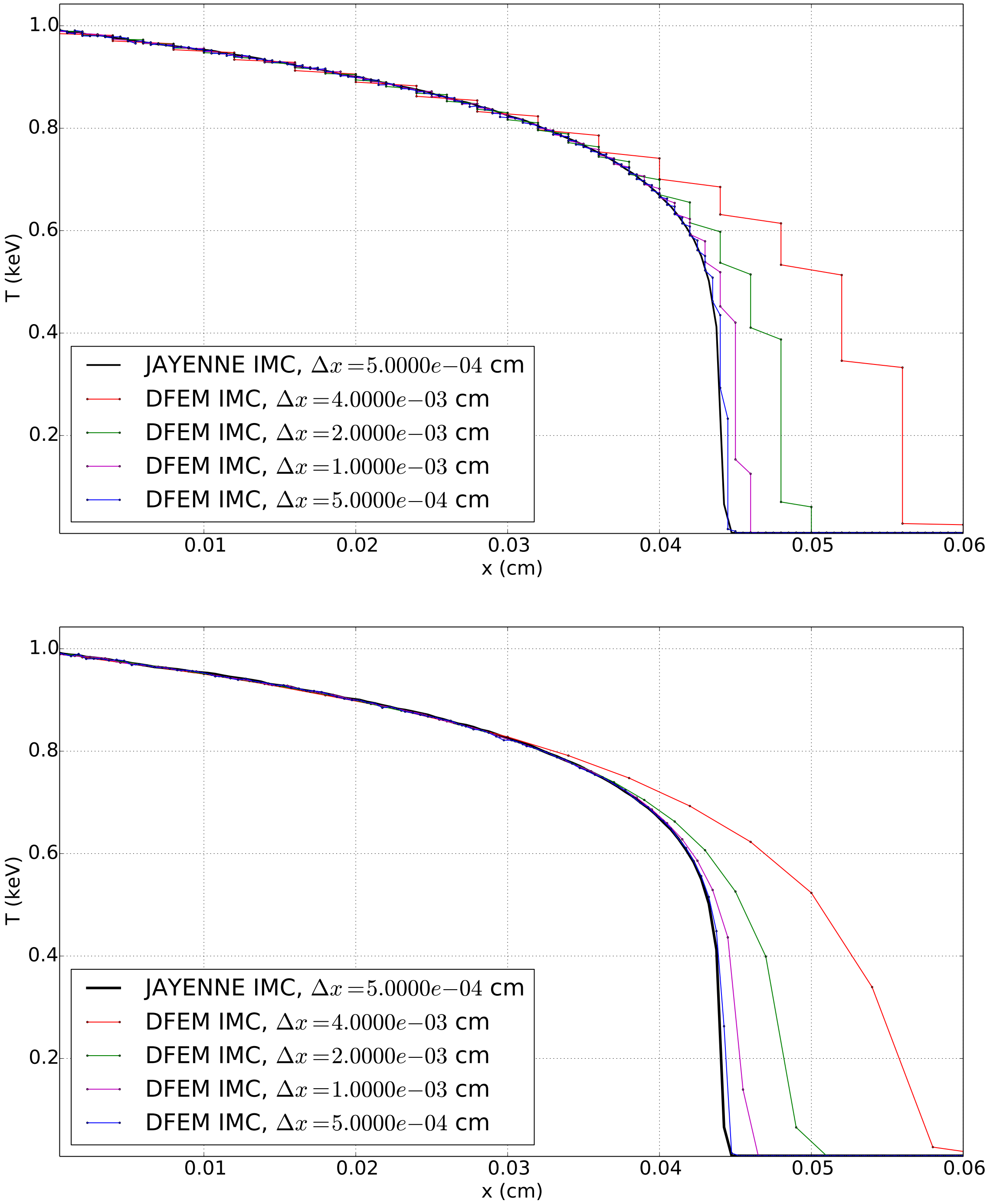


Figure 3.30: Discontinuous and average temperatures for the 1D DFEM IMC method with average temperature opacity for the Marshak wave problem at $t = 1.0$ ns compared to the JAYENNE solution

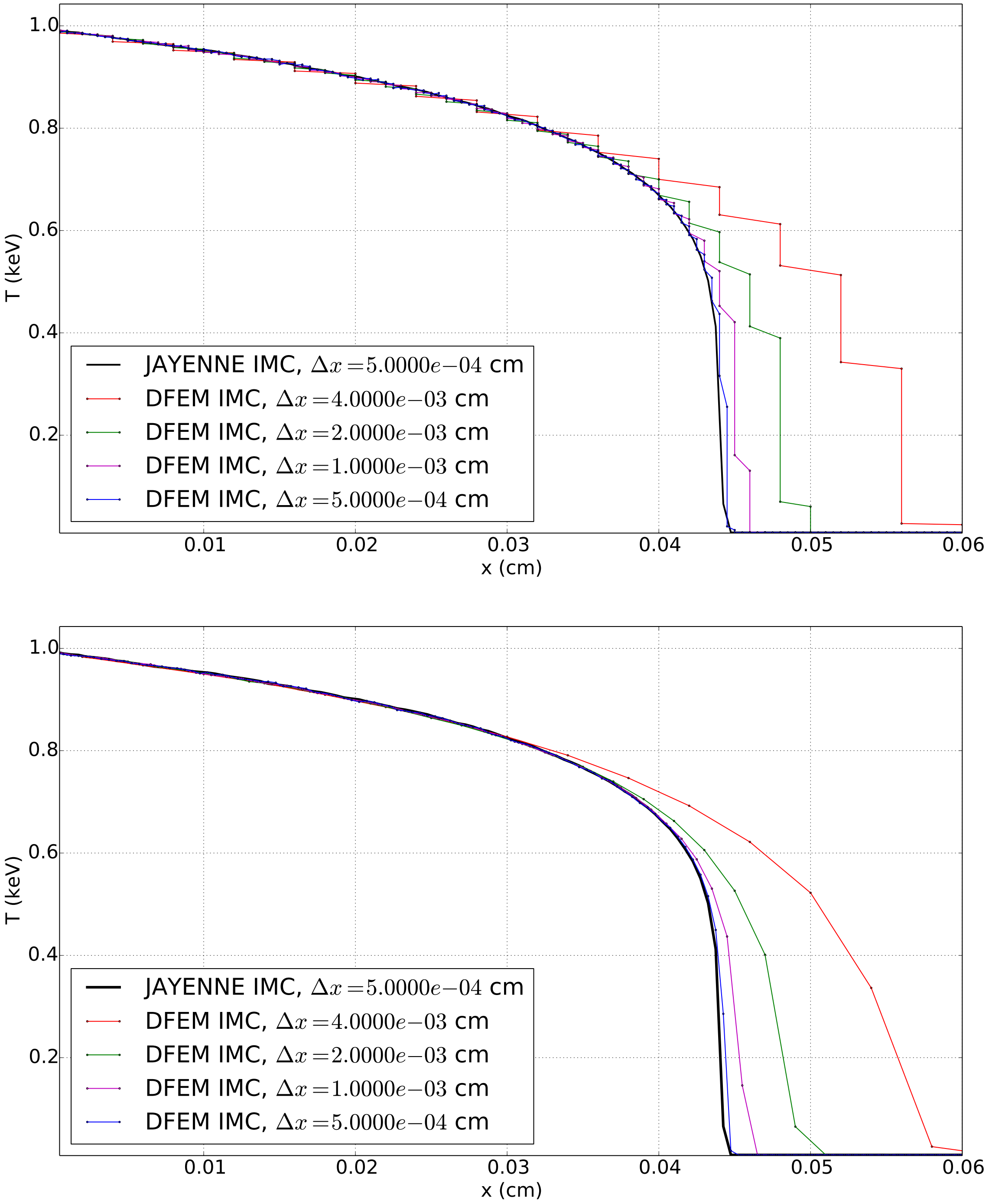
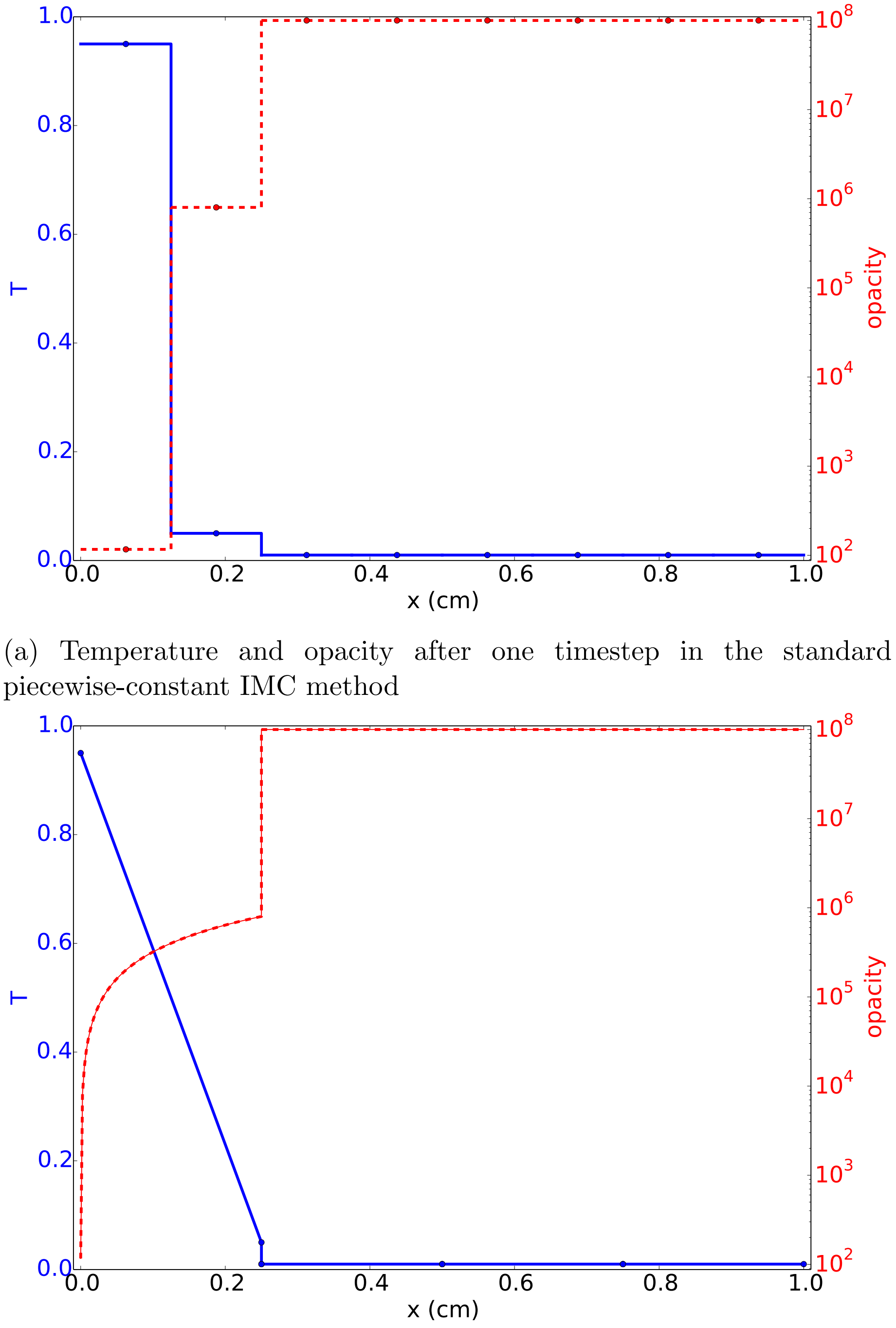


Figure 3.31: Discontinuous and average temperatures for the 1D DFEM IMC method with average T^4 opacity for the Marshak wave problem at $t = 1.0$ ns compared to the JAYENNE solution



(a) Temperature and opacity after one timestep in the standard piecewise-constant IMC method

(b) Temperature and opacity after one timestep in the DFEM IMC method

Figure 3.32: A comparison of the opacity and temperature at the wavefront in a Marshak wave problem—the DFEM method effectively has a larger opacity than the IMC method (the opacity is linear within an element in the DFEM method but the plot scale is logarithmic so the shape is curved)

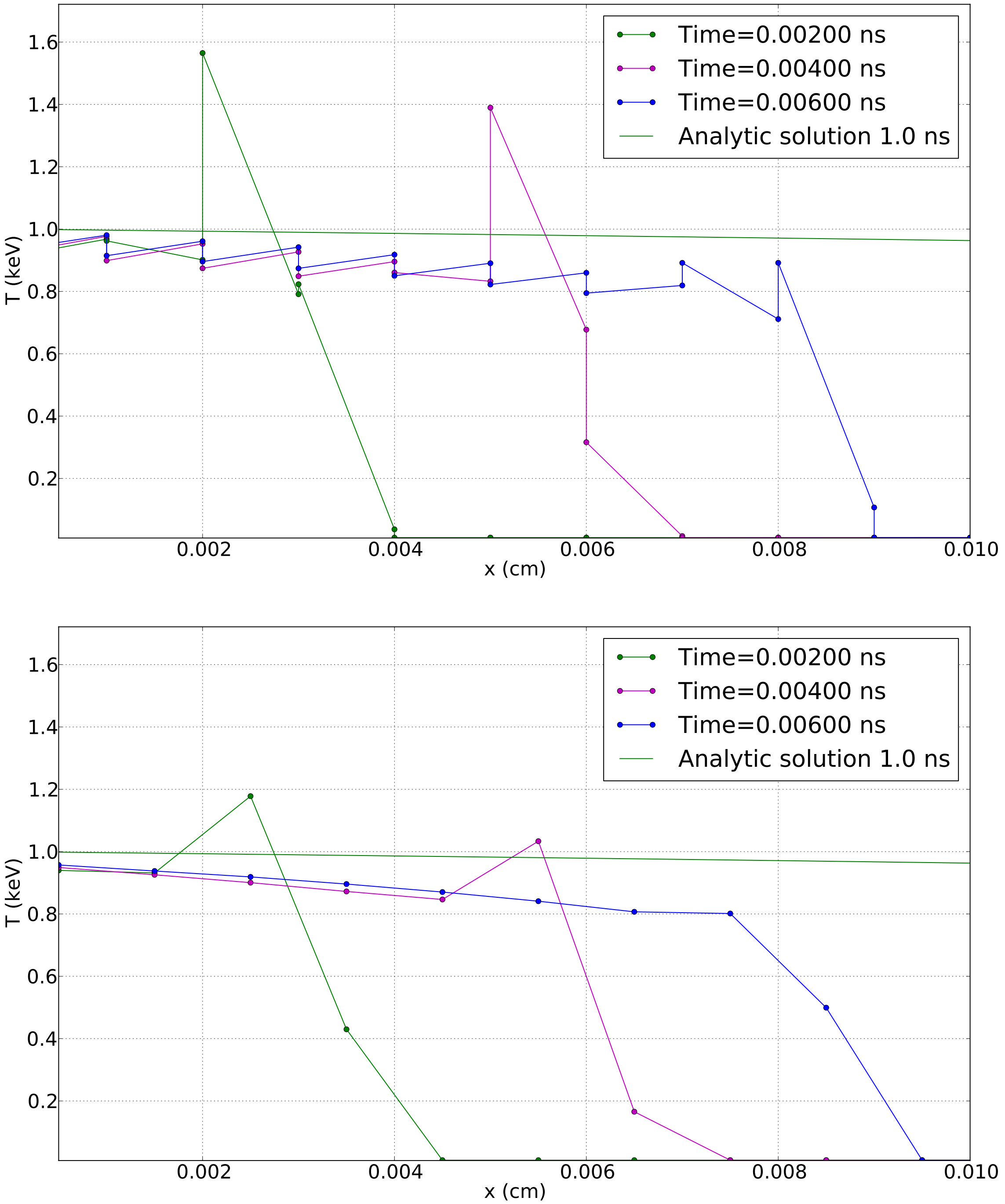


Figure 3.33: Damped maximum principle violations in the solution of the DFEM IMC equations with linear opacity at several timesteps

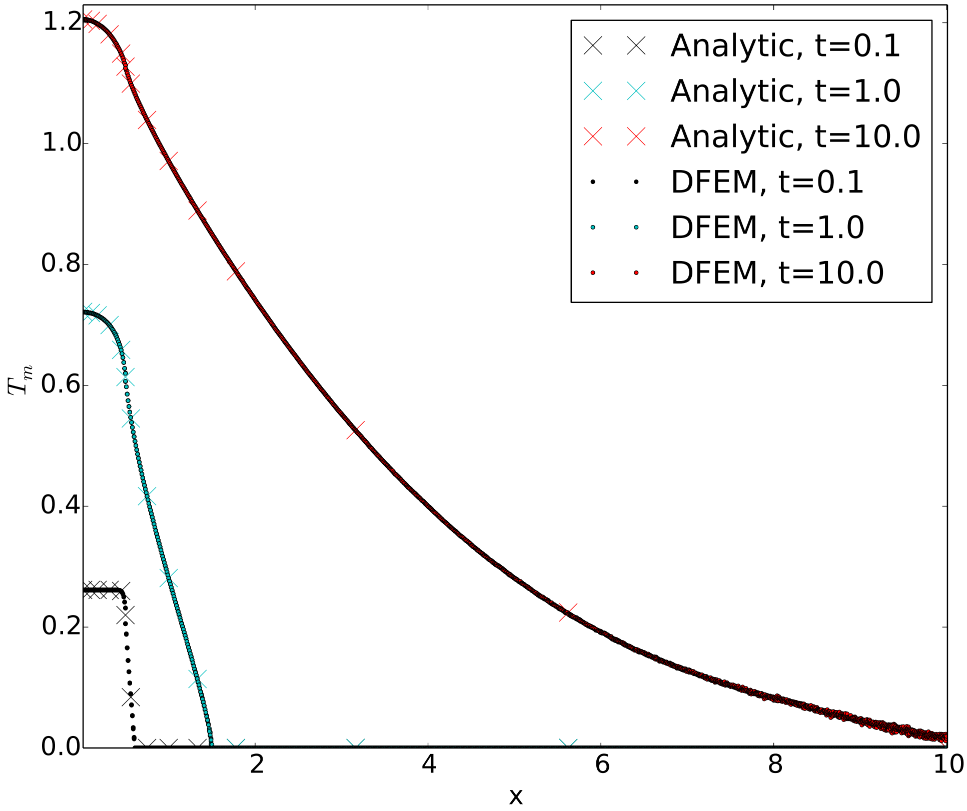


Figure 3.34: The material temperature in the Su-Olson problem with the corner-based IMC method at three times

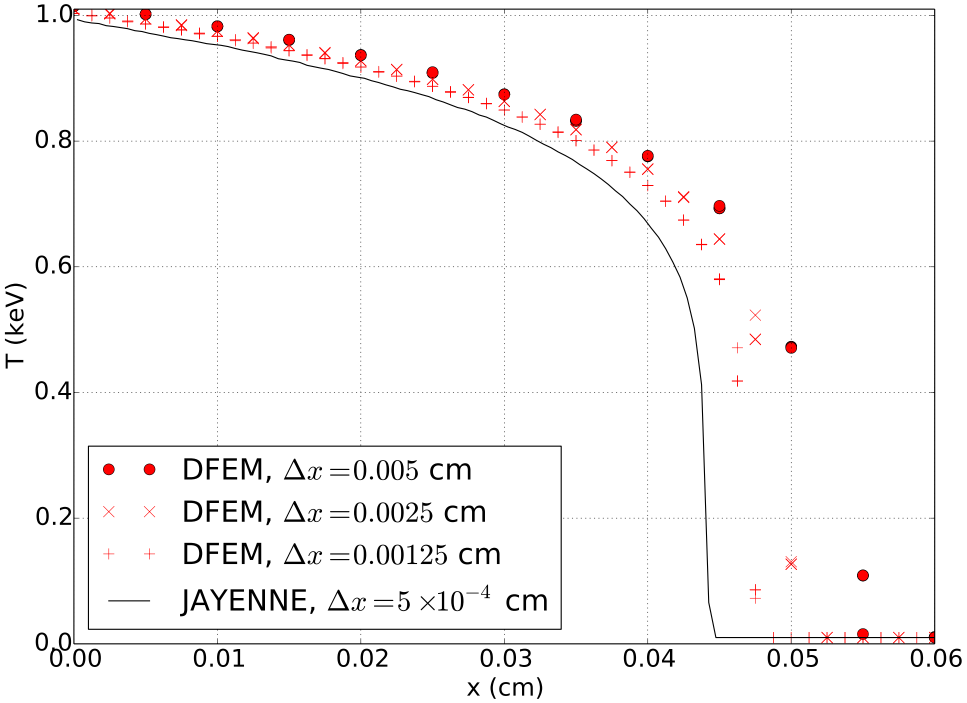


Figure 3.35: Temperature vs. time for a Marshak wave problem with the DFEM IMC method with linear opacity

