



LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

A Hybrid Monte Carlo-Deterministic Second Moment Method for Thermal Radiative Transfer

M. M. Pozulp

April 7, 2025

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

A Hybrid Monte Carlo-Deterministic Second Moment Method for Thermal Radiative
Transfer

by

Michael Mettler Pozulp

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Applied Science and Technology

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Jasmina Vujic, Chair
Professor David Attwood
Professor Panayiotis Papadopoulos
Dr. Terry Haut

Spring 2025

A Hybrid Monte Carlo-Deterministic Second Moment Method for Thermal Radiative
Transfer

Copyright 2025
by
Michael Mettler Pozulp

Abstract

A Hybrid Monte Carlo-Deterministic Second Moment Method for Thermal Radiative Transfer

by

Michael Mettler Pozulp

Doctor of Philosophy in Applied Science and Technology

University of California, Berkeley

Professor Jasmina Vujic, Chair

Computer simulations of hot matter often solve a coupled system of partial integro-differential equations for the matter temperature and the radiation specific intensity. While the former quantity is four dimensional through space-time dependence, the latter quantity also includes frequency and angle dependence, which makes it seven dimensional. The Monte Carlo method is often used in production to compute this high-dimensional quantity. Faster, less accurate calculations solve a diffusion approximation using a deterministic method. I present a novel hybrid method which combines the two approaches.

The aforementioned system of equations that my method solves is called the thermal radiative transfer (TRT) equations. The TRT system is nonlinear. I linearize the system using the standard implicit multigroup algorithm for x-ray photon transport called implicit Monte Carlo (IMC). One timestep of IMC solves a linear transport equation using random variates to create simulation particles called IMC photons and advances them through phase space. I replace a single IMC timestep with multiple steps which are computationally more efficient.

My replacement is an iteration. Every iteration cycle, I couple a Monte Carlo solve of the linear transport equation with a deterministic solve of a transport-corrected diffusion system in a hybrid Second Moment Method (SMM). The SMM system is linear because it uses an additive closure to close the moment system. This contrasts with another moment method called Variable Eddington Factor (VEF) which is nonlinear because it uses a multiplicative closure. The left-hand side of the SMM system is the standard radiation diffusion operator. The right-hand side includes the standard radiation diffusion sources, plus an additional source term involving a quantity called the correction tensor, which couples the SMM system with the transport equation.

Pre-existing work uses a deterministic method known as Discrete Ordinates (S_N) to compute the correction tensor. Angular discretization errors, known as ray effects, can create

unphysical starburst patterns in S_N solutions. My choice to use Monte Carlo instead of S_N avoids ray effects at the cost of introducing random variability, also known as statistical noise. This presents two problems: the noise itself and noise amplification. I manage the noise itself using standard Monte Carlo variance reduction techniques. The noise amplification is caused by differentiation of the correction tensor in the SMM system, which I avoid by solving the first-order SMM system using a mixed finite element method (FEM). Applying an integration by parts rule during the FEM derivation of the weak form of the first-order SMM system offloads the derivative from the correction tensor to the FEM test function.

I demonstrate the viability of my novel hybrid method by using it to solve linear transport problems in two spatial dimensions. I intend my demonstration of my method to serve as a foundation for its implementation in production computer simulations of hot matter.

Contents

Acronyms	iv
List of Symbols	vii
List of Publications	xiii
Acknowledgments	xv
1 Introduction	1
1.1 Physics of Thermal Radiative Transfer	2
1.1.1 Temperature	3
1.1.2 Hot Dense Plasma	6
1.2 Equations of Thermal Radiative Transfer	8
1.3 Methods for Thermal Radiative Transfer	10
1.3.1 Implicit Monte Carlo Linearization	10
1.3.2 Linear Transport	12
1.3.3 Deterministic Methods for Linear Transport	13
1.3.4 Monte Carlo Methods for Linear Transport	14
1.3.5 Moment Methods for Linear Transport	17
1.3.6 Thick Diffusion Limit	23
1.4 Hybrid Second Moment Method Summary	24
1.5 Similar Hybrid Methods	25
1.5.1 Interrupted Monte Carlo	25
1.5.2 Iterated Monte Carlo	26
2 Deterministic Component of Hybrid Second Moment	31
2.1 Olivier’s Finite Element Moment System Discretizations	31
2.2 Mixed Finite Element Discretization	32
2.2.1 Justification	32
2.2.2 Summary of the Finite Element Method	35
2.2.3 Finite Element Integration	37
2.2.4 Sobolev Spaces	38
2.2.5 Finite Element Spaces for the Mixed Problem	39