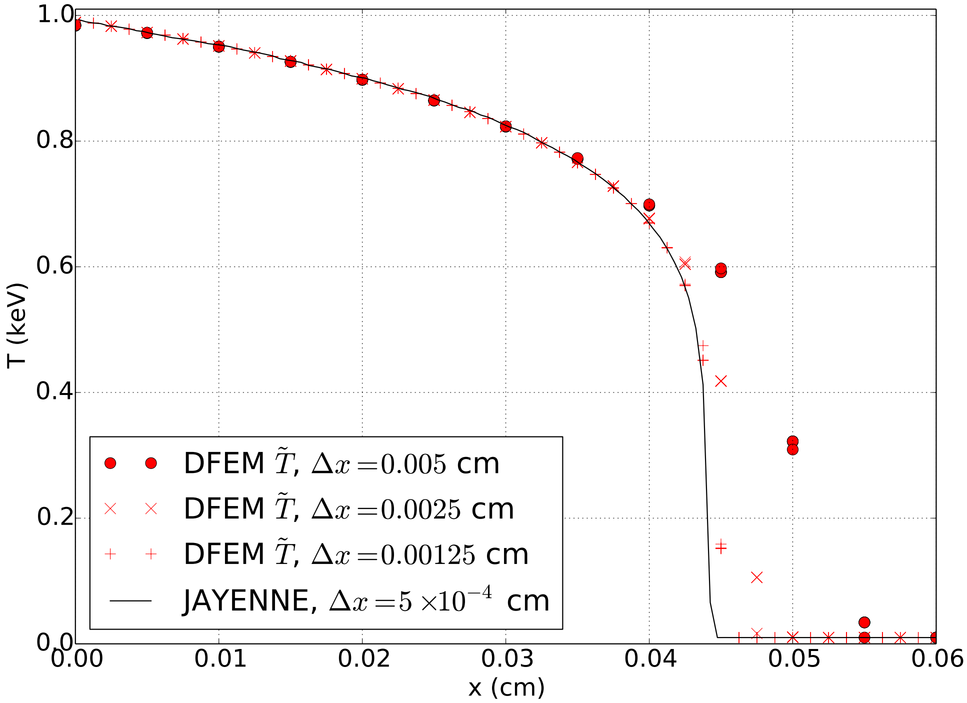


Figure 3.36: Temperature vs. time for a Marshak wave problem with the DFEM IMC method with a constant opacity from the average T in an element

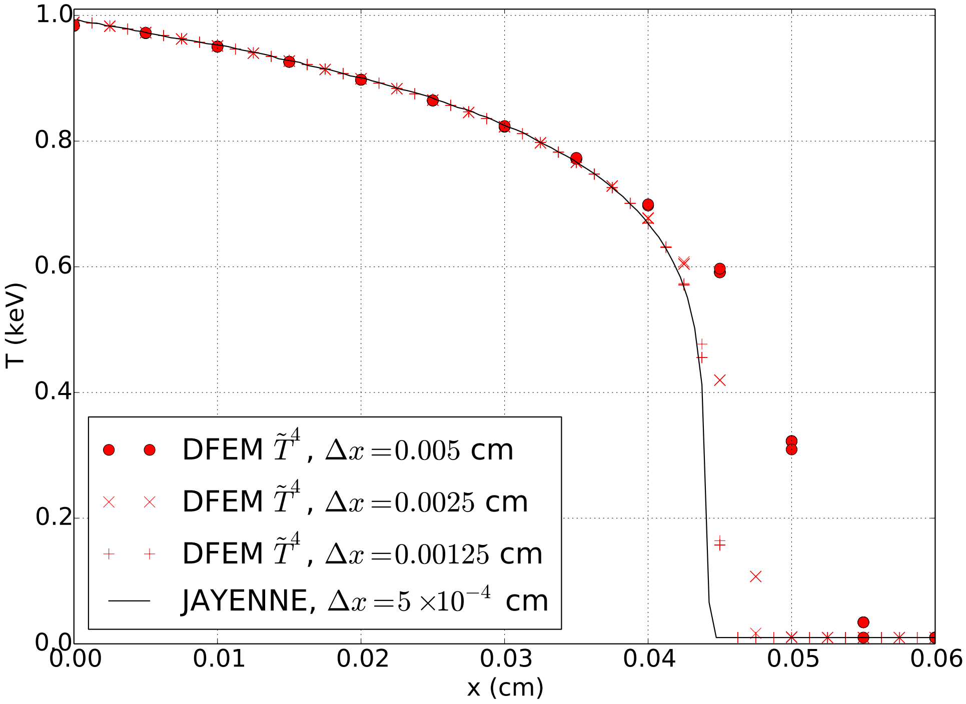


Figure 3.37: Temperature vs. time for a Marshak wave problem with the DFEM IMC method with a constant opacity from the average T^4 in an element

4. SAMPLING ON TETRAHEDRA AND CORNERS

4.1 Uniform Sampling

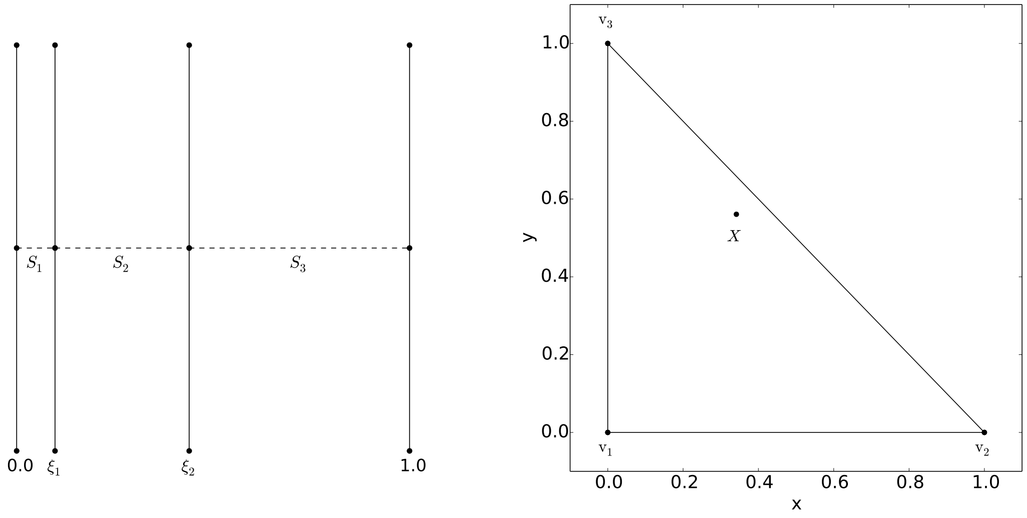
Sampling a particle location in an orthogonal quadrilateral or hexehedral mesh cell is done by sampling each coordinate independently. This is more difficult on triangular and tetrahedral meshes. Both triangles and tetrahedra are *simplexes.* A simplex is defined as a shape in d dimensions formed by $d+1$ points. Points within a simplex can be described with barycentric coordinates. The formula:

$$\mathbf{X} = \sum_{i=1}^{d+1} \lambda_i \mathbf{v_i}, \tag{4.1}$$

describes all points within a simplex. Where the spacing vector, λ, is a positive unit vector and $\mathbf{v_i}$ is the position of vertex i that forms the simplex and $\mathbf{X}$ is a location within the simplex. Devroye gives a method to sample the spacing vector, λ, such that a uniform distribution of points is produced within the simplex [16]. The spacing vector is obtained by generating a vector of random numbers, ξ of length d where each $\xi_i \in [0,1]$. These d random numbers are then sorted least to greatest and the spacing vector is determined with:

$$\lambda_i = \begin{cases} \xi_1, & i = 1 \\ \xi_{i+1} - \xi_i, & d+1 > i > 1 \\ 1 - \xi_d, & i = d+1 \end{cases} \tag{4.2}$$

The relationship between ξ, λ and $\mathbf{X}$ is shown for the case of triangle in Fig. (4.1a) and Fig. (4.1b). This method samples each component of λ from an exponential distribution, which is necessary for the process to be *memoryless.* If the samples



(a) Spacing produced by two random numbers, ξ_1 and ξ_2

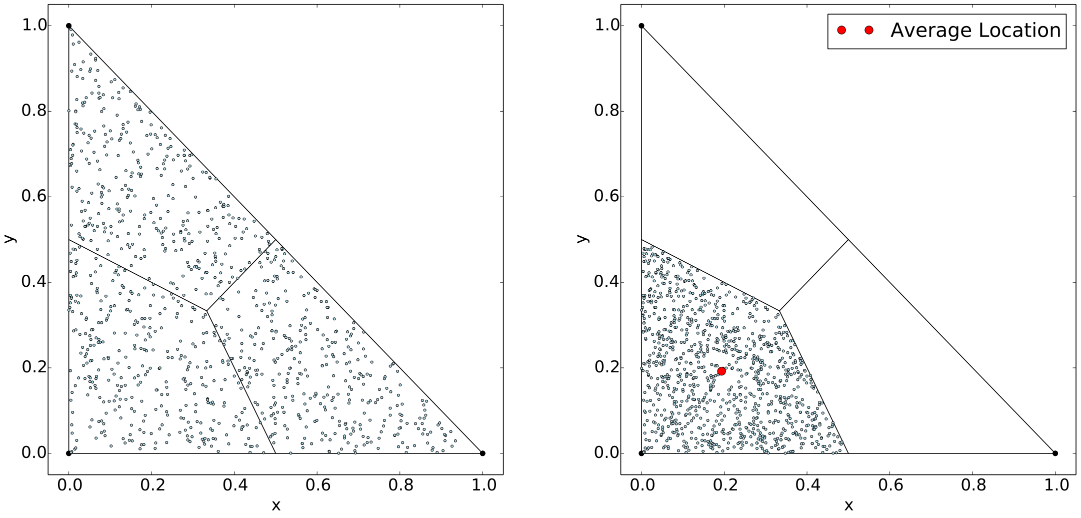
(b) Point, **X**, generated with the spacing in Fig. (4.1a)

Figure 4.1: Uniform sampling on a triangle using two random numbers (reprinted with permission from [37])

do not come from an exponential distribution, the points are clustered around the barycenter of a tetrahedron. Uniform particle locations on the tetrahedron can be sampled easily and without rejection by using Eq. (4.2) with Eq. (4.1).

Because energy is decomposed on to corners, particle emission locations need to be sampled on the corners as well. This can be done with rejection sampling by only accepting positions that are within the corner but this means on average three quarters of all samples are rejected. Instead we sample uniformly on the tetrahedron and translate positions outside the current corner onto the corner that is being sampled. In barycentric coordinates this can be done by multiplying the spacing vector λ by a transformation matrix, T:

$$\lambda_{\mathbf{i}} = \mathbf{T_i}\overline{\lambda_{\mathbf{i}}}. \tag{4.3}$$



(a) Uniform sampling on a triangle

(b) Points from Fig. (4.2a) transformed to corner 1

Figure 4.2: Uniform sampling within a triangle transformed to uniform sampling within a corner (reprinted with permission from [37])

Where $\overline{\lambda_{\mathbf{i}}}$ is the original sampled location in barycentric coordinates, and the transformation matrix, $\mathbf{T_i}$, depends on the corner that the point is in. In practice this equates to swapping the index of the maximum $\overline{\lambda_{\mathbf{i}}}$ value with the index of the corner you are sampling in. This method gives the same mean particle position as the rejection method. In a triangle, transforming from a point in corner 2 to corner 1 can be done with the transformation matrix:

$$T_{2,1} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} . \tag{4.4}$$

Transformations on a triangle are shown in Fig. (4.2).

4.2 Sampling from a Distribution

Particle emission is a strong function of temperature. Although the temperature is assumed constant in the standard IMC method, an emission shape is constructed from the emission in adjacent cells. This process is called *tilting* and is used to reduce spatial discretization error. To sample from this constructed emission distribution in orthogonal quadrilateral or hexehedral mesh cells is simplified by making the approximation that each dimension can be treated independently. This is more challenging on simplexes because each dimension cannot be treated independently. The rejection method can be used to sample particle locations from a linear emission distribution. This is done by probabilistically sampling particle position from a uniform distribution and accepting that position based on the emission, E, at that location:

$$P(x) = \frac{E(x)}{E_{\max}} \tag{4.5}$$

Where P is the probability of accepting the particle emission location. This method is implemented in three steps:

1. A particle position is sampled from a uniform distribution
2. $\frac{E(x)}{E_{\max}}$ is calculated at the sampled position
3. A random number, $\xi \in (0, 1)$, is compared to P—if ξ is less than P the point is accepted, otherwise start again at step 1

Emission locations using the rejection method on a triangle are shown in Fig. (4.3). This method can be expensive because it involves generating additional random numbers but using Eq. (4.3) can reduce some of the cost because it can be used

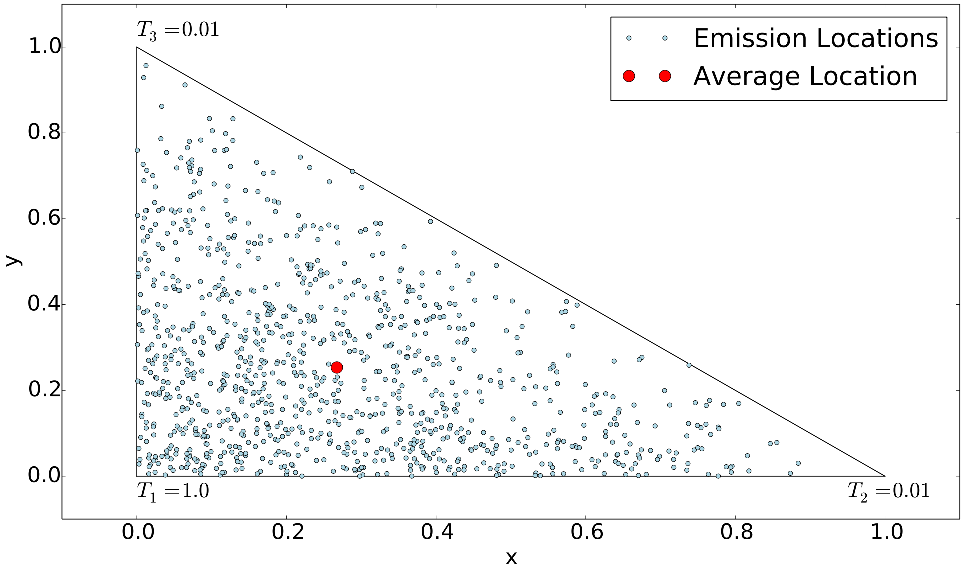


Figure 4.3: One thousand sampled particle emission locations with the rejection tilt method given a linear temperature field (reprinted with permission from [37])

to sample only within the corner of interest (although this makes calculating E_{max} slightly more complicated because it is not always simply the maximum E_i at the vertices). Tilting using node based values also differs from standard IMC algorithms because there is no ambiguity about how to construct the emission field because the temperatures are located at the vertices of a tetrahedron. The emission at any point within a tetrahedron can be calculated using the barycentric coordinate system:

$$E(\lambda) = \sum_{i=1}^{d+1} \lambda_i E_i = \sum_{i=1}^{d+1} \lambda_i \sigma_i T_i^4. \tag{4.6}$$

Where T_i is the material temperature at vertex i and σ_i is the opacity at vertex i. Equation (4.6) is equivalent to using linear finite element basis functions to represent the temperature, although only the DFEM IMC method explicitly treats the temperature as being linear during transport.

As an alternative to rejection sampling, each barycentric coordinate can be sampled using conditional probability. In conditional probability, the first barycentric coordinate is sampled from a PDF formed from Eq. (4.6) and subsequent barycentric coordinates depend on the previous sampled results. The CDF for each barycentric coordinate using conditional probability is:

$$\begin{aligned}\xi_1 = \frac{1}{\frac{E_1}{24}+\frac{E_2}{24}+\frac{E_3}{24}+\frac{E_4}{24}} \Bigg(& \lambda_1^4 \left(\frac{E_1}{8}-\frac{E_2}{24}-\frac{E_3}{24}-\frac{E_4}{24}\right) + \lambda_1^3 \left(-\frac{E_1}{3}+\frac{E_2}{6}+\frac{E_3}{6}+\frac{E_4}{6}\right) \\ & + \lambda_1^2 \left(\frac{E_1}{4}-\frac{E_2}{4}-\frac{E_3}{4}-\frac{E_4}{4}\right) + \lambda_1 \left(\frac{E_2}{6}+\frac{E_3}{6}+\frac{E_4}{6}\right) \Bigg),\end{aligned} \tag{4.7}$$

$$\begin{aligned}\xi_2 = & \frac{1}{n_2} \Bigg(\lambda_2^3 \left(-\frac{E_2}{3}+\frac{E_3}{6}+\frac{E_4}{6}\right) \\ & + \lambda_2^2 \left(-\frac{E_1\lambda_1}{2}-\frac{E_2\lambda_1}{2}+\frac{E_2}{2}+\frac{E_3\lambda_1}{2}-\frac{E_3}{2}+\frac{E_4\lambda_1}{2}-\frac{E_4}{2}\right) \\ & + \lambda_2 \left(-E_1\lambda_1^2+E_1\lambda_1+\frac{E_3\lambda_1^2}{2}-E_3\lambda_1+\frac{E_3}{2}+\frac{E_4\lambda_1^2}{2}-E_4\lambda_1+\frac{E_4}{2}\right)\Bigg),\end{aligned} \tag{4.8}$$

$$\xi_3 = \frac{\lambda_3^2\left(\frac{E_3}{2}-\frac{E_4}{2}\right)+\lambda_3\left(E_1\lambda_1+E_2\lambda_2-E_4\lambda_1-E_4\lambda_2+E_4\right)}{\left(\frac{E_3}{2}-\frac{E_4}{2}\right)\left(-\lambda_1-\lambda_2+1\right)^2+\left(-\lambda_1-\lambda_2+1\right)\left(E_1\lambda_1+E_2\lambda_2-E_4\lambda_1-E_4\lambda_2+E_4\right)}. \tag{4.9}$$

Where ξ_1, ξ_2 and ξ_3 are uniform random variables where $\xi_i \in (0, 1)$ and n_2 is the norm for Eq. (4.8):

$$
\begin{aligned}
n_2 = {} & (-\lambda_1+1)^3\left(-\frac{E_2}{3}+\frac{E_3}{6}+\frac{E_4}{6}\right) \\
& + (-\lambda_1+1)^2\left(-\frac{E_1\lambda_1}{2}-\frac{E_2\lambda_1}{2}+\frac{E_2}{2}+\frac{E_3\lambda_1}{2}-\frac{E_3}{2}+\frac{E_4\lambda_1}{2}-\frac{E_4}{2}\right) \\
& + (-\lambda_1+1)\left(-E_1\lambda_1^2+E_1\lambda_1+\frac{E_3\lambda_1^2}{2}-E_3\lambda_1+\frac{E_3}{2}+\frac{E_4\lambda_1^2}{2}-E_4\lambda_1+\frac{E_4}{2}\right).
\end{aligned}
\tag{4.10}
$$

λ_4 is determined by λ_1, λ_2, and λ_3:

$$\lambda_4 = 1 - \lambda_1 - \lambda_2 - \lambda_3 \tag{4.11}$$

These equations only require three random numbers to sample from a linear equation on a tetrahedron but Eq. (4.7) requires inverting a fourth order polynomial and Eq. (4.8) requires inverting a third order polynomial. These equations were tested and the mean sampled position converges to the rejection sampling average as the number of samples is increased. In the tested implementation, the rejection sampling is about ten times faster than the conditional sampling method. These results are not definitive: there may be a way to optimize the root finding method to select the correct root without an expensive root solve.