2.2.6	Derivation of the Weak Form	41
2.2.7	Convergence of the Discretization	43
2.2.8	Mixed Finite Element Method Algorithm	43
2.2.9	Linear Solvers and Preconditioners	44
2.2.10	Linear System Size	49
3	Monte Carlo Component of Hybrid Second Moment	50
3.1	Monte Carlo Simulation of Random Experiments	51
3.1.1	Discrete Random Variables	51
3.1.2	Continuous Random Variables	53
3.1.3	Probability Distribution Moment Estimation	54
3.1.4	Central Limit Theorem	58
3.2	Monte Carlo Simulation as Integral Estimation	58
3.3	Monte Carlo Integration of Linear Transport	60
3.3.1	Derivation of Characteristic Equation	61
3.3.2	Derivation of $\hat{\phi}$ Estimator	63
3.3.3	Derivation of $\hat{\phi}_s$ Estimator	70
3.3.4	Random Variate Generation	71
3.3.5	Summary of Fixed Source Integrals	76
3.3.6	Monte Carlo Transport Algorithm	78
3.4	Monte Carlo Integration of Second Moment Method	80
3.4.1	Derivation of Scattering Source Integral	80
3.4.2	Derivation of $\hat{\mathbf{T}}$ Estimator	82
3.4.3	Derivation of $\hat{\beta}$ Estimator	82
3.5	The Central Limit Theorem and Transport Estimators	83
3.6	Derivation of $\text{Var}[\hat{\mathbf{T}}]$ in the Thick Diffusion Limit	85
3.7	A Lower Variance Estimator for \mathbf{T}	91
3.7.1	Derivation of $\hat{\mathbf{R}}$ Estimator	92
3.7.2	Implementation Details of $\hat{\mathbf{R}}$ Estimator	95
3.7.3	Derivation of $\text{Var}[\hat{\mathbf{R}}]$ in the Thick Diffusion Limit	96
3.8	Deviatoric Estimator for \mathbf{T}	99
3.8.1	Derivation of Deviatoric Estimator	100
3.8.2	Implementation Details of Deviatoric Estimator	103
3.8.3	Derivation of Deviatoric Estimator Variance in the Thick Diffusion Limit	105
4	Combining the Two Components of Hybrid Second Moment	108
4.1	Numerical Properties	108
4.2	Separating the Sources	108
4.3	Hybrid Second Moment Algorithm	109
4.4	Implementation Specifics	110
4.4.1	Dimensionality	111
4.4.2	Programming Language, Compiler, Platform, and Dependencies	113

5	Non-deviatoric Numerical Results	114
5.1	Method of Manufactured Solutions Problem	115
5.1.1	Source Expressions	119
5.2	Thick Diffusion Limit Problem	123
5.3	Linearized Crooked Pipe Problem	126
5.3.1	Particle-Equivalent Comparisons with Varied Optical Thicknesses . .	131
5.3.2	Confirming Avoidance of Noise Amplification	137
5.4	Linearized Lattice Problem	140
5.4.1	Particle-Equivalent Comparisons with Varied Optical Thicknesses . .	145
5.4.2	Confirming Avoidance of Noise Amplification	150
6	Deviatoric Numerical Results	154
6.1	Method of Manufactured Solutions Problem	155
6.2	Thick Diffusion Limit Problem	156
6.3	Linearized Crooked Pipe Problem	159
6.3.1	Particle-Equivalent Comparisons with Varied Optical Thicknesses . .	160
6.3.2	Confirming Avoidance of Noise Amplification	166
6.4	Linearized Lattice Problem	169
6.4.1	Particle-Equivalent Comparisons with Varied Optical Thicknesses . .	171
6.4.2	Confirming Avoidance of Noise Amplification	176
7	Conclusion	179
7.1	Hybrid Second Moment Method	179
7.2	Deviatoric Hybrid Second Moment Method	180
7.3	Future Work	181
7.4	Coda	182
	Bibliography	184
A	Laser Fusion Experiment	194
A.1	Description of the laser fusion experiment	195
A.2	Computer model of the laser fusion experiment	199

Acronyms

ALE	Arbitrary Lagrangian-Eulerian
AMG	Algebraic Multigrid
BR2	Second Method of Bassi and Rebay
CASC	Center for Applied Scientific Computing
CBET	cross-beam energy transfer
CDF	cumulative distribution function
CG	conjugate gradient
CLT	Central Limit Theorem
CMFD	coarse mesh finite difference
CP	Crooked Pipe
CPU	central processing unit
DAIMC	Diffusion Accelerated Implicit Monte Carlo
DDMC	discrete diffusion Monte Carlo
DG	Discrete Galerkin
DHSM	deviatoric hybrid second moment
DSA	diffusion synthetic acceleration
ESE	effective scattering events
FEM	finite element method
FIESK	Fredholm integral equation of the second kind
FOM	figure of merit
GMRES	Generalized Minimal Residual Method
GPU	graphics processing unit
HOO	high-order/low-order
HPC	high performance computing

HSM hybrid second moment

iid independent and identically distributed

IMC implicit Monte Carlo

IMD implicit Monte Carlo diffusion

IP Interior Penalty

iQMC iterative Quasi-Monte Carlo

ISA instruction set architecture

JFNK Jacobian-Free Newton-Krylov

JFNK-NDA(MC) Jacobian-free Newton-Krylov nonlinear diffusion acceleration Monte Carlo

LANL Los Alamos National Laboratory

LCP Linearized Crooked Pipe

LL Linearized Lattice

LLN Law of Large Numbers

LLNL Lawrence Livermore National Laboratory

LTE local thermodynamic equilibrium

MCTP Monte Carlo Transport Project

MDLDG Minimal Dissipation Local Discontinuous Galerkin

MINRES Minimal Residual Method

MMS Method of Manufactured Solutions

MOC Method of Characteristics

NAS National Academy of Sciences

NDA nonlinear diffusion acceleration

NIF National Ignition Facility

PDE partial differential equation

PDF probability density function

PMF probability mass function
PRNG pseudo random number generator
QMC quasi-Monte Carlo
RT Raviart Thomas
RW random walk
 S_N Discrete Ordinates
SI source iteration
SM streaming multiprocessor
SMM Second Moment Method
SPD symmetric positive definite
TDL thick diffusion limit
TRRM the Random Ray Method
TRT thermal radiative transfer
UMC unaccelerated Monte Carlo
VEF Variable Eddington Factor

List of Symbols

Below is a list of the symbols used throughout this document. A symbol can have a different meaning in a different context. For example, I use the symbol ρ from the Greek alphabet in a physics context for the mass density, whereas I use $\rho(\mathbf{r}', \mathbf{\Omega}, s)$ in a Monte Carlo context for a joint probability density function. Thus, ρ appears in the Physics Symbols list and $\rho(\mathbf{r}', \mathbf{\Omega}, s)$ appears in the Monte Carlo Symbols list below. As a courtesy to the reader, I make sure to always write the joint probability density function $\rho(\mathbf{r}', \mathbf{\Omega}, s)$ with its arguments to visually distinguish it from the mass density ρ .

Another example is k , which I use for both the absorptivity and Boltzmann's constant. In this dissertation, if k multiplies the intensity I , then k is the absorptivity, whereas if k multiplies the temperature T , then k is Boltzmann's constant.