5. IMPLICIT CAPTURE WITH NON-CONSTANT OPACITY AND BASIS FUNCTIONS

In the Monte Carlo method, a particle interacts with the background medium. These interactions can be handled as discrete events or as continuous interactions. Modeling the interaction as continuous is a type of absorption weighting, which is a variance reduction technique [35]. Particle absorption is the most straightforward event to treat continuously, this is called *implicit capture*. In the Implicit Monte Carlo method, the end of time step event can also be treated as a continuous interaction [54]. For radiative transfer, the opacity is usually treated as being constant within a mesh cell. In that case, implicit capture is simply the attenuation of the particle's energy-weight over its path:

$$w(s) = w_0 e^{-\sigma_a s}. \tag{5.1}$$

Where w is the energy-weight of the particle, w_0 is the initial particle weight, σ_a is the absorption opacity and s is the path length. A loss of energy-weight in the particle corresponds to an increase in energy in the material. The absorbed energy in the material is thus:

$$w_{abs}(s) = w_0 \left(1 - e^{-\sigma_a s}\right). \tag{5.2}$$

In the finite element IMC method, the absorption is projected onto a trial space. If that space is linear, absorption is now weighted with a linear basis function. In the material energy balance, the absorption term becomes:

$$E_{absorbed} = \int\limits_{\mathbf{x}} \int\limits_{0}^{\infty} \int\limits_{4\pi} N^v(x)\sigma I(x,\Omega,\nu)\ d\Omega\ d\nu\ dx \tag{5.3}$$

Where N^v is a linear basis function and $\mathbf{x}$ is the spatial domain. The finite element method also changes the implicit capture method—instead of a depositing energy in to a single mesh cell as a particle streams, the absorbed energy is weighted by the basis functions in the cell. In analog tracking, the absorbed energy can simply be multiplied by the value of the shape functions at that point. With implicit capture, the shape function is multiplied by the differential absorption and then integrated over the path-length. The derivative of Eq. (5.2) with respect to the current path length, s, is:

$$\frac{dw_{abs}}{ds} = w_0\sigma_a e^{-\sigma_a s}. \tag{5.4}$$

Eq. (5.4) is then multiplied by a linear basis function and integrated over the total path length, d. This yields an expression for the energy absorption weighted by the linear basis function:

$$w_0 \int_0^d N^v(s)\sigma_a e^{-\sigma_a s} ds = w_0 m \left(\frac{1}{\sigma_a} - e^{-\sigma_a d}\left(\frac{1}{\sigma_a} + d\right)\right) + b w_0 \left(1 - e^{-\sigma_a d}\right). \tag{5.5}$$

Where m and b are the slope and intercept of the basis function of node v along the path length. The total amount of energy absorbed into the spatial cell is the same for both Eq. (5.5) and Eq. (5.2). Because the total energy absorbed in the material is the same, the particle's energy can be updated simply with Eq. (5.1).

5.1 Linear Opacity

In the DFEM IMC method and in discrete ordinates methods for thermal radiative transfer the material temperature is treated as being piecewise linear. This implies that the opacity is not constant within a mesh cell and should be projected onto the same space as the temperature. If a linear opacity is used in the IMC method, the differential absorption becomes:

$$\frac{dw_{abs}}{ds} = w_0 \frac{d\tau(s)}{ds} e^{-\tau} = w_0 \sigma_a(s) e^{-\tau}. \tag{5.6}$$

Where τ, is the total optical depth for the linear opacity case:

$$\tau = \int_0^s \sigma_a(s')ds' = \int_0^s m_\sigma s' + \sigma_0 ds' \tag{5.7}$$

Where σ_0 is the opacity at $s = 0$ and m_σ is the slope of the opacity along the particle path. When combined with the linear basis function, the differential absorbed energy for a node, v is:

$$\frac{dw_{v,abs}}{ds} = w_0 N^v(s) \sigma_a(s) e^{-\tau}. \tag{5.8}$$

Equation (5.8) is not analytically integrable and must be evaluated numerically. Several techniques for evaluating this integral are presented. Gauss-Legendre and Gauss-Labotto quadrature are examined. Unfortunately, quadrature rules for integrals over the range of $(0, \infty)$, such as Gauss-Laguerre, are not useful for this integral because the linear basis functions are zero outside the mesh cell and the linear opacity functions can become negative outside the mesh cell. In addition to quadrature rules, the integral can be evaluated by breaking up the path length and moving the particle as if it there were some number of constant opacity sub-steps. This is done

case	σ_0	m_σ	b	m	τ
1	1.0	10.0	1.0	0.0	1.0
2	1.0	10.0	1.0	-1.0	1.0
3	100.0	1000.0	1.0	-1.0	1.0
4	100.0	1000.0	0.0	1.0	10.0
5	100.0	1000.0	0.0	1.0	10.0
6	100.0	1000.0	1.0	-1.0	10.0
7	100.0	1000.0	0.0	1.0	20.0
8	100.0	1000.0	1.0	-1.0	20.0

Table 5.1: Parameters of a the absorption function to be integrated numerically

by assuming the opacity and shape function are constant within a sub-step and evaluating both properties at the mid-point of the sub-step. We call this integration method *discrete tracking*. Equation (5.5) can be used to treat the shape function as linear while assuming the opacity is constant over a sub-step, we call this method *shape integral tracking*.

5.2 Results

The quadrature rules and particle tracking methods are compared for several cases shown in Table (5.1). The cases represent circumstances that would likely be encountered during IMC simulations—thick and thin opacities that vary rapidly within a cell and the basis function varying drastically along a path length. The first case tests the ability of the integration method without basis functions. The second case tests relatively thin opacity. The subsequent cases test thick opacity cases, where the particle weight changes rapidly over the path length. Each method is compared to the result from a midpoint-rule numerical integration with 10^6 points. The results are shown in Figs. (5.1) to (5.8).

The particle tracking methods are exact for case 1, when the basis function is held constant. The Gauss-Legendre quadrature is effective at integrating cases 1-4

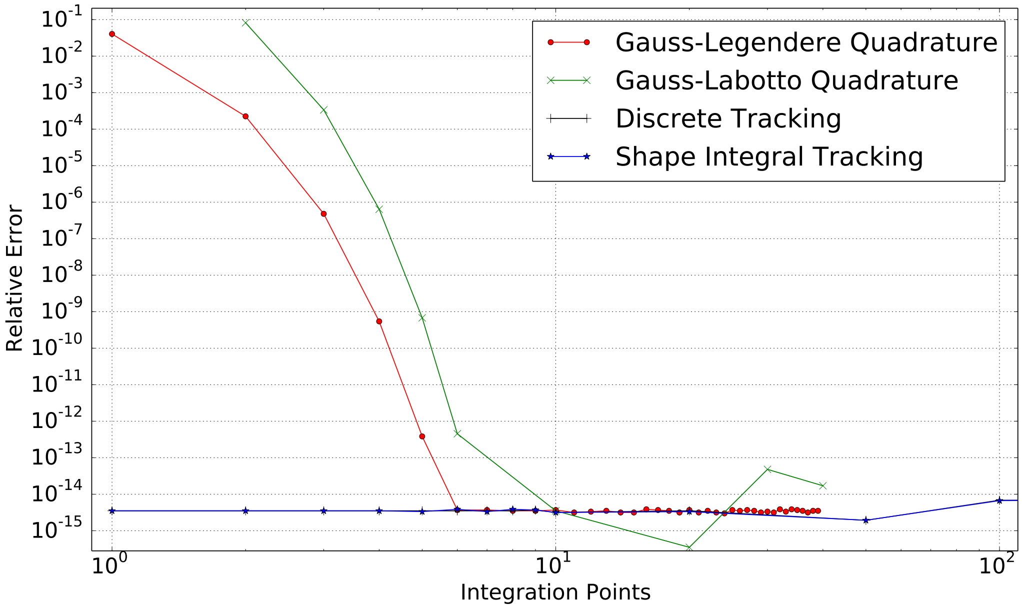


Figure 5.1: Absorption integral for case 1 with various numerical integration methods

where the optical depth, τ, is one. Case 7 proves to be the most difficult—in this case the particle travels a large number of mean free paths in the material and the linear basis function starts at 0.0 and increases to about 0.12. Using four integration points with the Gauss-Legendre method yields the lowest error compared to the other methods for all but case 8. In the majority of cases, using four points produces less than 0.1% relative error in the solution. Thus Gauss-Legendre with four integration points was used to compute the implicit capture integral in the DFEM IMC method. For the test problems studied in this work, a particle traveling more than ten mean free paths is unlikely because of effective scattering and the Gauss-Legendre method will be sufficient.

Evaluating the implicit capture term with a quadrature rule is a potential target for vectorization. It is difficult to find vectorizable operations in the IMC particle loop because adjacent particle histories quickly diverge and require different data or

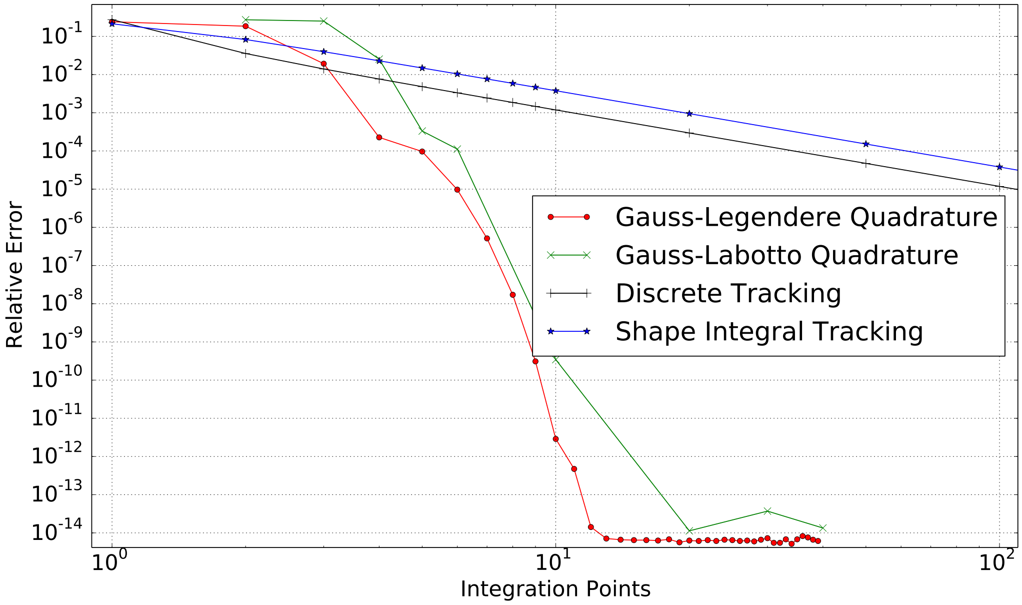


Figure 5.2: Absorption integral for case 2 with various numerical integration methods

have different interactions. Evaluating the implicit capture term in a DFEM IMC method with Gaussian quadrature is two nested loops: the outer loop is over test functions in the cell and the inner loop is over the integration points. If these loops could be fused or if more integration points were used, it may be large enough to benefit from vectorization.

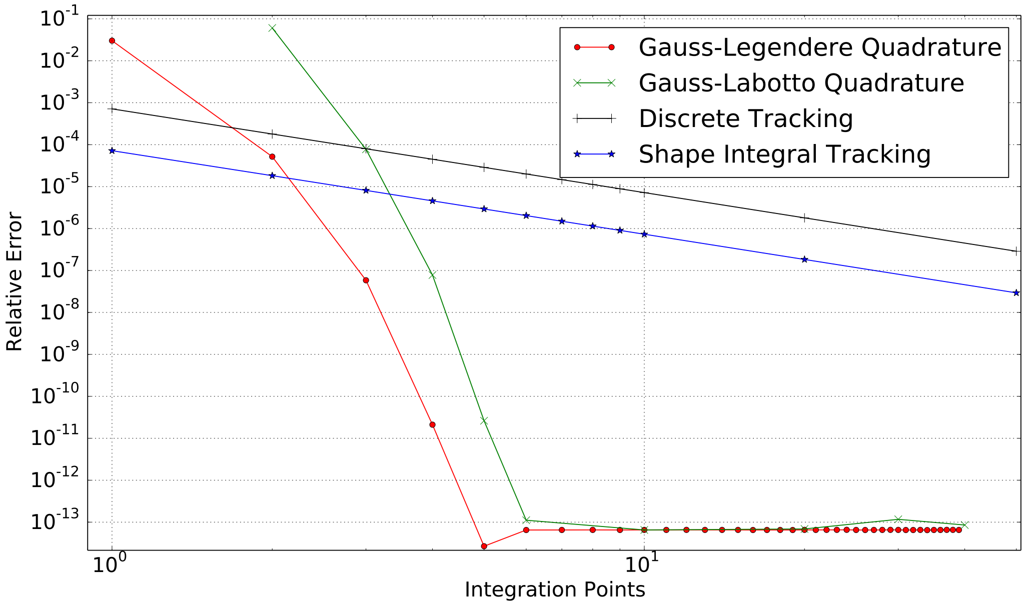


Figure 5.3: Absorption integral for case 3 with various numerical integration methods

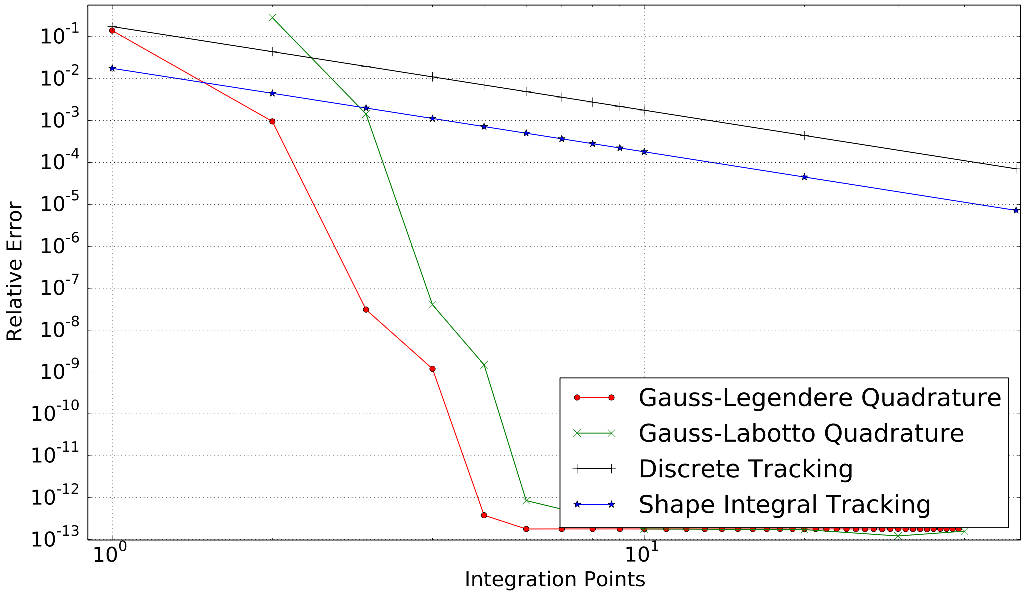


Figure 5.4: Absorption integral for case 4 with various numerical integration methods

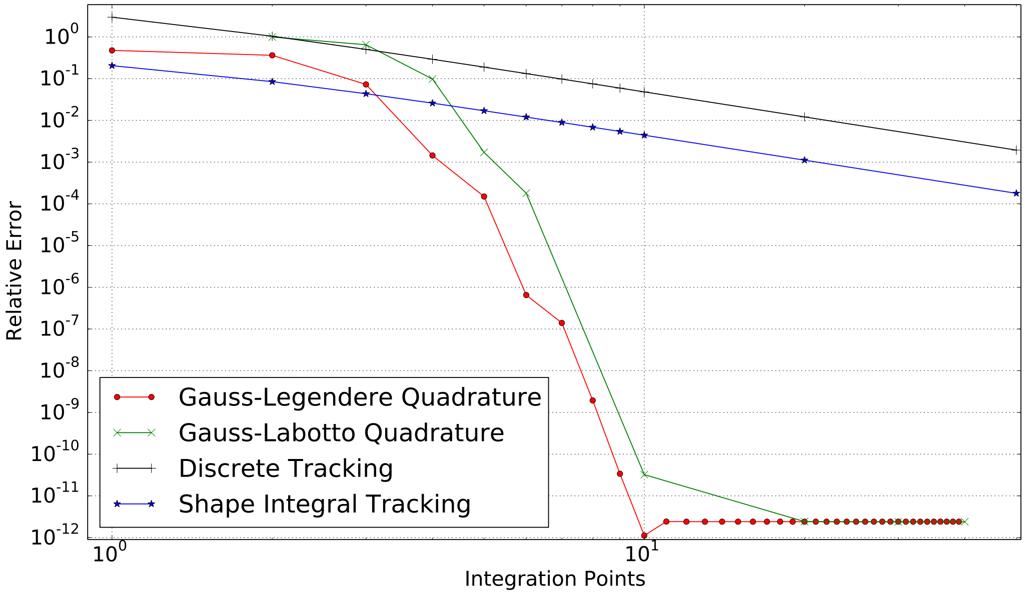


Figure 5.5: Absorption integral for case 5 with various numerical integration methods

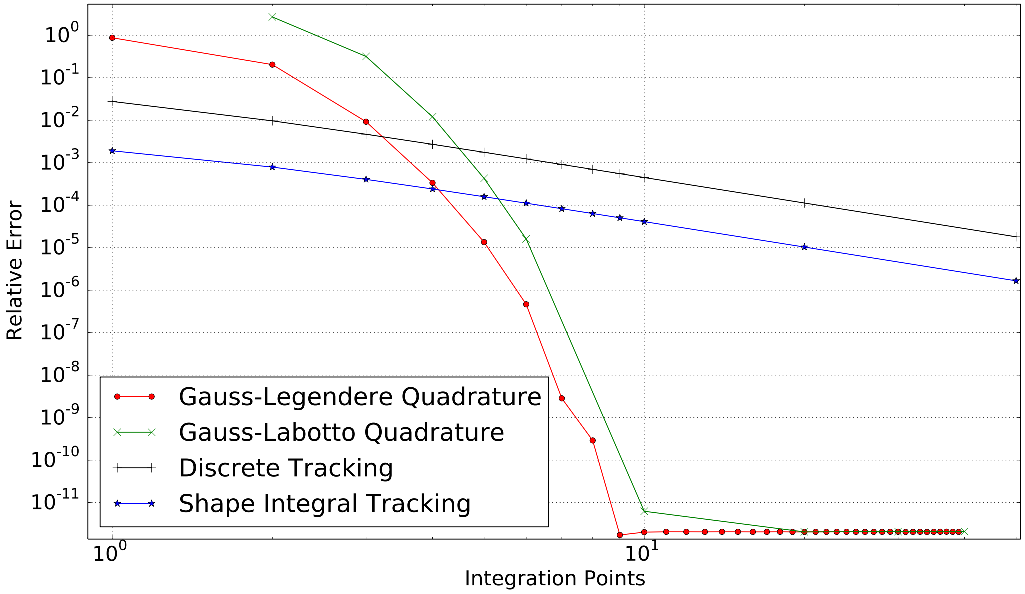


Figure 5.6: Absorption integral for case 6 with various numerical integration methods

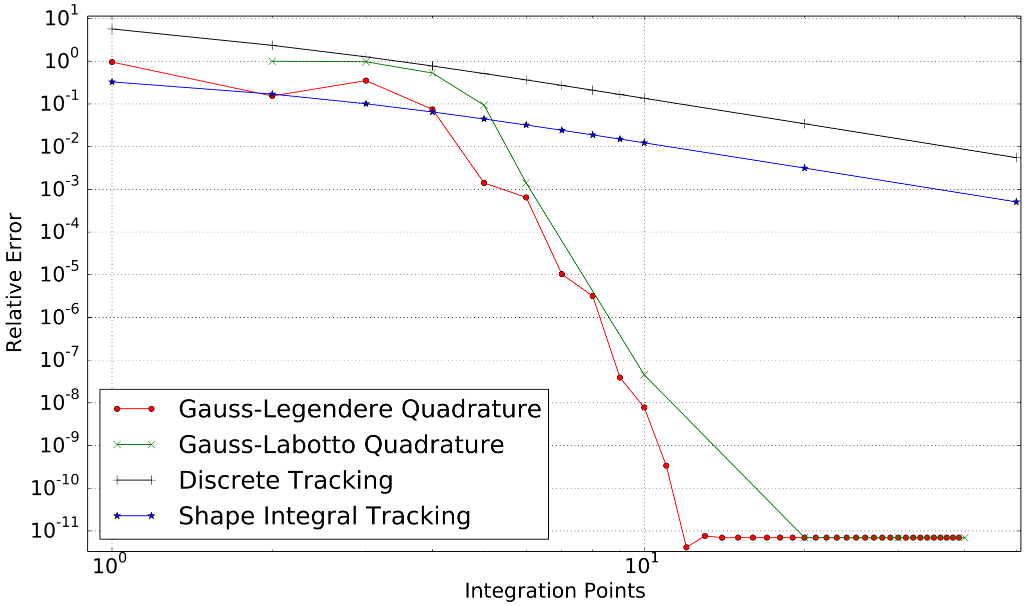


Figure 5.7: Absorption integral for case 7 with various numerical integration methods

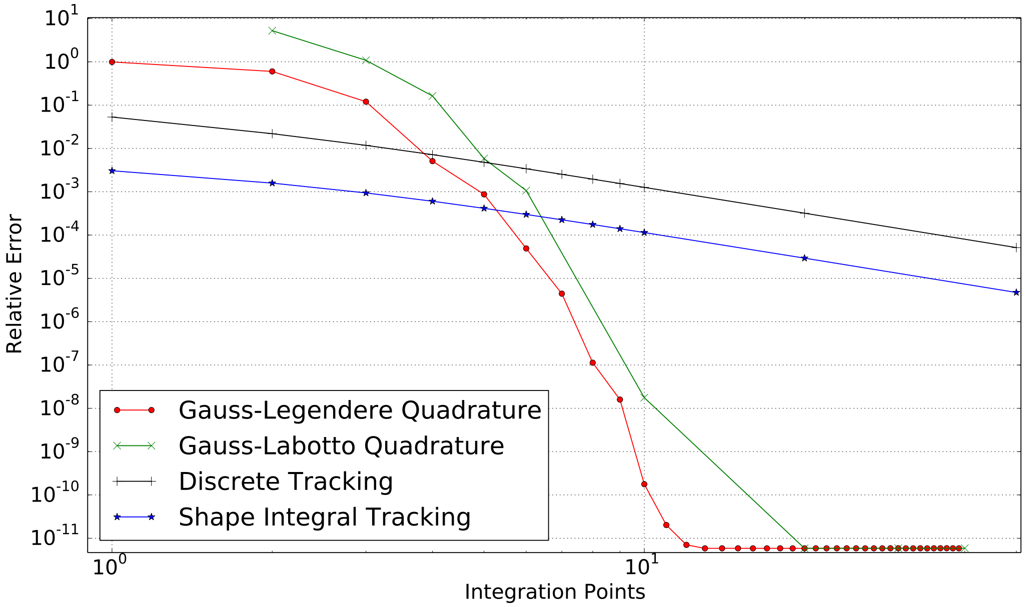


Figure 5.8: Absorption integral for case 8 with various numerical integration methods

6. SERIAL AND PARALLEL PERFORMANCE

6.1 The High Performance Computing Landscape

The technology of supercomputers has changed drastically over the past 50 years and continues to change [27]. Previously, gains in performance were achieved by placing more transistors on a processor. Because the power dissipation of a CPU is proportional to the cube of the clock speed, the trend of simply placing more transistors on a chip became impractical. To obtain gains in performance without significantly increasing power consumption, multiple linked processors with lower clock speeds are placed together on a processor. This approach is known as *multicore.* Multicore machines still form most of the top supercomputers in the world. The newest supercomputers, including the Tianhe supercomputer in China [28] and the impending Trinity supercomputer at Los Alamos National Laboratory [29] use what is known as *manycore* architectures. In manycore systems, the multicore approach is extended further to tens to hundreds of low-clock-speed cores. This increases the power efficiency of the system, but more cores means that applications must be able to parallelize computational work effectively. In the multicore era, the spatial domain was usually decomposed so that each individual core was given a spatial subdomain to do work on. When the solution at the boundary [1] of the subdomain is required, these processors would communicate with adjacent processors. In manycore systems this strategy for parallelism is inefficient—the surface to volume ratio of each subdomain becomes large as the number of cores is increased and the amount of memory required to store boundary cells becomes significant. One solution to this approach is to use multiple parallel processes within a subdomain. This can be accomplished

[1] The boundary of a subdomain is often called a *halo*, because it looks like a ring in 2D

with a variety of different software tools used together or individually. The newer manycore architectures also have multiple instruction pipelines on a core, which allows for *threading*. Threading can potentially allow the arithmetic processing unit on a CPU to be used more effectively. Manycore machines also have wider vector registers, which allow a single instruction to performed on multiple data simultaneously. Making use of these wider vector registers is called *vectorization*. To use the full resources of a manycore system, an application must be able to effectively parallelize computational work, make use of threads and vectorization.

6.2 Requriements for High Energy Density Physics Simulations

In High Energy Density Physics problems, there are certain phenomena that require 3D modeling—this is especially true in ICF problems, where mixing, instabilities and turbulence are represented poorly by 1D and 2D models [19]. In 3D, the memory required to store a discrete mesh with multiple unknowns and physical data per cell grows quickly. For example, if each dimension has 10000 discrete points, then the amount of memory required to store pressure, density, temperature, energy and 30 opacity values in each cell is:

$$10000^3 \text{ cells} \times 34 \, \frac{\text{double values}}{\text{cell}} \times 8 \, \frac{\text{bytes}}{\text{double value}} = 2.72 \times 10^{14} \text{ bytes.}$$

This is about 20% of the total 1,572,864 GB of memory available on the Sequoia supercomputer at Lawrence Livermore National Laboratory, this corresponds to about 270,000 cores. In addition to the memory requirements, the amount of computational work is also dependent on the number of mesh cells—physics algorithms solve a discrete system of equations to determine the values of the unknowns for each mesh cell.

Many physics equations involve spatial gradients of the solution. When these equations are discretized these spatial gradients introduce a dependence of the unknown values in one cell on the unknowns in adjacent cells. This means that when the mesh is decomposed on to subdomains, the solution on one sub-domain is dependent on the data in adjacent subdomains. In IMC this coupling between sub-domains comes from particles that are emitted in one sub-domain that deposit energy in other sub-domains. This dependence on the data in adjacent sub-domains means that subdomains must communicate with each other in order to solve discrete physics equations. If the cost of this communication were negligible, the time to run a simulation with 1000 mesh cells on one core would be the same as the time to run 1.0×10^9 mesh cells on one million cores. In reality, the communication cost is non-negligible. *Scaling* describes an algorithm's ability to run in parallel effectively. *Weak scaling* describes how well an algorithm runs on larger problems and *strong scaling* describes how well an algorithm runs the same amount of computational work on more parallel processes.

6.3 Monte Carlo, IMC and Scaling

A critical assumption of neutral particle transport in the Boltzmann equation is that particles do not interact with each other. This means that each Monte Carlo particle history can be carried out independently, which makes a parallel implementation of the Monte Carlo method relatively simple. Because particle histories are independent, Monte Carlo is often referred to as being "embarrassingly parallel." In reality, Monte Carlo is only embarrassingly parallel when the mesh information and physical data are small enough to be stored locally on each parallel process. That model, called *domain replication* is simple to implement: the only parallel communication is a reduction operation at the end of a simulation step, as illustrated in

Fig. (6.1). If the mesh data and physical data are larger than the memory of a single parallel process (which is common for HEDP problems, as described above), the mesh and physical data must be decomposed and assigned to parallel processes. This case is called *domain decomposed*. As HPC architectures trend toward more cores and threads, the available memory per core and thread decreases and the use of domain replication becomes less feasible for most problems of interest. Therefore, domain decomposition is the standard procedure for multiphysics problems and Monte Carlo transport.

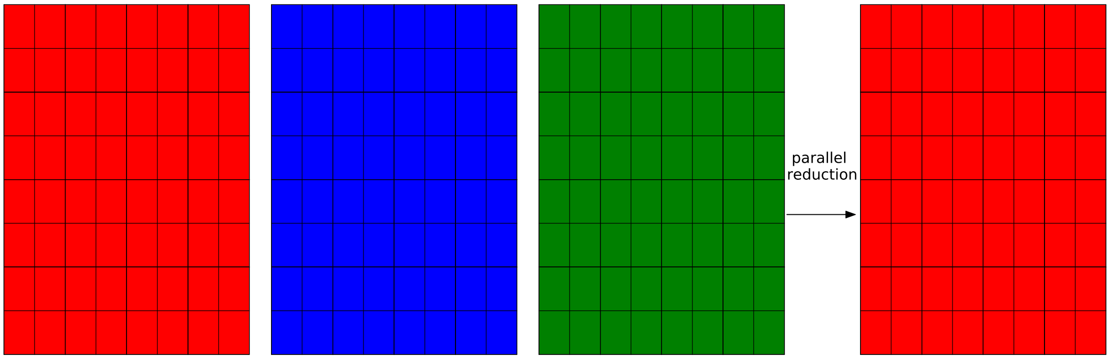


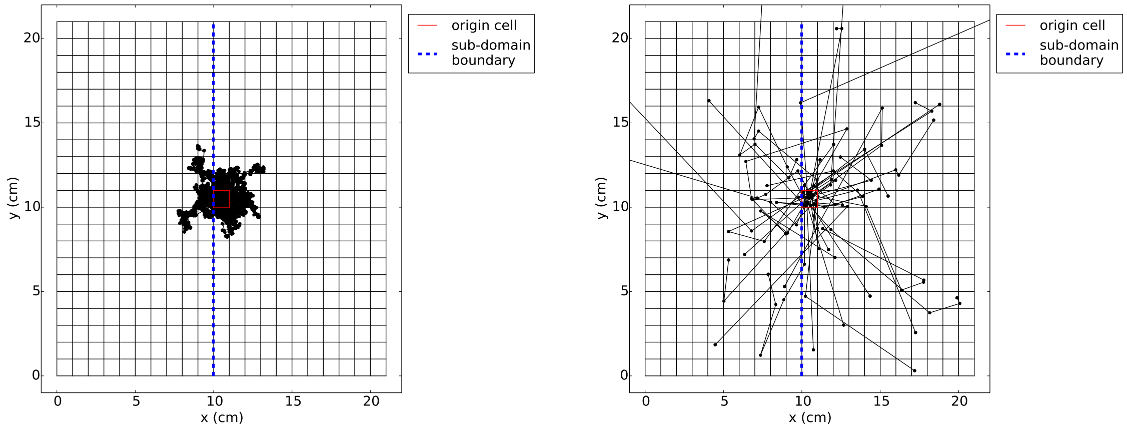
Figure 6.1: A mesh replicated on three parallel processes—simulation data from each rank is reduced to one rank

To evaluate the performance of a domain decomposed IMC algorithm we need to understand the computational requirements of the IMC method. The IMC method must perform well in the optically thick and optically thin regimes. Fig. (6.2) shows particle histories originating from a single cell on a 2D mesh in both the optically thick and thin regimes respectively. In the optically thick case, shown in Fig. (6.2a), each particle has many interactions but stays very close to its birth place. Many interactions are expensive—at each point a distance to boundary, scatter and end of timestep must be calculated. The DDMC method of Cleveland, Densmore and Gentile aids in reducing the computational work in processing these particles. Although the DDMC method is not considered in this work, it could improve load balancing because it means that particles in optically thick regions will not take significantly longer than thin particles. In the thin limit, shown in Fig. (6.2b), particles generally move far from their origin cell but have relatively few scattering interactions. The geometric data, opacity data and energy and momentum tallies are required at each cell and are not reused once the particle leaves the cell. This means that particles in optically thin regions will load much more data from memory than thick particles. Because particle histories are determined with random numbers, subsequent particles are not likely to reuse data loaded from memory by the previous particle, which could lead to poor cache usage. A parallel algorithm for IMC also must work well in the thick and thin limit. In the thick case that means that an algorithm must be able to handle many particle crossings between sub-domains. In the thin limit, an algorithm must be able to account for particles penetrating far into other sub-domains.

6.4 On-Node Algorithms

The simplest method of domain decomposition is to implement an algorithm where cores or threads share access to local memory. In this work, we examine shared memory systems known as a *compute nodes*.

A compute node is a grouping of cores and threads that all share access to some amount of memory. In HPC machines, multiple compute nodes are networked together. The time for a core or node to access memory on its local compute node is much faster than accessing memory on another compute node in the network, which generally requires additional communication protocols. Algorithms can take



(a) Particle paths from 40 particles starting in the center mesh cell with a mean free path of 0.1 cm

(b) Particle paths from 40 particles starting in the center mesh cell with a mean free path of 10.0 cm

Figure 6.2: Comparison of particle histories in thin and thick regions

advantage of the close proximity to memory on a compute node by reworking parallel algorithms and assuming that on-node access is very inexpensive. There is a often a memory hierarchy on compute nodes: within a compute node there are non-uniform memory access (NUMA) nodes. On a compute nodes, some subset of the cores will be closer (and thus have faster access) to a subset of the total system memory. Memory access between NUMA nodes is relatively slower than access to memory within a NUMA node but much faster than memory access off of a compute node.

To optimize parallel performance on a compute node, an algorithm must take into account NUMA. This can be done by ensuring that cores or threads on a NUMA node do work on memory that is on their NUMA node whenever possible. Memory that is allocated by a thread or core is close to the NUMA node of that thread or core. This is the principle of *first touch*.

To have the advantages of first touch in an IMC algorithm, each thread or core needs to allocate the data for particles that it will transport. This means that the particle data is as close as possible to the thread that will use it. An IMC algorithm using the OpenMP threading library was implemented with a simple Cartesian mesh. In this implementation, threads make and transport the particles for a group of cells. The purpose of giving cells to threads instead of a group of particles is because the thread working on a group of cells will allocate the memory for those particles. If instead the thread was given a group of particles from an already allocated array it would be difficult to ensure that those particle were close to the thread in the memory hierarchy.

Two additional challenges in implementing a shared memory parallel algorithm arise because memory is shared between threads on a compute node. The first challenge is *race conditions*: the programmer must ensure that a memory location cannot be accessed while a thread is writing to that location. For example, if a thread tries to increment an integer at a memory location, it first reads the integer value from memory, adds one to it, then writes it back. If a second thread reads the memory location before the first thread is finished writing, the second thread will have read the incorrect value of the memory location. If the second thread increments the value and writes it back to memory the first threads actions will have been nullified. Race conditions can be avoided by locking a memory location when a thread wants to write it—this forces other treads attempts to access the memory location to wait until the memory location is unlocked. Another solution is to have each thread make a copy of the memory location and perform a reduction after the parallel loop is finished. This adds memory overhead but avoids the waiting involved in locking a memory location. In the IMC on-node memory algorithm, the absorbed energy array is where race conditions can occur—because particles owned by threads can travel to any cell on a mesh, multiple threads may try to update the amount of absorbed energy in the same cell. The amount of memory needed to replicate the absorbed energy array for each thread (an array of doubles of size N, where N is the number of mesh cells) is generally much smaller than the amount of memory required by the IMC particles, so we use the replicated array strategy to avoid race conditions.

The on-node parallel IMC algorithm can be described in five steps:

1. A thread is assigned a group of cells by using OpenMP loop over cells on the mesh.

2. This thread creates and transports particles representing the energy in the cells that it was assigned. Each thread keeps a copy of the absorbed energy array to avoid race conditions when writing to the absorbed energy array.

3. After the loop over cells has been completed, a reduction is performed on the absorbed energy array and the temperature is updated

4. The list of census particles is sorted in parallel by cell index. This is done so that a thread assigned a cell can easily simulate the census particles belonging to that cell on the next timestep.

5. The timestep is advanced and step 1 begins again.

This algorithm was tested in weak and strong scaling on the Intel Xeon Phi and Intel Knights Corner (KNC-MIC). The results for an infinite medium problem are shown in Figs. (6.3) to (6.4). The Intel Knight's Corner has 60 cores with 4 hardware threads each for a total of 240 parallel processes. The Xeon Phi has 32 cores and 2 software threads per core, for a total of 64 parallel processes. The algorithm scales well on both architectures when the number of threads is less than or equal to the number of cores. The possible effect of NUMA nodes can be seen with the decrease in weak scaling efficiency on the Xeon-Phi, where eight cores comprise a NUMA node. The efficiency drops from 97.5% on 8 cores to 93% on 16 cores. The efficiency continues to drop to 83% on 32 cores, when all NUMA nodes are being used. The algorithm scales well on the MIC, but the absolute runtimes are longer for the same number of cores on the Xeon Phi. The cores are less powerful on the KNC-MIC, but the algorithm could be faster if it attempted to use the longer vector registers on the KNC-MIC. Currently, there is no work on modifying IMC for vectorization but it could follow the event based Monte Carlo approach of Brown and Martin [4].

A brief discussion on the particulars of threading helps explain what appears to be very poor scaling on the KNC-MIC above 60 cores. Modern processors use *simultaneous multithreading*, which allows the instruction pipeline to be more fully utilized. This is accomplished by having multiple hardware threads per processor core. These hardware threads can independently issue instructions to be executed by shared CPU resources. In the results shown in Fig. (6.3), the scaling results above 60 threads are using multiple threads per core. Although the scaling drastically decreases as multiple threads per core are used, the amount of power required to run in this mode does not significantly increase. To run with 60 cores and four hardware threads per core uses about the same amount of power as 60 cores with 1 hardware thread per core. We assume that the power requirement with all cores active is twice the thermal designed power (TDP). If a smaller number of cores are used, the power use is assumed proportional to the number of active cores relative to the total TDP. The TDP for the Intel Knight's Corner used in this study is 225 W. Figure (6.5), a second weak scaling plot, shows the weak scaling when each core is run with 4 hardware threads. In this case the scaling at 60 cores (for a total of OpenMP 240 threads) is about 0.77. Comparing runtimes where 4 cores are used with one hardware thread to the runtime of 1 core with 4 hardware threads shows the benefit of hardware threads: 4 cores with one hardware thread each takes 66 seconds and uses about 1950 joules. The same problem run with 1 core and four hardware threads takes 121 seconds and 907 joules. If two cores with four hardware threads are used for the same amount of work as the previous case, the runtime is 60.8 seconds and 912 joules of energy are used. Thus, for less than half the energy requirement the same amount of computational work can be performed when threading is used.

An efficient on-node algorithm could be beneficial in the context of a multi-node domain decomposed IMC algorithm. Currently, domain decomposed algorithms do not utilize the shared memory within a compute node. Treating one compute node as a spatial sub-domain would reduce the surface to volume ratio, which would mean less particles moving off of the sub-domain and incurring parallel communication.

6.5 Multiple Node Domain Decomposed Algorithms

If the domain is too large to fit on one compute node, the mesh must be decomposed across compute nodes. The current IMC algorithm for a decomposed mesh is to pass particle between sub-domains during a timestep. A sub-domain must also receive particles from other sub-domains. An illustration of this method is shown in Fig. (6.6), where the arrows represent particles moving between sub-domains and thus incurring parallel communication. A method for minimizing the time spent in communication in the particle passing method is examined by Brunner et al [7]. This method finds an optimal frequency to check for incoming particles and an optimal buffer size for passing particles to other sub-domains.

A drawback of this approach is that it must be done very carefully to avoid load imbalance—even if the number of particles is the same on each processor, a domain boundary in a diffusive region will result in poor scaling as scattering particles are passed back and forth between processors. The issue of load imbalance becomes

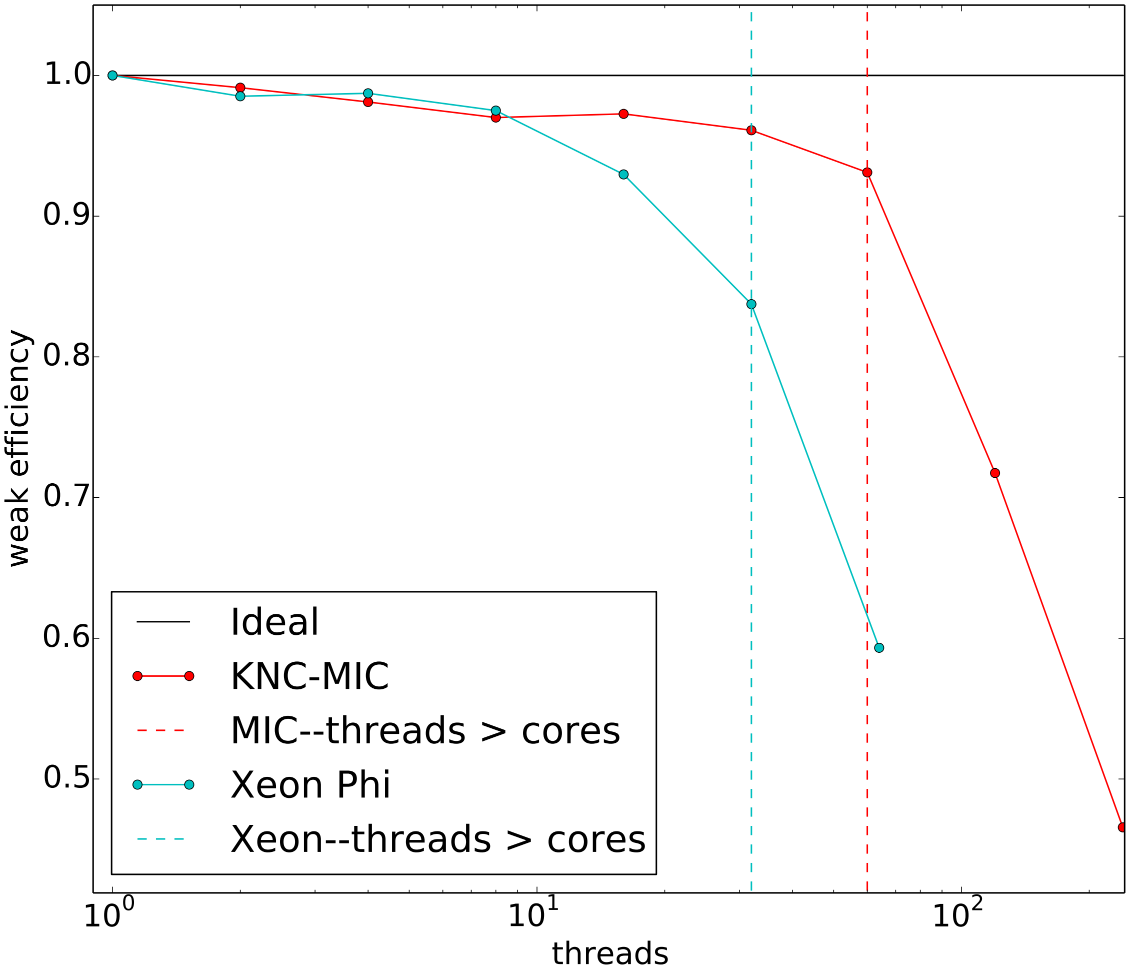


Figure 6.3: Weak scaling in the infinite medium problem with 10 timesteps

more difficult as the number of subdomains increases.