Physics Symbols

a	radiation constant (note: $a = 8\pi^5 k^4 / (15h^3 c^3)$, appears in aT^4)
B	Planck's function
b	frequency-normalized Planck spectrum (note: $b = B / \int_0^\infty B d\nu$)
b_n	frequency-normalized Planck spectrum at time t_n (note: $b_n = b(\nu, T(t_n))$)
c	speed of light
c_v	material specific heat capacity
\mathbf{E}	Eddington tensor
E_b	Eddington boundary factor
\mathbf{F}	force
f	particle phase space number density
f_n	Fleck factor
g_1	degeneracy of state 1
g_2	degeneracy of state 2
h	Planck's constant
I	radiation intensity
I_n	radiation intensity at time t_n (note: $I_n = I(\mathbf{x}, \mathbf{\Omega}, \nu, t_n)$)
I_{n+1}	radiation intensity at time t_{n+1} (note: $I_{n+1} = I(\mathbf{x}, \mathbf{\Omega}, \nu, t_{n+1})$)
I^b	radiation inflow boundary function
I^i	radiation intensity initial condition function
\mathbf{J}	first angular moment of ψ
J_{in}	inflow function contribution to moment system boundary condition
J_n^\pm	half-range, normal-dotted, first angular moments of ψ
j	emissivity

k	absorptivity or Boltzmann's constant (note: kI indicates the former and kT indicates the latter)
m	mass
m_e	electron mass
n_1	number of ions in electronic excitation state 1
n_2	number of ions in electronic excitation state 2
n_e	electron number density
n_i	number density of ions with i missing electrons
\mathbf{P}	second angular momentum of ψ
q	fixed source of radiation
Q_0	zeroth angular momentum of q
Q_1	first angular momentum of q
Q_m	material energy source
Q_r	photon source
\mathbf{T}	Second Moment Method transport correction tensor
T	material temperature
T^i	material temperature initial condition function
t	time
U_m	material energy density
$U_{m,n}$	material energy density at time t_n (note: $U_{m,n} = U_m(\mathbf{x}, t_n)$)
$U_{m,n+1}$	material energy density at time t_{n+1} (note: $U_{m,n+1} = U_m(\mathbf{x}, t_{n+1})$)
U_r	equilibrium radiation energy density (note: $U_r = aT^4$)
$U_{r,n}$	equilibrium radiation energy density at time t_n (note: $U_{r,n} = U_{r,n}(\mathbf{x}, t_n)$)
v	speed
\mathbf{v}	velocity
w	energy-weight
\mathbf{x}	position
Z_i	partition function of ionization state i
α	convex combination coefficient (note: $\alpha \in [0, 1]$, appears in f_n)
β	Second Moment Method boundary correction factor
Δt	timestep
ϵ	Thick Diffusion Limit dimensionless scaling parameter
λ	photon wavelength
ν	photon frequency
ρ	mass density
σ	Stefan-Boltzmann constant
σ_a	absorption opacity

σ_n	absorption opacity at time t_n (note: $\sigma_n = \sigma_a(\mathbf{x}, \nu, T(t_n))$)
σ_p	Planck opacity (note: $\sigma_p = \int_0^\infty \sigma_a b d\nu$)
$\sigma_{p,n}$	Planck opacity at time t_n (note: $\sigma_{p,n} = \sigma_p(\mathbf{x}, T(t_n))$)
σ_s	scattering opacity
σ_t	total opacity
$\phi(\mathbf{x})$	angle integrated radiation intensity (note: zeroth angular moment of ψ)
χ_i	ionization potential energy of ionization state i
ψ	gray, steady-state radiation intensity
$\bar{\psi}$	gray, steady-state radiation inflow boundary function
Ω	the direction of particle motion (note: $\Omega \in \mathbb{S}^2$)