In this work a method is studied where the mesh is decomposed but the particles are not necessarily decomposed in the same way as the mesh. The crux of this method is that particles are not passed between parallel processes. Each processor is given some amount of particles and then requests spatial information as it is needed by the particles. This spatial information may or may not belong to the processor that is transporting the particles. If it does not belong to the mesh, a parallel com-

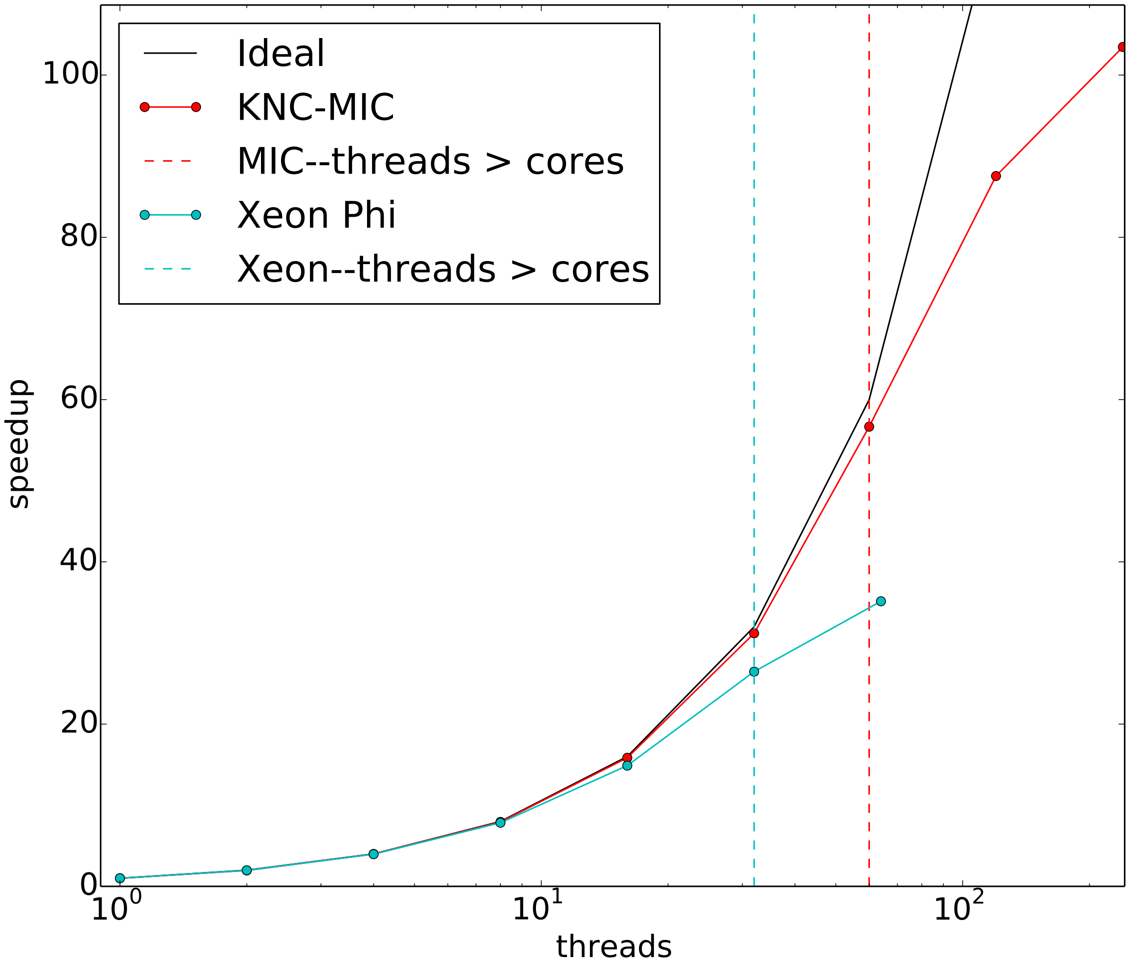


Figure 6.4: Strong scaling in the infinite medium problem with 10 timesteps

munication is incurred. We called this approach *particle-based domain decomposition* as opposed to the current approach which we call *particle-passing domain decomposition*. This method is illustrated in Fig. (6.7)—particles owned by one processor are using mesh owned by another processor. The particle-based method removes the pitfall present in the particle-passing method where particles in diffusive regions can simply scatter between spatial subdomains. In the particle-based method, the particle can scatter back and forth between local and non-local data and if that non-local data is stored locally after it is requested, scattering no longer incurs parallel

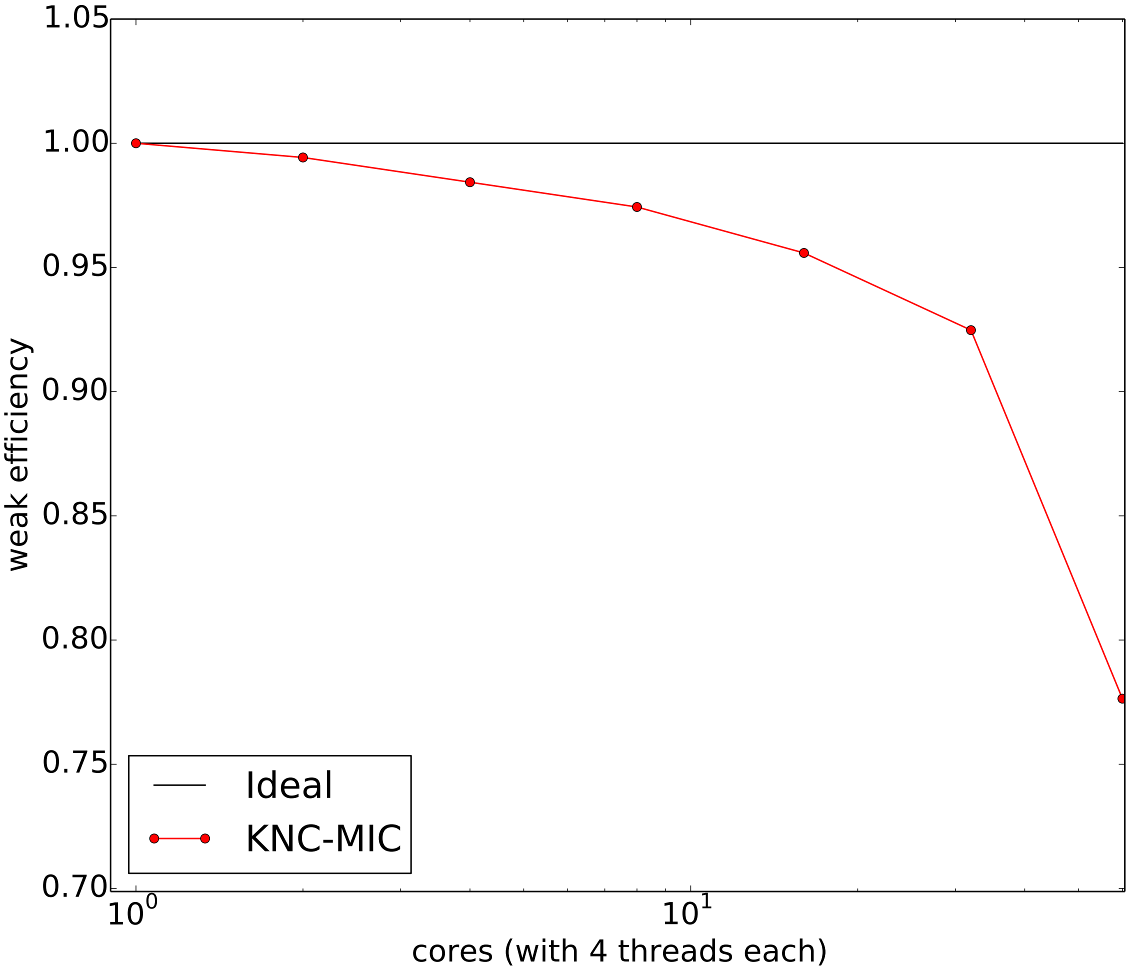
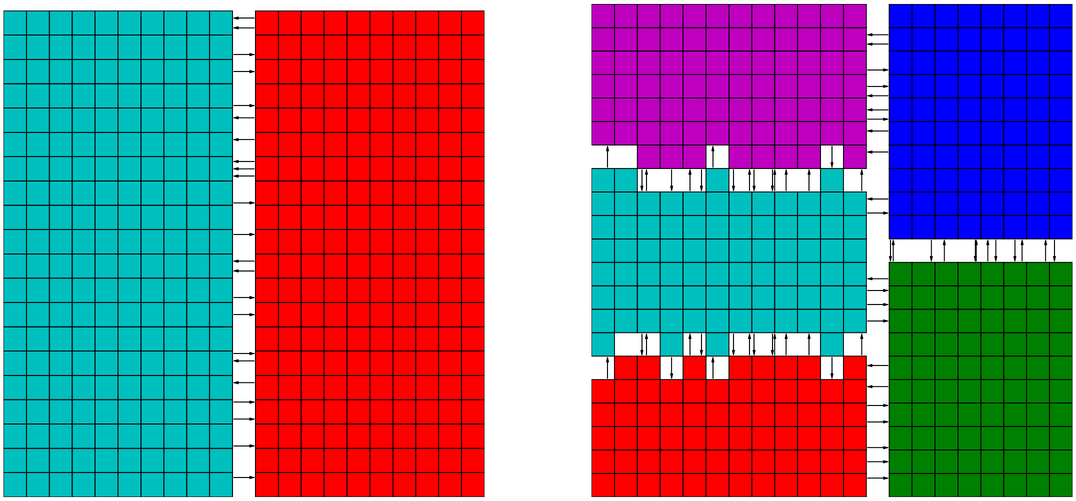


Figure 6.5: Weak scaling in the infinite medium problem with 10 timesteps using four hardware threads for each core

communication. To achieve the same goal of reducing communication due to scattering particles, the Monte Carlo reactor physics code SHIFT at Oakridge National Laboratory defines some number of mesh cells from the sub-domain boundary that a parallel process also stores in local memory, but does not own [55][48]. This is equivalent to having a larger layer of ghost cells. This method is not pursued here as a way to reduce the communication cost of scattering particles because the end goal of the particle-based method is to reduce load imbalance by having the abil-



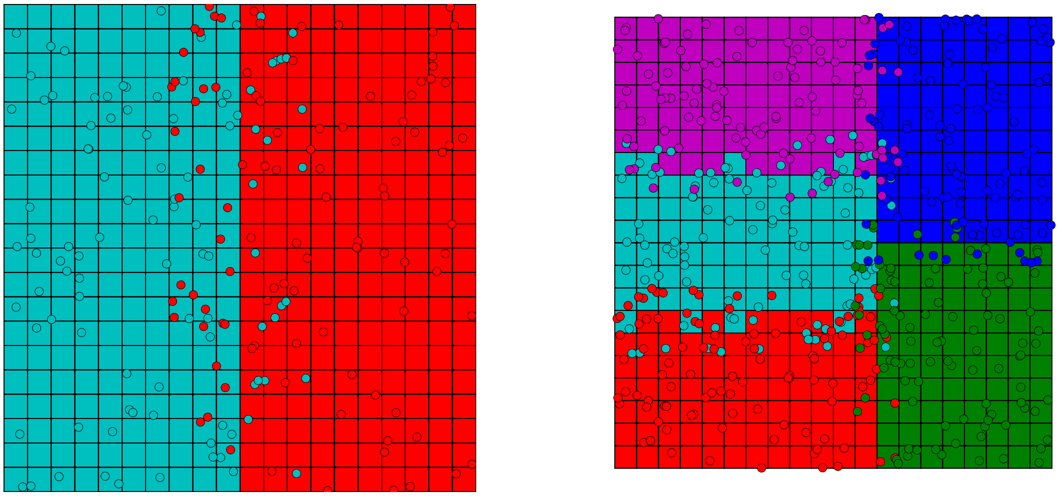
(a) Domain decomposition and particle-passing with two sub-domains

(b) Domain decomposition and particle-passing with five sub-domains

Figure 6.6: Examples of a decomposed domain and particles moving between sub-domains

ity to partition particles equally between parallel processes. If this could be done effectively, hot regions in a problem could easily be load balanced. In the current domain-decomposed method, hot regions of the problem have a large share of the work and put a communication strain on the system. The solution to this problem in the domain-decomposed method is to replicate domains that have a large amount of work. This, however, does not solve the problem of particles scattering back and forth between domain boundaries. This is made worse by the fact that effective scattering is higher in hot regions of the problem.

An additional detail of parallel algorithms for IMC is how to handle the census particles. One downside of the particle-passing method is that the census particles usually need to be rebalanced after a timestep. For the infinite medium cases studied here, census particles that are not on the local mesh are sent to the parallel process



(a) Example of particle-based method with two sub-domains

(b) Example of particle-based method with five sub-domains

Figure 6.7: Examples of a particle-based domain decomposed method with two sub-domains and five sub-domains

that does own the mesh. In the simple particle-passing method all particles are always on the processor that has their mesh data locally.

Two implementations of the particle-based decomposition are explored. In the first implementation, which we call the *window* method, a parallel process sets up its mesh data so that other processors can read that data during a timestep. This memory access is read-only but the speed of this operation is dependent on the machine architecture and the parallel library implementation. In the second implementation, non-blocking messages are used to interleave parallel communication and work, we call this method the *non-blocking* method.

6.5.1 The Window Method Implementation

In the window method, we use MPI windows to provide global access to the mesh data. After the timestep, a reduction is performed where a parallel process sends absorbed energy in cells that it does not own back to the processes that do own that mesh.

The particle-based decomposition relies on Remote Memory Access (RMA) operations to load mesh data that is needed by particles. RMA operations are the main parallel communication in the particle-based decomposition.

Avoiding unnecessary RMA operations provides the same benefit as queuing particle passing events and particle receiving events in the particle-passing method—both serve to reduce parallel communication. In the simplest case, each time a particle moves into an element that is not on the parallel process it requests the mesh data with an RMA operation.

The first improvement is to use a "working mesh" that stores mesh data that has recently been loaded with an RMA operation. Now when a particle needs mesh data that is not local, it first checks the working mesh to see if this data has already been loaded. The working mesh cannot be arbitrarily large as that would limit the size of the mesh on the parallel process and thus defeat the goal of domain decomposition. Instead, the working mesh must be a limited size. Ideally, the working mesh should keep data that is accessed often and when the working mesh is full and evict data that is used rarely. This behavior is essentially like the desired behavior of cache memory used in modern computers.

6.5.2 The Non-Blocking Method Implementation

The non-blocking method is implemented by adding an additional step in the particle transport loop where a parallel process requests data that it needs and checks to see if requested data has been received. This processing of parallel communication can be set with some user defined frequency—if the problem is dominated by optically thin behavior, particles will make many requests for data that is not owned by their parallel process and the frequency to process parallel communication should be higher. In the optically thick case, particles are less likely to stream far from the local memory and the parallel communication can be less frequent. Particles that are waiting on data are placed on a waiting list, which is implemented as a first in, first out queue. When new data is received from a parallel communication the particles in the waiting list are simulated—if they complete their history they are removed, otherwise they are added to the back of the waiting list. The non-blocking method also uses a working mesh to store mesh data that has been requested by other processors.

The non-blocking method is potentially much better than the window method because it can effectively interleave parallel communication and work—while some particles are waiting for mesh data to be received, particles that only need local data are being simulated. In the window method, the whole simulation waits for the RMA access to complete before continuing. The main impetus for trying the window method is because there are network architectures that support RMA operations which require very low overhead from the operating system. These high efficiency operations are called *direct remote memory access* (DRMA). Modern interconnects such as InfiniBand support these DRMA operations[43] and DRMA operations have been studied as a means of doing collective parallel operations [26].

6.5.3 Domain Decomposed Algorithms Results

The particle-based method was implemented using a simple Cartesian mesh. MPI was used for message passing and creating the shared memory windows. The METIS library [32] from The University of Minnesota was used to decompose the mesh with the Recursive Coordinate Bisection method.

The particle-based decomposition is examined for optically thick and thin cases in an infinite medium problem. The results are shown for one time step and for multiple timesteps. The case for one timestep isolates the effect of RMA operations on scaling since there is no need for census rebalancing. Multiple timesteps show the influence of census rebalancing on scaling and the influence of transport the census particles that accumulate from timestep to timestep. To truly test the weak scaling behavior of the algorithms, the number of cells per rank is kept constant.

The particle-based parallel method is compared to the standard particle-passing method for an optically thick problem. The first case is shown in Table (6.1). In this case, particles can travel a total of 3.0 cm in a timestep, or three mesh cell widths. The mean free path between scattering interactions is about 0.012 cm. In this case, sub-domain boundary crossings are common for particles born on the edge of the sub-domain but particles will not penetrate deep into other sub-domains because of the timestep size. The weak scaling results for this case with one timestep are shown in Fig. (6.8a) and for all ten timesteps in Fig. (6.8b). The total number of RMA operations per rank are shown in Fig. (6.9). For this case, the weak efficiency for one timestep is about 88% at 256 cores with the non-blocking implementation and about 50% with the window implementation. When the problem is run for ten timesteps the weak scaling of the non-blocking implementation is not affected—the weak scaling at 256 cores is still about 88%. The scaling of the window implementation is reduced to about 40% when multiple timesteps are used. Both implementations use the same method for rebalancing the census photons. The fact that the particle rebalancing does not impact the scaling of the non-blocking implementation suggests that this is not the cause of the decrease in scaling efficiency present in the window implementation. Multiple timesteps does increase the number of particles simulated per timestep (no population control on the particles is used in this scaling study). The decrease in weak scaling efficiency suggests that the window implementation does not scale well with an increased number of particles per core, likely due to the blocking nature of its parallel communication. When comparing the RMA operations graphs, it is important to note that the non-blocking implementation is not making remote memory requests to a memory window instead we use “RMA operation” to mean “request for a remote piece of data.” In both implementations an RMA operation is incurred when a particle requires data that is not available or is not in the process of being requested. In Fig. (6.9) the non-blocking implementation makes more requests for remote memory than the window implementation. This difference is likely due to mesh partitioning and in this case, the increased number of RMA operations does not affect the performance of non-blocking implementation relative to the window implementation. The size of the working mesh, which is implemented as a map data type, does not affect the scaling results for this case. This is because the total number of RMA operations per core is less than the size of the working mesh (for each working mesh size used: 5000, 10000 and 50000). Unfortunately, The working mesh cannot be used for multiple timesteps. This is because the opacity and Fleck factor will change after the temperature of a remote cell is updated by its remote host at the end of the timestep. This means that the copy of the cell data that is stored in the working mesh is no longer valid.

The flat weak scaling beyond 8 cores for the non-blocking implementation and beyond 8 cores in the window implementation show that there is some initial cost in the parallel overhead but that cost does not significantly increase as the number of processors is increased to 256.

A second case with physical scattering and a lower absorption opacity emphasizes scattering and streaming behavior, the parameters are shown in Table (6.2). For this case, the mean free path between scattering events is longer (about 0.08 cm) and we thus expect the photon to be able to travel farther from local memory. The weak

Parameter		Value
particles/rank	$=$	2.0×10^6
Δt	$=$	0.1 ns
t	$=$	1.0 ns
cells/rank	$=$	25^3
$\Delta x, \Delta y, \Delta z$	$=$	1.0 cm
σ_a	$=$	100.0 cm^{-1}
f	$=$	0.1955
$f\sigma_a$	$=$	19.5 cm^{-1}
$(1-f)\sigma_a$	$=$	80.44 cm^{-1}
σ_s	$=$	0.0 cm^{-1}
c_v	$=$	1.0 $\frac{\text{jk}}{\text{g keV}}$
$T_{m,0}$	$=$	1.0 keV
$T_{r,0}$	$=$	1.0 keV

Table 6.1: Parameters of first case: a thick test problem with no physical scattering

scaling results for this case are shown in Fig. (6.10a) and Fig. (6.10b). The RMA operations are shown in Fig. (6.11). The weak scaling results are very similar to those for the first case for the two larger working mesh sizes. The smallest working mesh size of 5000 cells is now less than the total number of RMA operations per core in the non-blocking method. Managing memory and evicting data from the working mesh causes the weak scaling efficiency to decrease to about 43% at 256 processors. The window method never reaches the point where the number of more RMA operations are larger than the size of the working mesh and its scaling is not affected. The same affect on scaling is seen when multiple timesteps are used, only the decrease is seen at a smaller number of cores. This is most likely due to the additional census photons being transported—these photons will travel farther in a timestep (they start at $t = t^n$, forming the initial radiation conditions). In this test IMC code, the census photons are not sorted by cell and grouped with the emission photons from the same cell. This means that memory requested by the emission

Parameter		Value
particles/rank	$=$	2.0×10^6
Δt	$=$	0.1 ns
t	$=$	1.0 ns
cells/rank	$=$	25^3
$\Delta x, \Delta y, \Delta z$	$=$	1.0 cm
σ_a	$=$	10.0 cm^{-1}
f	$=$	0.7085
$f\sigma_a$	$=$	7.085 cm^{-1}
$(1-f)\sigma_a$	$=$	2.915 cm^{-1}
σ_s	$=$	10.0 cm^{-1}
c_v	$=$	$1.0\ \frac{\text{jk}}{\text{g keV}}$
$T_{m,0}$	$=$	1.0 keV
$T_{r,0}$	$=$	1.0 keV

Table 6.2: Parameters for the second case: a thick test problem with physical scattering

particles in a cell is less likely to be present when it is needed by the census photons from that same cell. This emphasizes the importance of sorting the census photons and grouping particles that are likely to have similar histories together.

The differences between the weak scaling results for one timestep and ten timesteps are more pronounced in the second case. Because the mean free path between scattering events is longer, a particle is more likely to end its history on a non-local mesh cell, meaning there is more particles that are passed in the rebalance step. This problem also takes less overall runtime because there is less overall scattering, meaning that the census particle rebalance takes more time relative to the particle transport.

For the thin case, the same mesh is used as the previous cases but the timestep size is increased and the absorption opacity is decreased, as shown in Table (6.3). Now particles can travel a distance of 10 cell widths within a timestep and have a mean free path of about 8.0 cm between scattering interactions. In this case we expect more RMA operations and worse weak scaling efficiency relative to the the particle-decomposed method. The weak scaling results for the thin case are shown in Fig. (6.12a) and Fig. (6.12b). The RMA operations are shown in Fig. (6.16). The results show that weak scaling is significantly improved by having a large enough working mesh to store non-local mesh. The non-blocking method is especially sensitive to the size of the working mesh in the optically thin regime—unless the largest mesh size is used, the weak scaling immediately degrades to below 10% efficiency. The window method is not impacted as severely, although the weak scaling is poor compared to particle passing method. The ability of the window implementation to finish the simulation in a reasonable time is likely because particle histories are forced to wait for data and are thus allowed to complete their history. In the non-blocking implementation, all particles waiting for data are moved until they need data again. If the number of particles waiting is large enough they may be waiting on more data than the size of the working mesh. This causes these particles to request the same data over and over again until other particles finish their histories.

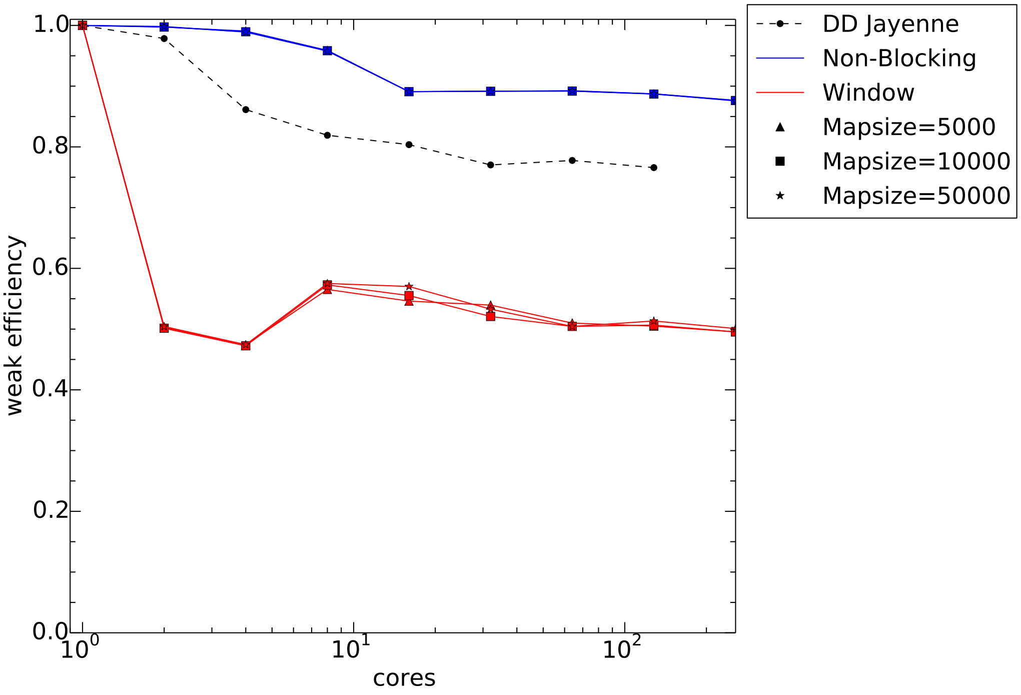
As expected, both implementations of the particle-based method are inferior to the particle passing method as the problem becomes more optically thin. Both implementations of the particle-based method could be improved by requesting addition mesh at a given time. Both implementations are operating in regime that is straining the latency of the network—this is especially true of the window implementation because it requests one mesh cell at a time. If the neighbors of a cell were also delivered by a remote host, a particle could potentially move further before requesting data again.

Parameter		Value
particles/rank	$=$	2.0×10^6
Δt	$=$	0.33 ns
t	$=$	3.33 ns
cells/rank	$=$	25^3
$\Delta x, \Delta y, \Delta z$	$=$	1.0 cm
f	$=$	0.879
$f\sigma_a$	$=$	0.879 cm^{-1}
$(1-f)\sigma_a$	$=$	0.121 cm^{-1}
σ_s	$=$	0.0 cm^{-1}
c_v	$=$	$1.0\ \frac{\mathrm{jk}}{\mathrm{g\ keV}}$
$T_{m,0}$	$=$	1.0 keV
$T_{r,0}$	$=$	1.0 keV

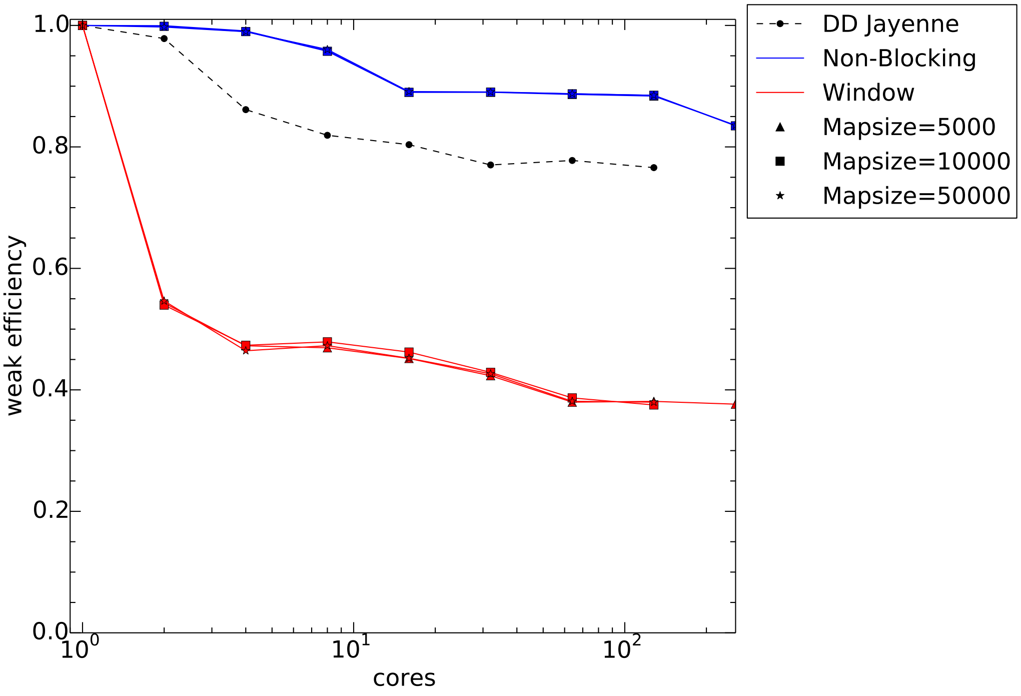
Table 6.3: Parameters of the thin test problem with no physical scattering

6.5.4 Stratified Sampling

In thin cases, a single particle can stream through many mesh cells. If this occurs off of the memory owned by the parallel process, it will incur many RMA operations. This is made worse by the fact the next particle could stream through entirely different mesh cells, which would incur more RMA operations and possible evict memory that could be used by the next particle. In an ideal situation, memory use would be synchronized such that a group of particles would use the same memory during a timestep. Because Monte Carlo uses random numbers to determine particle histories, grouping particles that will have the same interactions is difficult. One way to improve memory access patterns is *stratified sampling*. In stratified sampling, the phase space is divided into a number of discrete bins and then samples are taken from within those bins. Stratified sampling reduces variance because samples are more uniform. To achieve the largest reduction in variance, the number of bins should be equal to the number of samples. An example of Monte Carlo numerical integration with and without stratified sampling is shown in Fig. (6.14). In this simple example,

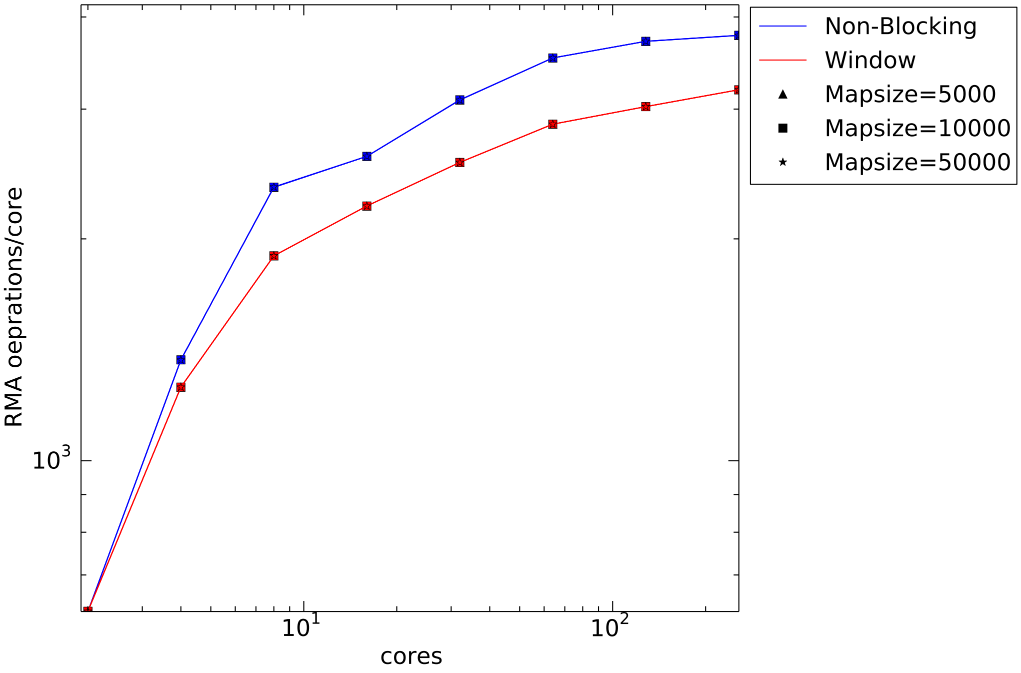


(a) Weak scaling in the infinite medium problem with 1 timestep

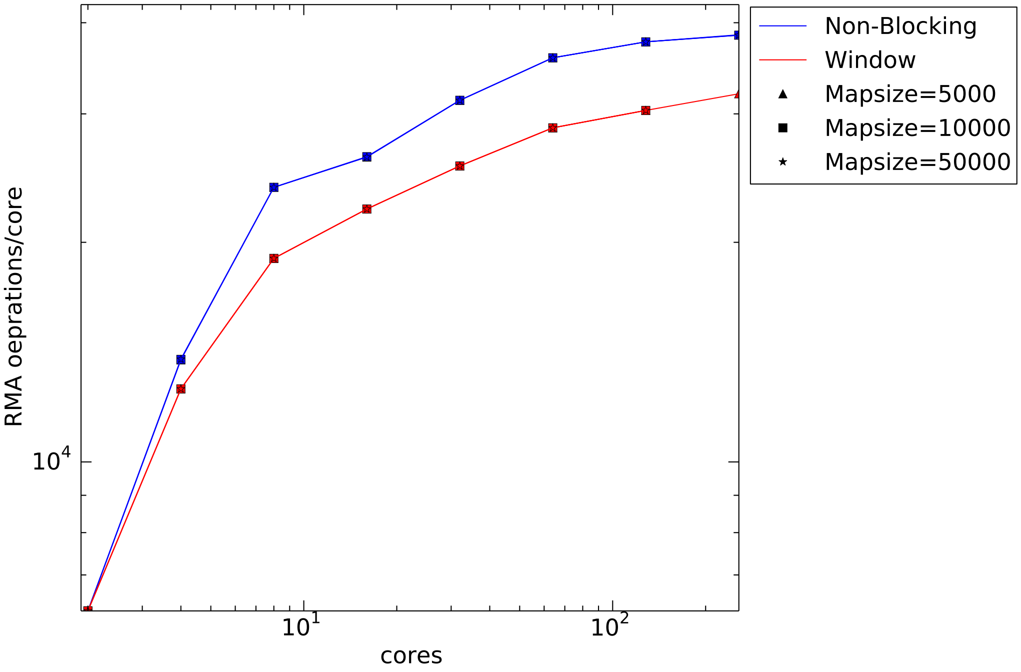


(b) Weak scaling in the infinite medium problem with 10 timesteps

Figure 6.8: Weak scaling for the optically thick infinite medium problem for various sizes of working mesh

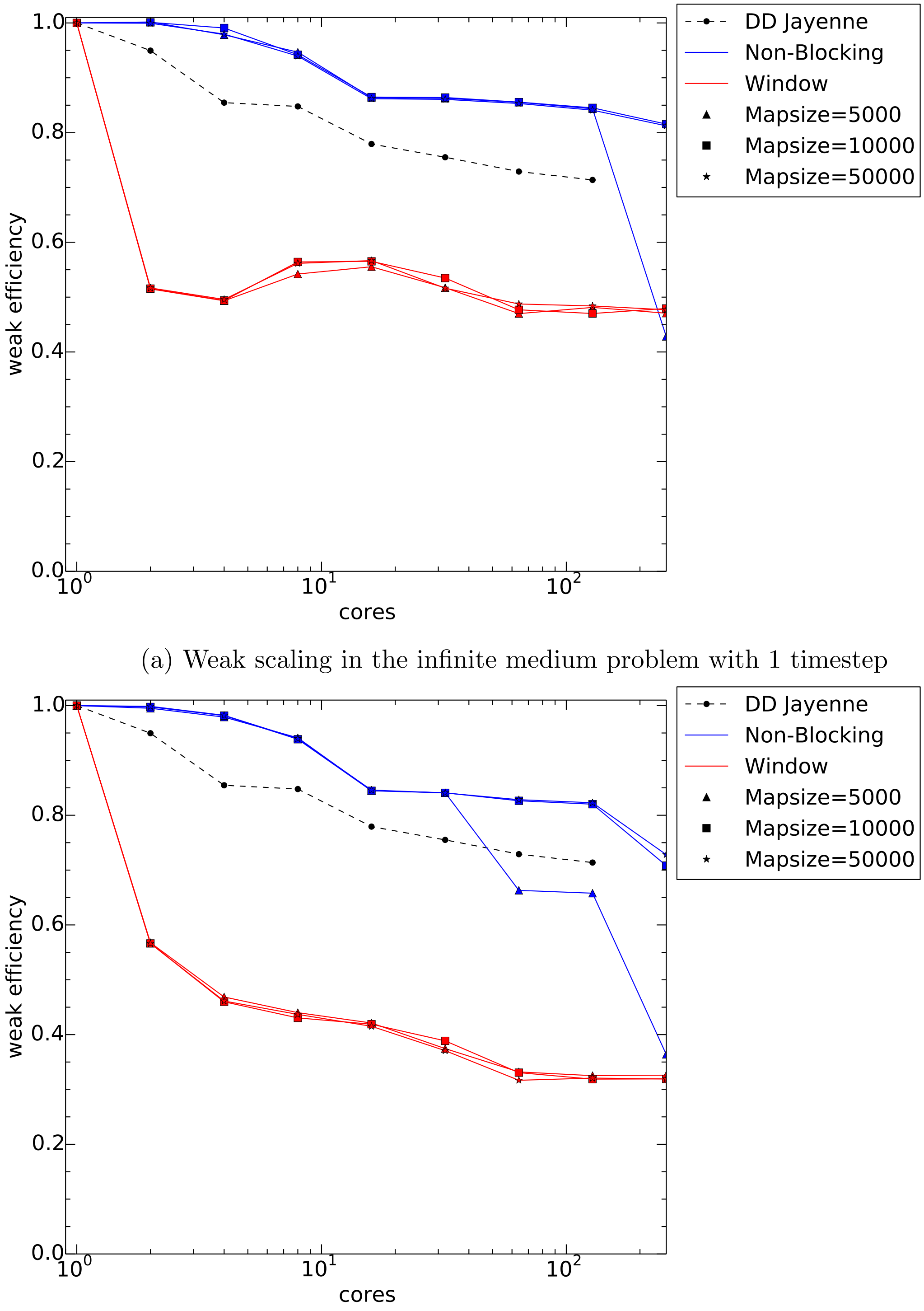


(a) Total RMA operations per rank for various working mesh sizes with one timestep



(b) Total RMA operations per rank for various working mesh sizes with ten timesteps

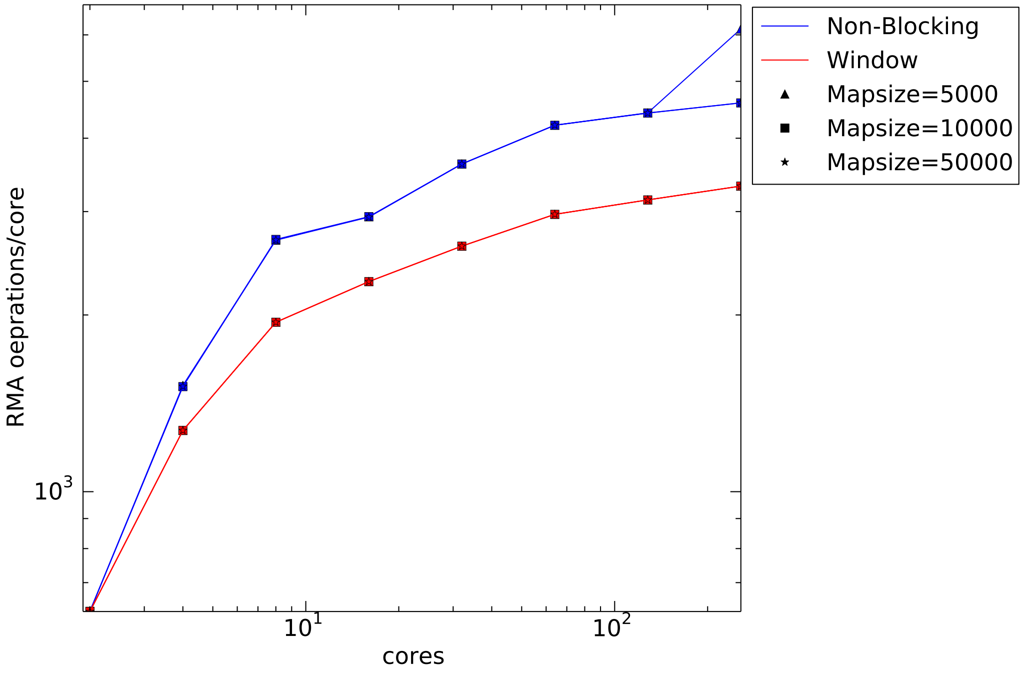
Figure 6.9: Total RMA operations per rank for the optically think infinite medium problem