Math Symbols

\mathcal{D}	computational domain (note: $\mathcal{D} \subset \mathbf{R}^{\dim}$)
\dim	spatial dimensionality (note: $1 \leq \dim \leq 3$)
\mathbf{e}_i	canonical Cartesian basis
\mathbf{I}	identity operator
\mathbf{n}	outward unit normal vector on \mathcal{D}
$O(\cdot)$	order of some quantity in an asymptotic limit
r	residual of a linear system of equations
\mathbb{R}	set of all real numbers
\mathbb{R}^+	set of positive real numbers (note: $\mathbb{R}^+ = \{x \in \mathbb{R} \mid x > 0\}$)
\mathbb{S}^2	the unit sphere
\top	the matrix transpose (note: $(A^\top)_{ij} = A_{ji}$ for any matrix A)
\mathbf{x}	spatial coordinates (note: $\mathbf{x} \in \mathcal{D}$)
\mathbb{Z}^+	the set of all positive integers 1, 2, ...
δ	Dirac delta function
δ_{ij}	Kronecker delta function (note: $\delta_{ij} = 0$ if $i \neq j$ and 1 if $i = j$)
$\partial\mathcal{D}$	the boundary of the domain \mathcal{D} (note: $\partial\mathcal{D} \subset \mathbf{R}^{\dim-1}$)
κ	condition number of linear system of equations
σ_{\max}	largest singular value of linear system of equations
σ_{\min}	smallest singular value of linear system of equations
∇	spatial gradient (note: $\nabla = \frac{\partial}{\partial x}\mathbf{e}_x + \frac{\partial}{\partial y}\mathbf{e}_y + \frac{\partial}{\partial z}\mathbf{e}_z$)
$\nabla_{\mathbf{v}}$	velocity gradient
$\nabla \cdot$	divergence operator

Finite Element Symbols

\mathbf{F}	Jacobian matrix of the mesh transformation (note: $\mathbf{F} = \partial\mathbf{T}/\partial\xi$)
\mathcal{F}	mesh element face (note: $\mathcal{F} \in \Gamma$)
$H^1(\mathcal{D})$	space of functions with square-integrable gradient
$H(\text{div}; \mathcal{D})$	space of vector-valued functions with square-integrable divergence
h	characteristic mesh element width
J	Jacobian determinant (note: $J = \mathbf{F} $)
K	mesh element (note: $K \in \mathcal{T}$)
\hat{K}	reference element
$L^2(\mathcal{D})$	the space of square-integrable functions
p	finite element basis function polynomial order
\mathcal{P}_k	univariate polynomial space
$\mathbb{Q}_p(K)$	space of mapped polynomials defined by composition of $\mathcal{Q}_p(\hat{K})$ with the inverse mesh transformation
$\mathcal{Q}_p(\hat{K})$	tensor product polynomial space of equal degree in each variable
$\mathcal{Q}_{m,n}(\hat{K})$	tensor product polynomial space of $\mathcal{P}_m(\hat{K}^{\text{dim}})$ and $\mathcal{P}_n(\hat{K}^{\text{dim}})$ for $\text{dim} = 1$
$\mathcal{Q}_{\ell,m,n}(\hat{K})$	tensor product polynomial space of $\mathcal{P}_\ell(\hat{K}^{\text{dim}})$, $\mathcal{P}_m(\hat{K}^{\text{dim}})$, and $\mathcal{P}_n(\hat{K}^{\text{dim}})$ for $\text{dim} = 1$
RT_p	degree- p Raviart-Thomas space
\mathcal{T}	the computational mesh
\mathbf{T}	mesh transformation (note: $\mathbf{T}(\xi) : \hat{K} \rightarrow K$)
\mathbf{T}^{-1}	inverse mesh transformation (note: $\mathbf{T}^{-1}(\mathbf{x}) : K \rightarrow \hat{K}$)
Y_p	degree- p Discontinuous Galerkin space
Γ	set of unique faces in the mesh
Γ_0	set of unique faces internal to the mesh
Γ_b	set of unique faces on the boundary of the mesh
∂K	boundary of mesh element K
Λ_p	interior trace of RT_p
λ	Lagrange multiplier
ξ	coordinates in reference space (note: $\xi = (\xi, \eta, \zeta)^\top$)
∇_h	broken gradient
$[\![\cdot]\!]$	jump operator
$\{\!\{ \cdot \}\!\}$	average operator

Monte Carlo Symbols

\mathcal{A}	support of a random variable
$\text{area}(\mathcal{F})$	area of element face \mathcal{F}
\hat{B}	estimator for B
$\text{Cov}[X, Y]$	covariance of the random variables X and Y
d_i	path length i traversed by a simulation particle
$E[\cdot]$	expectation or expected value
F	cumulative distribution function
f	probability density function
$\hat{\mathbf{P}}$	estimator for the second angular moment of the intensity \mathbf{P}
$p(\mathbf{r}', \boldsymbol{\Omega}, s)$	joint probability density function of \mathbf{R} , $\boldsymbol{\omega}$, and S
p_{σ_t}	probability density function for the exponential random variable S
Q	combined scattering source and fixed source of radiation
\mathbf{R}	random variable representing the initial position of a simulation photon from the volume source
$\hat{\mathbf{R}}$	estimator for the second moment method transport correction tensor \mathbf{T}
\mathbf{R}_b	random variable representing the initial position of a simulation photon from the boundary source
\mathbf{r}	spatial position parameterizing the characteristic equation
\mathbf{r}'	spatial position some distance s from \mathbf{r} (note: $\mathbf{r}' = \mathbf{r} - s\boldsymbol{\Omega}$)
S	non-constant rate, exponentially-distributed random variable
\mathbb{S}_h^2	all directions on the unit hemisphere defined by $\boldsymbol{\Omega} \cdot \mathbf{n} < 0$
s	distance from \mathbf{r} in the characteristic equation
s_0	distance from \mathbf{r} to the domain boundary in the direction $-\boldsymbol{\Omega}$
s_1	distance from \mathbf{r} to element entry along particle path
s_2	distance from \mathbf{r} to element exit along particle path
s_d	minimum distance from \mathbf{r} to the element boundary face \mathcal{F}
$\hat{\mathbf{T}}$	estimator for the second moment method transport correction tensor \mathbf{T}
$U(0, 1)$	uniformly distributed random variable between 0 and 1
$U(\mathcal{D})$	uniformly distributed random position variable in the spatial domain \mathcal{D}
$U(\mathbb{S}^2)$	uniformly distributed random direction variable on the unit sphere \mathbb{S}^2
$\text{Var}[\cdot]$	variance
$\text{vol}(K)$	volume of element K
w_i	weight of simulation particle i
w_0	sum of simulation particle weights

X	a random variable
\bar{X}	the sample mean
Y	indicator function for crossing the element face \mathcal{F}
$\hat{\beta}$	estimator for the second moment method boundary correction factor β
$\hat{\Lambda}$	an estimator of some quantity Λ
λ	exponential distribution rate parameter
μ	Gaussian distribution population mean parameter
$\rho(\mathbf{r}', \boldsymbol{\Omega}, s)$	joint probability density function of \mathbf{R}_b , $\boldsymbol{\omega}_h$, and S
σ^2	Gaussian distribution population variance parameter
τ	piecewise function for the distance traveled in element K
$\hat{\phi}$	estimator for the angle integrated intensity ϕ
$\hat{\phi}_s$	estimator for the angle integrated intensity on the domain boundary ϕ_s
ψ_{inc}	radiation inflow boundary function
$\boldsymbol{\omega}$	random variable representing the initial direction of a simulation photon from the volume source
$\boldsymbol{\omega}_h$	random variable representing the initial direction of a simulation photon from the boundary source

List of Publications

I contributed five peer-reviewed publications as first author between matriculating in my PhD program in August 2020 and completing my dissertation in May 2025. Two of the five publications include intermediate results which I achieved en route to the results that I present in my dissertation:

“M&C” is the International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering.