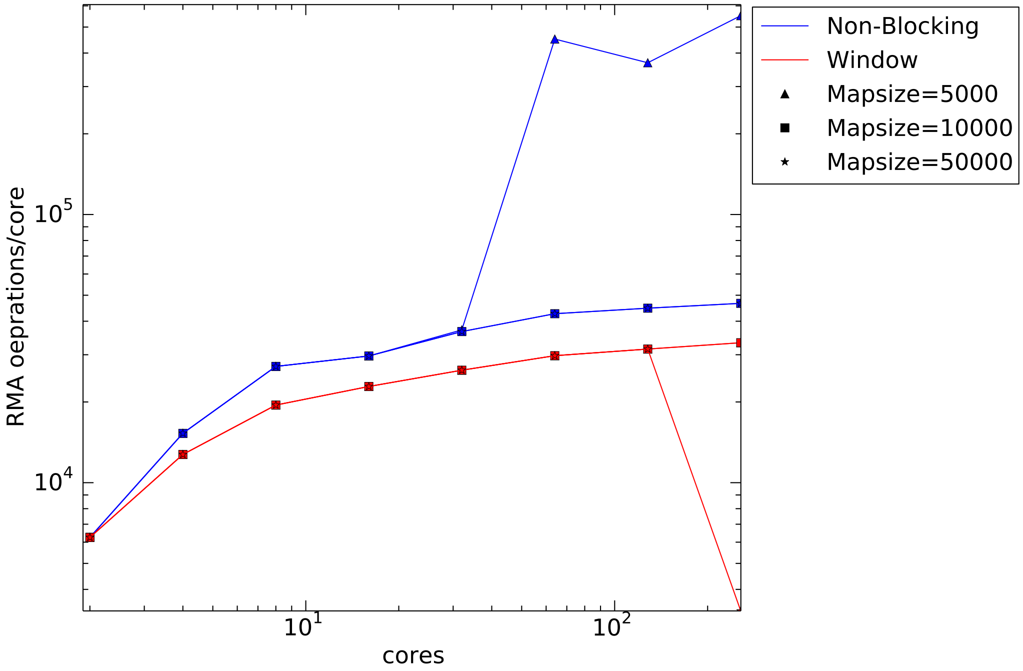
(a) Weak scaling in the infinite medium problem with 1 timestep

(b) Weak scaling in the infinite medium problem with 10 timesteps

Figure 6.10: Weak scaling results with various working mesh sizes for the optically thick infinite medium problem with physical scattering

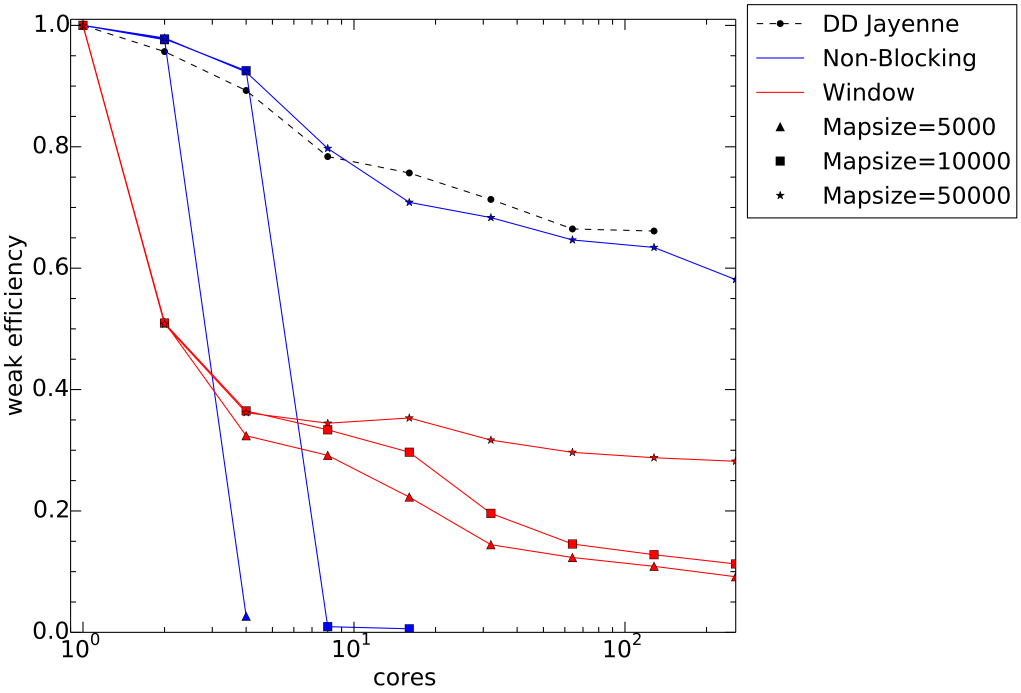


(a) Total RMA operations per rank for various working mesh sizes with one timestep

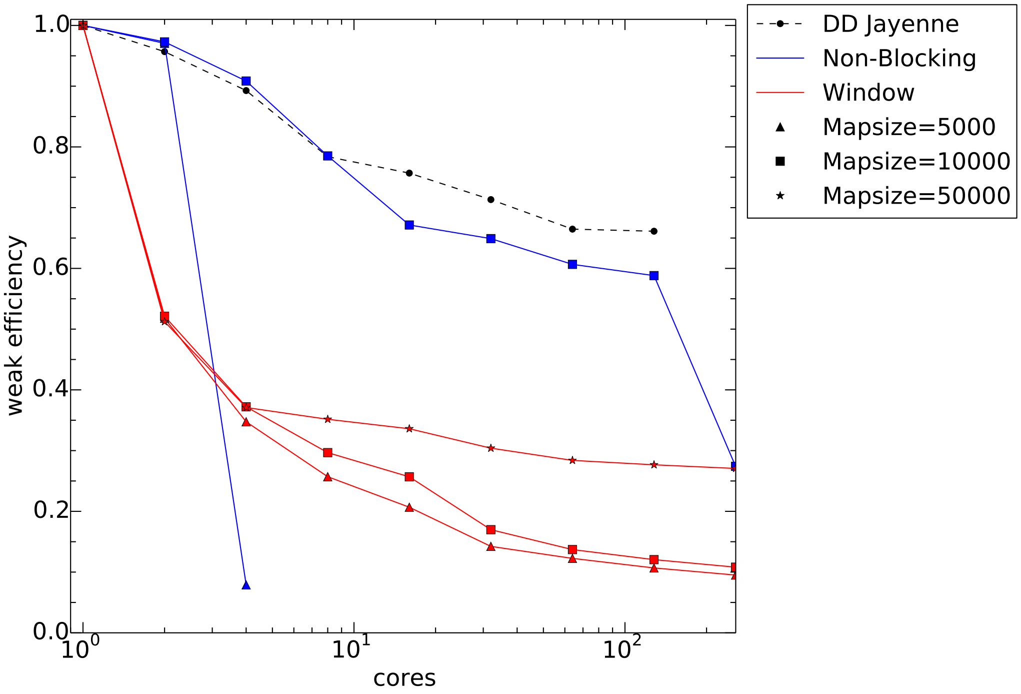


(b) Total RMA operations per rank for various working mesh sizes with ten timesteps

Figure 6.11: Total RMA operations per rank for the optically thick infinite medium problem with physical scattering

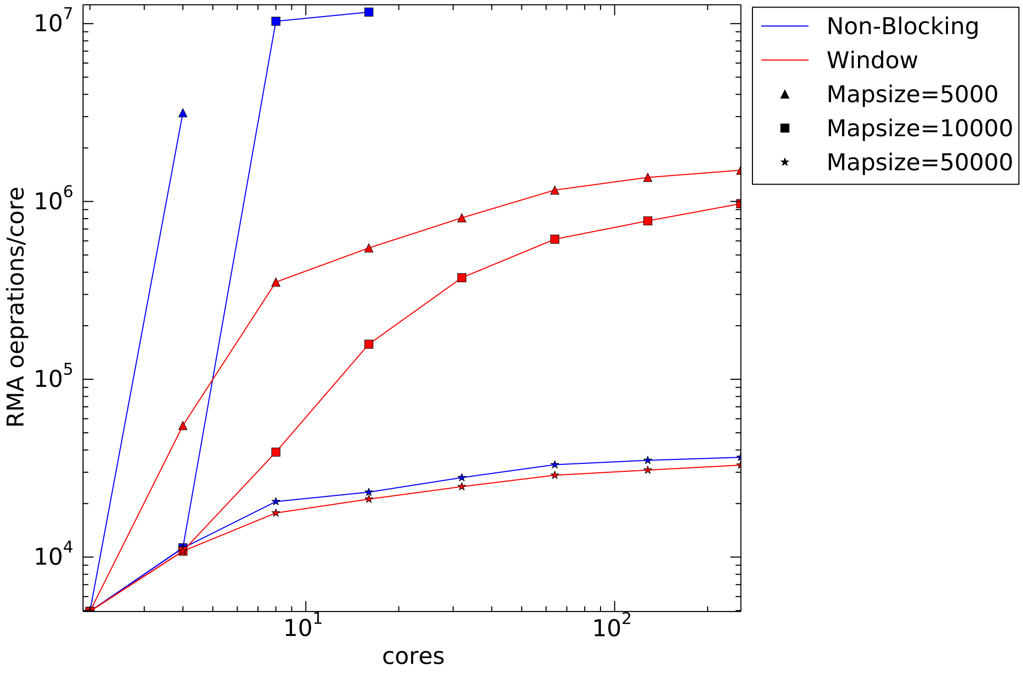


(a) Weak scaling in the optically thin infinite medium problem with 1 timestep

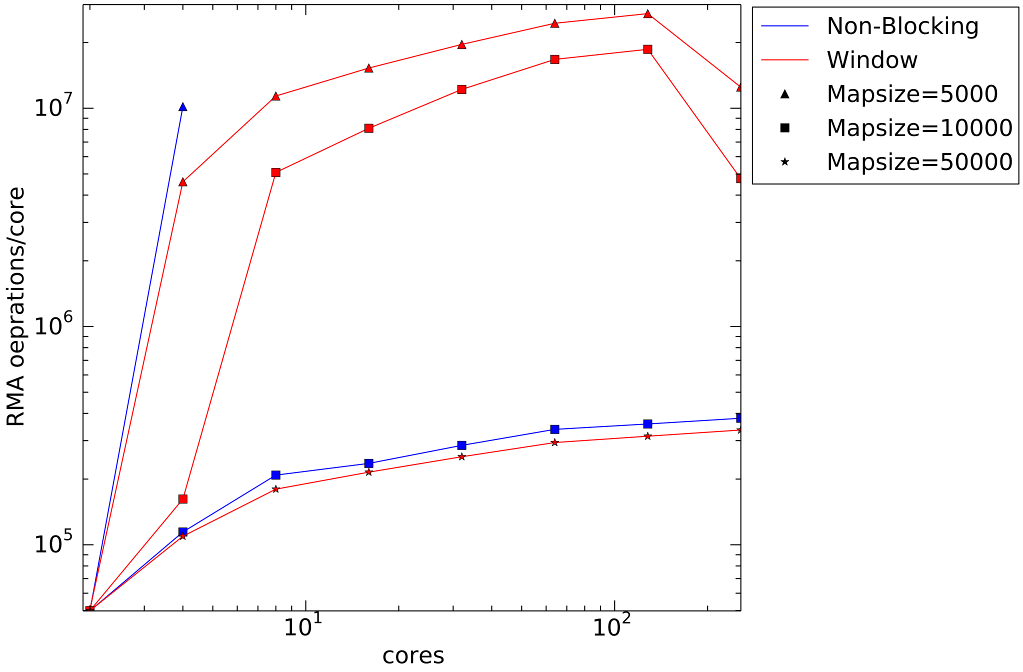


(b) Weak scaling in the optically thin infinite medium problem with 10 timesteps

Figure 6.12: Weak scaling results with various working mesh sizes for the optically thin infinite medium problem



(a) Total RMA operations per rank for various working mesh size with one timestep



(b) Total RMA operations per rank for various working mesh sizes with ten timesteps

Figure 6.13: Total RMA operations per rank for the optically thin infinite medium problem

the error in the integral is reduced by a factor of about 50 by using stratified sampling with 100 strata.

Using stratified sampling to reduce variance in IMC was suggested by Long and McClarren [36]. The reduction in variance is not the main benefit that stratified sampling can provide to the particle-based domain decomposition method—instead stratified sampling can be used to ensure that consecutive particles in memory will have roughly the same history. To obtain this benefit from stratified sampling, the angle of an emission particle is stratified by octant. The case of particle angle with and without stratified sampling is shown in Fig. (6.15). In Fig. (6.15b), particles P_2 and P_3 are likely to access the same mesh data, especially if the region is optically thin.

As an example, there are 32 particles to represent the emission energy in a cell. The first 4 particles will be sampled from the first octant ($\theta \subset [0, \pi/2], \phi \subset [0, \pi/2]$), the second four particles from the second octant and so on. Now particles that are adjacent in memory are much more likely to use the same mesh data.

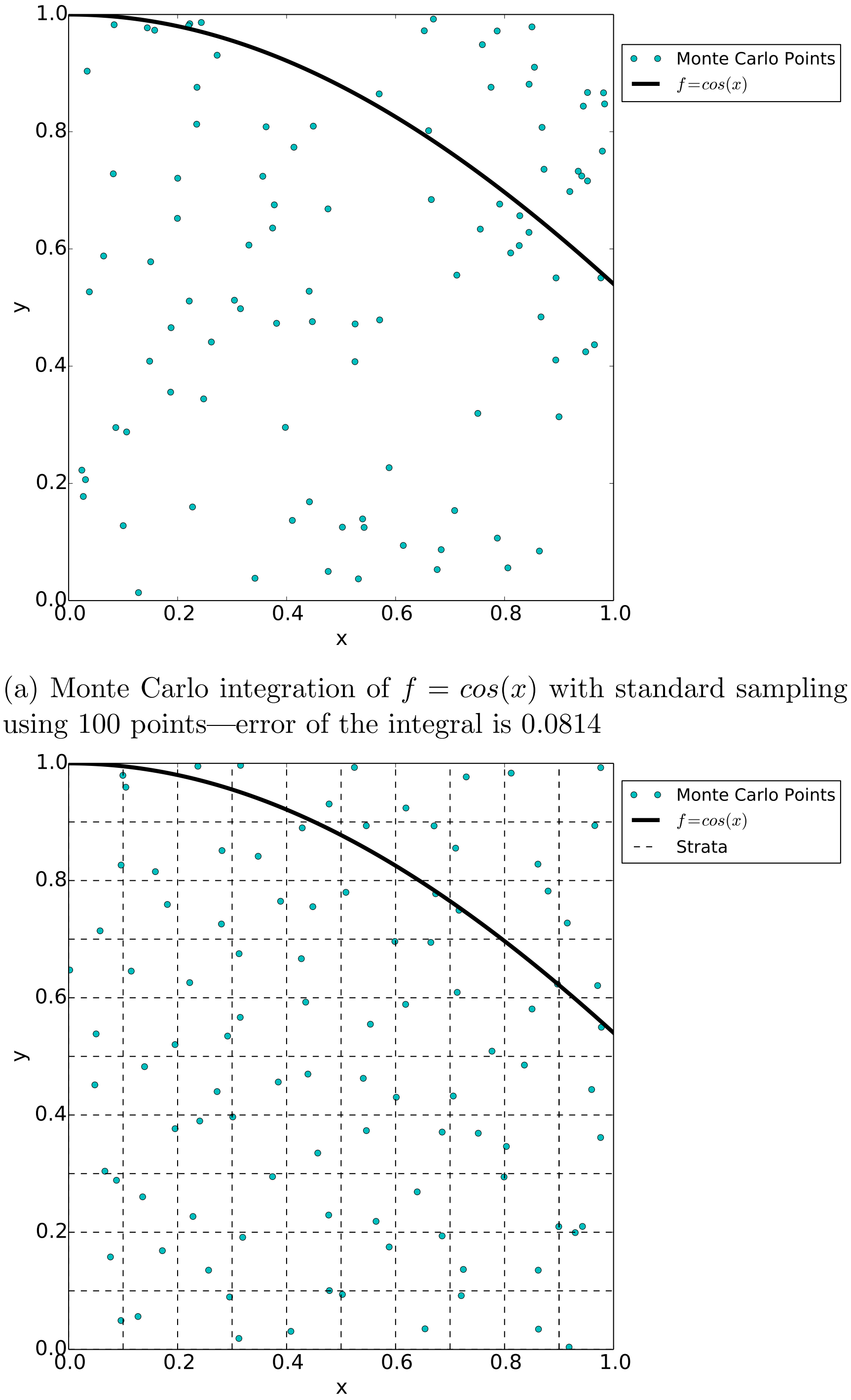
Stratified sampling is likely to provide the most benefit when the problem is optically thin and the working mesh size is small. To test the effects of stratified sampling, the thin case in Table (6.3) was run. The results are shown in Fig. (6.16a). The RMA operations are shown in Fig. (6.16b). Currently, stratified sampling does not have a significant effect on the weak scaling for the window implementation. This is likely due to the fact that particles within a cell are not likely to evict mesh memory that is needed by other particles in that cell. If the working mesh size were very small, which is not advised based on the results above, stratified sampling could have a larger effect on runtime. For the non-blocking implementation, stratified sampling reduces the runtime for the case of 16 processors with a working mesh size of 10000 from 586 seconds to 450 seconds. This case still takes much longer to run than the 50000 working mesh size case but it does show that stratified sampling can have a positive on the runtime. The benefit is seen in the non-blocking implementation because it appears to be sensitive to the case where are large number of particles are waiting on disparate data.

A variant of stratified sampling would be to group all of the particles on a processor that have a direction in the same octant and organize them to be adjacent in memory. Sorting particles could be expensive but if the RMA operations are truly the limit on scaling and performance it could be an advantageous trade off.

6.6 Discussion

Both the on-node and domain-decomposed algorithms in this chapter show promise as a means to improve the performance of IMC on large, non-uniform HEDP problems. The particle-based method performs well on optically thick problems compared to the particle-passing method. Multiple methods were suggested as a means to improve the scaling in the optically thin regime, but as an aside, the performance in the thick regime could arguably be considered more important. The reasoning behind this is that most work is performed in energetic regions of the problem, which are optically thick and highly scattering. The portions of the problem that are very optically thin should have a lower mesh resolution away from interfaces because the solution will not significantly change in those areas. The on-node algorithms could be paired with domain decomposed algorithms to increase the size of the local domain, reducing the surface area to volume ratio and possibly reducing overall parallel communication.

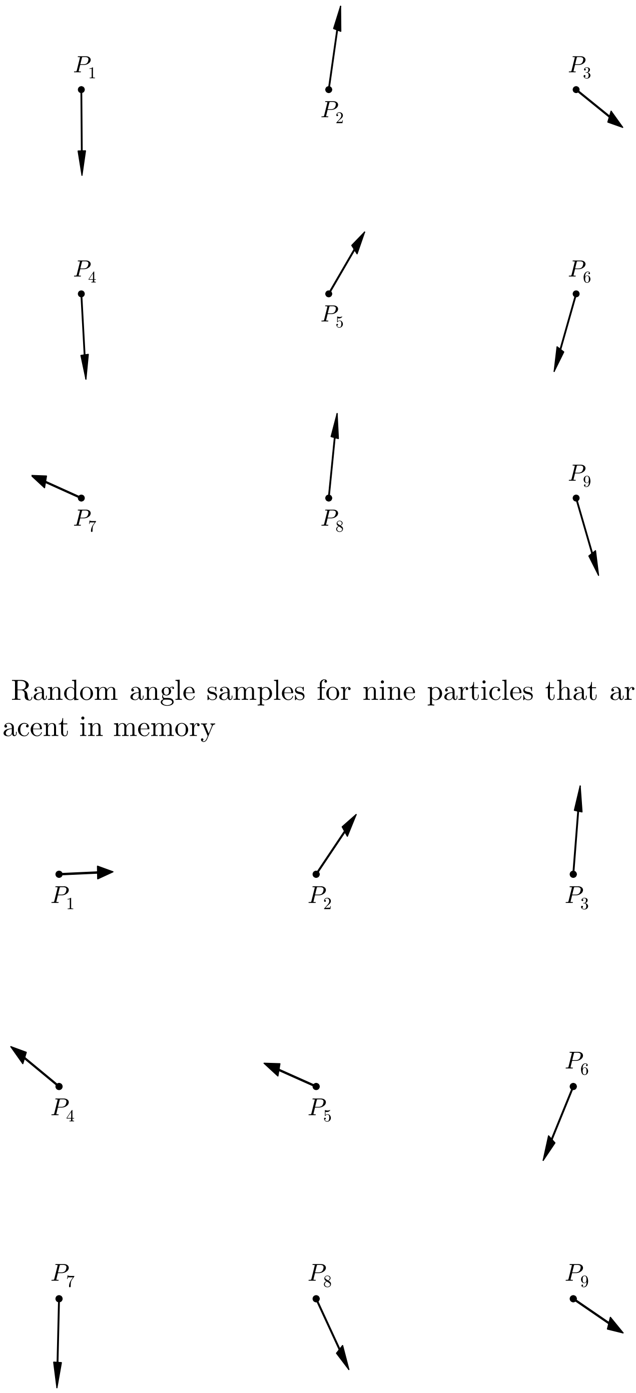
The main challenge of the particle-based method is developing a load-balancing algorithm that scales well to a large number of processors. The challenge for on-node parallel methods is to utilize the vector registers of modern HPC machines.



(a) Monte Carlo integration of $f = cos(x)$ with standard sampling using 100 points—error of the integral is 0.0814

(b) Monte Carlo integration of $f = cos(x)$ with stratified sampling using 100 points and 100 strata—error of the integral is 0.0014

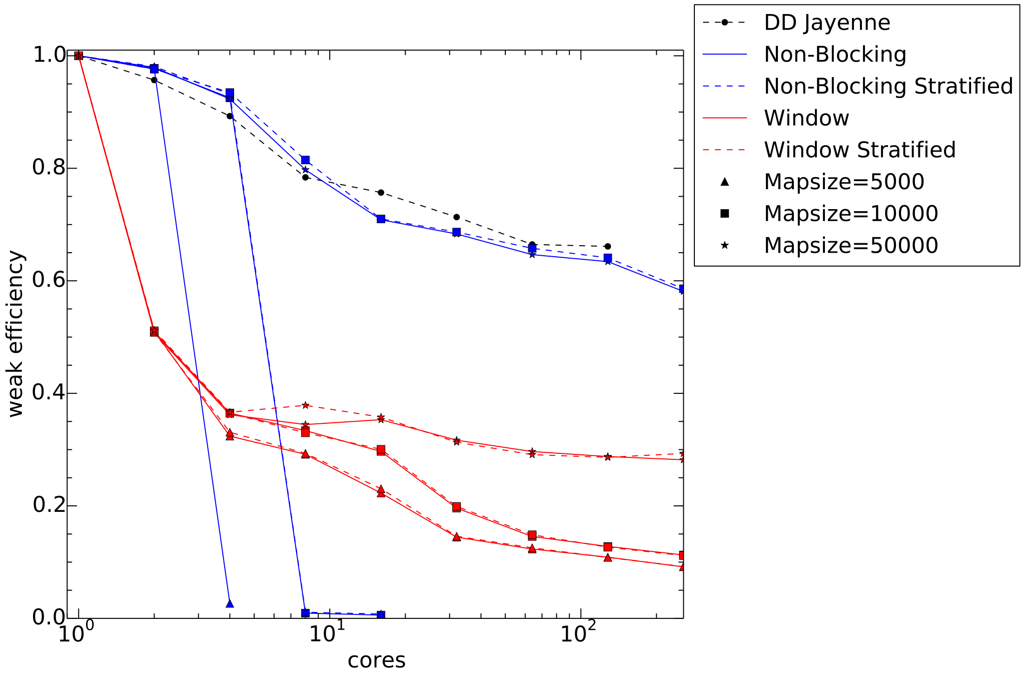
Figure 6.14: Monte Carlo integration of the function $f = cos(x)$ with 100 points with standard sampling and stratified sampling



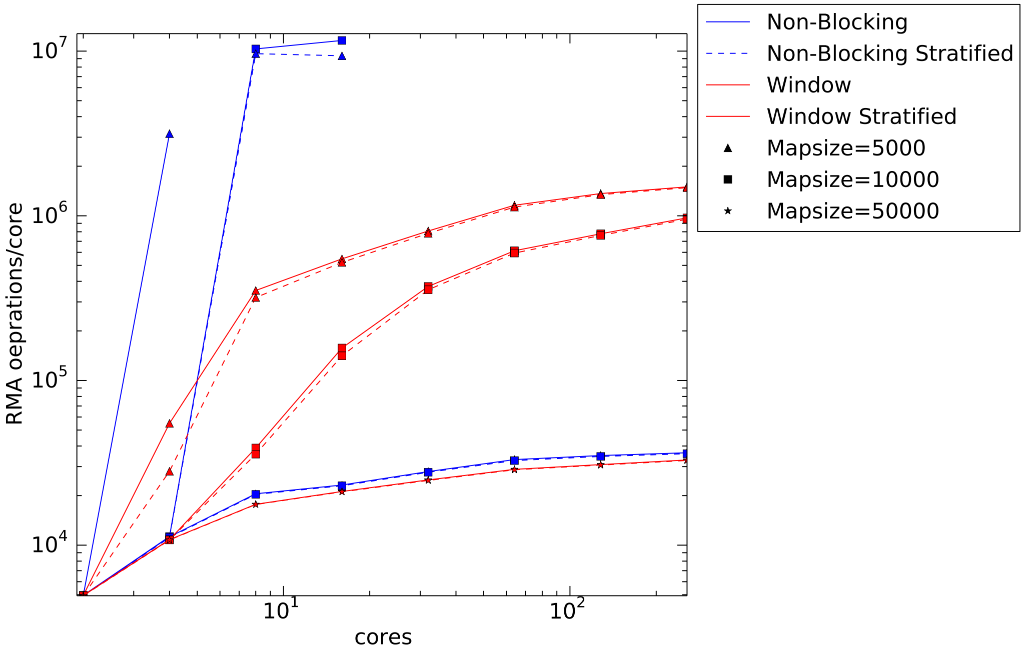
(a) Random angle samples for nine particles that are adjacent in memory

(b) Stratified azimuthal angle samples for nine particles that are adjacent in memory

Figure 6.15: Comparison of particle angles with and without stratified sampling



(a) Weak scaling in the infinite medium problem with 1 timestep for the thin case with stratified sampling



(b) Total RMA operations per rank for various working mesh size with one timestep

Figure 6.16: Total RMA operations per rank with various sizes of working mesh for the optically thin infinite medium problem with stratified sampling

7. CONCLUSIONS AND FUTURE WORK

7.1 IMC in the CHICOMA Hydro Code

IMC was adapted to work with node-based unknowns on unstructured tetrahedral meshes. Adapting the IMC algorithm to this context was done by avoiding the cost of tracking and sampling on the complicated dual mesh geometry. Using a 3D, unstructured tetrahedral mesh yields two problems: the first is that boundary sources can yield asymmetrical wavefronts when the material is very optically thick. This problem appears to be an issues of unstructured meshes in general. The second problem is that refining material interfaces is difficult in three dimensions without significantly increasing the number of cells in the simulation. This interface resolution has been solved in other contexts but is not address directly here.

7.2 Spatial Discretizations of IMC

The corner-based method is the most accurate and robust method studied for running on IMC on node-based, unstructured tetrahedral meshes. The node-centered method with the emission upwinding technique is equivalent to transporting on the dual mesh directly, but using the corner-based method provides a more accurate while requiring only slightly more memory. The DFEM IMC method gives solutions that are too diffusive when a piecewise linear opacity is used on optically thick problems and yields a similar solution to the corner based method when an averaged opacity is used. Although the implicit capture method of variance reduction is not as simple in DFEM IMC, it can be done accurately using a Gaussian quadrature integration technique when the optical depth of a single particle path is less than 10. The DFEM method does show promise as a means of reducing simulation runtime by eliminating the need to track to the dual mesh cell interface. An equilibrium diffusion limit analysis is likely necessary to determine how the opacity and emission should be treated in the DFEM IMC method, or a method that allows for a robust solution that does not require lumping of the mass matrix. A potential benefit of DFEM IMC methods is that they could employ vectorization in the partilce transport loop—evaluating the implicit capture absorption for each test function in a cell is a simple loop that could benefit from vectorization.

7.3 Parallel Algorithms

The amount of memory per core is trending downward as low power, manycore architectures enter the landscape. The on-node IMC algorithms attempts to exploit parallelism in the particle transport and respect the memory hierarchy of the compute node. Shared-memory systems can be effectively used in parallel and can benefit simulations that are too large to be replicated across all cores. Shared memory algorithms could also allow sub-domains to be larger when used in a hybrid configuration with a domain-decomposed IMC algorithm.

Domain decomposition is difficult to do because it must load-balance the problem and perform well in optically thick and optically thin regimes. A new method for domain decomposition in IMC was explored that focuses on the moving mesh data instead of particle data. This method is better than the particle-passing method for optically thick problems with a large amount of scattering.

The particle-based algorithm is a new idea and there are many ways it could be improved. The first and foremost issue is to develop a load-balancing algorithm that allows processors with many particles to share their work. The memory management of the working mesh could be improved drastically such that RMA operations could be reduced for thick and thin problems. Parallel communication and computational work is effectively balanced using the non-blocking MPI implementation of the particle-based decomposition method. The window implementation is currently less effective than the non-blocking implementation but it could be improved by using non-blocking RMA operations and utilizing lower level RDMA operations on architectures where they are available. Deferred parallel communication could also be used to improve the particle rebalancing that occurs at the end of a timestep. Stratified sampling does improve the parallel performance for cases where a small working mesh is used. The larger implication of the stratified sampling results are that similar particle histories should be grouped as close as possible in memory.

7.4 Broad Strokes

The IMC method is known as a standard solution in TRT simulations because it is robust and accurate. Because it has been used for so long, its defects and difficulties are well known. In this work we attempted to address the spatial discretization errors and pave the way to use IMC in the most effective way on a specific mesh type. We also examined ways to reduce the high cost of running IMC on large HPC machines. These improvements will hopefully allow IMC to be used with emerging hydrodynamics methods on the computer architectures of the future.

REFERENCES

[1] Marvin L Adams. Even-parity finite-element transport methods in the diffusion limit. *Progress in Nuclear Energy*, 25(2):159–198, 1991.

[2] Marvin L Adams. Subcell balance methods for radiative transfer on arbitrary grids. *Transport Theory and Statistical Physics*, 26(4-5):385–431, 1997.

[3] Eugene D. III Brooks. Symbolic Implicit Monte Carlo. *Journal of Computational Physics*, 83(2):433 – 446, 1989.

[4] Forrest B. Brown and William R. Martin. Monte Carlo methods for radiation transport analysis on vector computers. *Progress in Nuclear Energy*, 14(3):269 – 299, 1984.

[5] Forrest B Brown and William R Martin. Direct sampling of Monte Carlo flight paths in media with continuously varying cross-sections. In *Proceedings of ANS Mathematics & Computation Topical Meeting*, volume 2, 2003.

[6] Thomas Brunner. Forms of approximate radiation transport. Technical Report SAND2002-1778, Sandia National Laboratory, Albuquerque, New Mexico 87185, 2002.

[7] Thomas A. Brunner, Todd J. Urbatsch, Thomas M. Evans, and Nicholas A. Gentile. Comparison of four parallel algorithms for domain decomposed implicit monte carlo. *Journal of Computational Physics*, 212(2):527 – 539, 2006.

[8] L. L. Carter and C. A. Forest. Nonlinear radiation transport simulation with an Implicit Monte Carlo method. Technical Report LA-5038, Los Alamos National Laboratory, 100, 1973.

[9] Marc RJ Charest, Clinton PT Groth, and Pierre Q Gauthier. High-order ceno finite-volume scheme for low-speed viscous flows on three-dimensional unstructured mesh. *Convergence*, 59:60, 2012.

[10] J. R. Cheatham. *Truncation Analysis and Numerical Method Improvements for the Thermal Radiative Transfer Equations.* PhD thesis, University of Michigan, Ann Arbor, Michigan, 2010.

[11] Mathew A. Cleveland, Nick A. Gentile, and Todd S. Palmer. An extension of implicit monte carlo diffusion: Multigroup and the difference formulation. *Journal of Computational Physics*, 229(16):5707 – 5723, 2010.

[12] Plasma Science Committee et al. *Frontiers in High Energy Density Physics: The X-Games of Contemporary Science.* National Academies Press, 2003.

[13] Jeffery D Densmore. Asymptotic analysis of the spatial discretization of radiation absorption and re-emission in implicit monte carlo. *Journal of Computational Physics*, 230(4):1116–1133, 2011.

[14] Jeffery D Densmore and Edward W Larsen. Asymptotic equilibrium diffusion analysis of time-dependent monte carlo methods for grey radiative transfer. *Journal of Computational Physics*, 199(1):175–204, 2004.

[15] Jeffery D. Densmore, Todd J. Urbatsch, Thomas M. Evans, and Michael W. Buksas. A hybrid transport-diffusion method for monte carlo radiative-transfer simulations. *Journal of Computational Physics*, 222(2):485 – 503, 2007.

[16] Luc Devroye. *Non-Uniform Random Variate Generation.* Springer-Verlag New York Inc., 1986.

[17] Veselin A Dobrev, Tzanio V Kolev, and Robert N Rieben. High-order curvilinear finite element methods for lagrangian hydrodynamics. *SIAM Journal on*

Scientific Computing, 34(5):B606–B641, 2012.

[18] Tingxing Dong, Veselin Dobrev, Tzanio Kolev, Robert Rieben, Stanimire Tomov, and Jack Dongarra. Hydrodynamic computation with hybrid programming on cpu-gpu clusters. *Innovative Computing Laboratory, University of Tennessee*, 2013.

[19] R Paul Drake. *High-energy-density physics: fundamentals, inertial fusion, and experimental astrophysics*. Springer Science & Business Media, 2006.

[20] J. A. Fleck. The calculation of nonlinear radiation transport by a Monte Carlo method. *Methods in Computational Physics*, 1:46–65, 1963.

[21] J.A Fleck and E.H Canfield. A random walk procedure for improving the computational efficiency of the implicit monte carlo method for nonlinear radiation transport. *Journal of Computational Physics*, 54(3):508 – 523, 1984.

[22] N. A. Gentile, N. Keen, and J. Rathkopf. The kull imc package. Technical Report UCRL-JC-132743, Lawrence Livermore National Laboratory, 100, 1998.

[23] N.A. Gentile. Implicit Monte Carlo diffusion–an acceleration method for Monte Carlo time-dependent radiative transfer simulations. *Journal of Computational Physics*, 172(2):543 – 571, 2001.

[24] W Goad and R Johnston. A monte carlo method for criticality problems. *Nuclear Science and Engineering*, 5(6):371–375, 1959.

[25] F. Graziani and J. LeBlanc. The crooked pipe test problem. Technical Report UCRL-MI-143393, Lawrence Livermore National Laboratory, 100, 2000.

[26] Rinku Gupta, Pavan Balaji, Dhabaleswar K Panda, and Jarek Nieplocha. Efficient collective operations using remote memory operations on via-based clus-

ters. In *Parallel and Distributed Processing Symposium, 2003. Proceedings. International*, pages 9–pp. IEEE, 2003.

[27] Michael T Heath. A tale of two laws. *International Journal of High Performance Computing Applications*, 29(3):320–330, 2015.

[28] Nicole Hemsoth. Full details uncovered on chinese top supercomputer, June 2013.

[29] Nicole Hemsoth. First details emerge from cray on trinity supercomputer, July 2014.

[30] Adam Irvine. Reducing teleportation error in IMC. Presentation LLNL-PRES-442111, Lawrence Livermore National Laboratory, July 2010.

[31] EL Kaplan. Monte carlo methods for equilibrium solutions in neutron multiplication. Technical report, California. Univ., Livermore. Lawrence Radiation Lab., 1958.

[32] George Karypis and Vipin Kumar. Metis-unstructured graph partitioning and sparse matrix ordering system, version 2.0. 1995.

[33] Daniel Kasen, RC Thomas, and Peter Nugent. Time-dependent monte carlo radiative transfer calculations for three-dimensional supernova spectra, light curves, and polarization. *The Astrophysical Journal*, 651(1):366, 2006.

[34] EW Larsen, GC Pomraning, and VC Badham. Asymptotic analysis of radiative transfer problems. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 29(4):285–310, 1983.

[35] E. E. Lewis and Jr. D. P. Miller. *Computational Methods of Neutron Transport*. American Nuclear Society, Inc., Illinois, 1993.

[36] Alex R Long and Ryan G McClarren. Improved convergence rates in Implicit Monte Carlo simulations through stratified sampling. *Transactions of the American Nuclear Society*, 109(1):675–678, November 2013.

[37] Alex R. Long, Ryan G. McClarren, Jacob I. Waltz, and John G. Wohlbier. Implcit Monte Carlo adaptations for tetrahedral meshes with node-based unknwons. In *Proceedings of ANS M&C 2015*, Nashville, Tennessee, USA, April 19-23, 2015.

[38] Min Long, George C Jordan IV, Daniel R van Rossum, Benedikt Diemer, Carlo Graziani, Richard Kessler, Bradley Meyer, Paul Rich, and Don Q Lamb. Three-dimensional simulations of pure deflagration models for thermonuclear supernovae. *The Astrophysical Journal*, 789(2):103, 2014.

[39] Peter G Maginot, Jean C Ragusa, and Jim E Morel. Lumping techniques for dfem s n transport in slab geometry. *Nuclear Science and Engineering*, 179(2), 2015.

[40] RE Marshak. Effect of radiation on shock wave behavior. *Physics of Fluids (1958-1988)*, 1(1):24–29, 1958.

[41] R. G. McClarren and T. J. Urbatsch. A modified Implicit Monte Carlo method for time-dependent radiative transfer with adaptive material coupling. *Journal of Computational Physics*, 228(16):5669 – 5686, 2009.

[42] Michael Scott McKinley, Eugene D. Brooks III, and Abraham Szoke. Comparison of implicit and symbolic implicit monte carlo line transport with frequency weight vector extension. *Journal of Computational Physics*, 189(1):330 – 349, 2003.

[43] Mellanox Technologies Inc., 2900 Stender Way, Santa Clara, CA 95054. *Introduction to InfiniBand*. Rev. 1.9.

[44] Nicholas Metropolis and Stanislaw Ulam. The Monte Carlo method. *Journal of the American Statistical Association*, 44(247):335–341, 1949.

[45] Osamu Mori, Hirotaka Sawada, Ryu Funase, Mutsuko Morimoto, Tatsuya Endo, Takayuki Yamamoto, Yuichi Tsuda, Yasuhiro Kawakatsu, Jun'ichiro Kawaguchi, Yasuyuki Miyazaki, et al. First solar power sail demonstration by ikaros. *Transactions of the Japan Society for Aeronautical and Space Sciences, Aeropsace Technology Japan*, 8(ists27):To_4_25–To_4_31, 2010.

[46] S. W. Mosher and J. D. Densmore. Stability and monotonicity conditions for linear, grey, 0-D Implicit Monte Carlo calculations. *Transactions of the American Nuclear Society*, 93:520–522, 2005.

[47] T N'kaoua. Solution of the nonlinear radiative transfer equations by a fully implicit matrix monte carlo method coupled with the rosseland diffusion equation via domain decomposition. *SIAM Journal on Scientific and Statistical Computing*, 12(3):505–520, 1991.

[48] Tara M. Pandya, Seth R. Johnson, Gregory G. Davidson, Thomas M. Evans, and Steven P. Hamilton. Shift: A massively parallel monte carlo radiation transport package. Technical Report CASL-U-2015-0170-000, Oak Ridge National Laboratory, 2015.

[49] Shahyar Z Pirzadeh. Advanced unstructured grid generation for complex aerodynamic applications. *AIAA Journal*, 48(5):904–915, 2010.

[50] G. C. Pomraning. *The Equations of Radiation Hydrodynamics.* Pergamon, Oxford, 1973.

[51] Wm H Reed and TR Hill. Triangularmesh methodsfor the neutrontransporte-quation. *Los Alamos Report LA-UR-73-479*, 1973.

[52] Richard P Smedley-Stevenson and Ryan G McClarren. Asymptotic diffusion limit of cell temperature discretisation schemes for thermal radiation transport. *Journal of Computational Physics*, 2013.

[53] B. Su and G. L. Olson. An analytical benchmark for non-equilibrium radiative transfer in an isotropically scattering medium. *Annals of Nuclear Energy*, 24(13):1035 – 1055, 1997.

[54] Travis J. Trahan and Nicolas A. Gentile. Analytic treatment of source photon emission times to reduce noise in Implicit Monte Carlo calculations. *Transport Theory and Statistical Physics*, 41:265–283, 2012.

[55] John C Wagner, Scott W Mosher, Thomas M Evans, Douglas E Peplow, and John A Turner. Hybrid and parallel domain-decomposition methods development to enable monte carlo for reactor analyses. *Progress in Nuclear Science and Technology*, 2:815–820, 2011.

[56] J. Waltz, N.R. Morgan, T.R. Canfield, M.R.J. Charest, L.D. Risinger, and J.G. Wohlbier. A three-dimensional finite element arbitrary Lagrangian-Eulerian method for shock hydrodynamics on unstructured grids. *Computers & Fluids*, 92(0):172 – 187, 2014.

[57] A. B. Wollaber. *Advanced Monte Carlo Methods for Thermal Radiation Transport*. PhD thesis, University of Michigan, Ann Arbor, Michigan, 2008.

[58] A. B. Wollaber and J. S. Warsaw. A new, coupled transport-diffusion method for radiative transfer calculations. In *Proceedings of ANS M&C 2013*, Sun Valley, Idaho, USA, May 5-9, 2013.

[59] Allan B. Wollaber and Edward W. Larsen. A linear stability analysis for nonlinear, grey, thermal radiative transfer problems. *Journal of Computational Physics*, 230(4):1528 – 1546, 2011.

[60] R. T. Wollaeger, T. J. Urbatsch, A. B. Wollaber, and J. D. Densmore. An analysis of source tiliting and sub-cell opacity sampling for IMC. Technical Report LA-UR-12-23258, Los Alamos National Laboratory, 100, 2012.

[61] Ryan T Wollaeger and Daniel R Van Rossum. Radiation transport for explosive outflows: Opacity regrouping. *The Astrophysical Journal Supplement Series*, 214(2):28, 2014.