1. M. Pozulp, T. Haut, S. Olivier, P. Brantley, and J. Vujic, **A Hybrid Second Moment Method for Thermal Radiative Transfer**, Proceedings of M&C 2025, Denver, CO, USA. April 2025.
2. M. Pozulp, T. Haut, P. Brantley, and J. Vujic, **An Implicit Monte Carlo Acceleration Scheme**, Proceedings of M&C 2023, Niagara Falls, Canada. August 2023.

My three other first-author publications regard adjacent topics which I do not discuss in my dissertation, but may be of interest to readers of my dissertation:

“EPJ Nuclear Sci. Technol.” is the European Physics Journal - Nuclear Sciences & Technology

“SNA+MC” is the Joint International Conference on Supercomputing in Nuclear Applications + Monte Carlo

3. M. Pozulp, B. Beck, R. Bleile, P. Brantley, S. Dawson, N. Gentile, E. Gonzalez, J. Grondalski, M. Lambert, M. McKinley, M. O’Brien, R. Procassini, D. Richards, A. Robinson, S. Sepke, D. Stevens, R. Vega, and M. Yang, **Status of Mercury and Imp: Two Monte Carlo Transport Codes Developed Using Shared Infrastructure at Lawrence Livermore National Laboratory**, EPJ Nuclear Sci. Technol. **10**, 19 (2024) and presented at SNA+MC 2024, Paris, France. October 2024.
4. M. Pozulp, R. Bleile, P. Brantley, S. Dawson, M. McKinley, M. O’Brien, A. Robinson, and M. Yang, **Progress Porting LLNL Monte Carlo Transport Codes to Nvidia GPUs**, Proceedings of M&C 2023, Niagara Falls, Canada. August 2023.
5. M. Pozulp, P. Brantley, T. Palmer, and J. Vujic, **Heterogeneity, Hyperparameters, and GPUs: Towards Useful Transport Calculations Using Neural Networks**, Proceedings of M&C 2021, Raleigh, North Carolina. October 2021.

Finally, I participated in a large collaboration that resulted in world-first scientific milestones known as “target gain” and “ignition”, which I briefly discuss in Appendix A of my dissertation. Myself and over one thousand collaborators are co-authors of two publications, one associated with each milestone:

- i. H. Abu-Shawareb *et al.*, **Achievement of Target Gain Larger than Unity in an Inertial Fusion Experiment**, Physics Review Letters **132**, 065102 (2024).
- ii. H. Abu-Shawareb *et al.*, **Lawson Criterion for Ignition Exceeded in an Inertial Fusion Experiment**, Physics Review Letters **129**, 075001 (2022).

Acknowledgments

I am indebted to my faculty advisor, Professor Jasmina Vujic, for admitting me to the University of California at Berkeley, guiding me through every graduation requirement, and entrusting the supervision of my PhD research to the staff of the United States Department of Energy National Laboratories, where Professor Vujic has fostered enduring research connections over many decades of collaborative PhD student mentorship. Professor Vujic taught the first course that I took at Berkeley, through which I received my first formal introduction to the equation that would occupy the entirety of my PhD research effort. There is no other professor, scientist, or administrator who I admire more than Professor Vujic, and for her mentorship I am forever grateful.

For the last ten years, I have worked continuously under the supervision of Lawrence Livermore National Laboratory (LLNL) Monte Carlo Transport Project (MCTP) Leader Dr. Patrick Brantley. Dr. Brantley's mentorship ability and project management expertise has made the MCTP a lush garden for PhD propagation: my PhD graduation (2025) was preceded by that of my fellow MCTP members Dr. Ryan Bleile (2021) and Dr. Matt O'Brien (2013). Dr. Brantley's technical depth was responsible for liberating me from many technical problems that I encountered while conducting my research, and his patient support of my educational ambitions enabled me to establish an enduring connection with Professor Vujic.

For the last three years, I have had the immense pleasure of receiving a majority of my research mentorship from Dr. Terry Haut, an applied mathematician in the Center for Applied Scientific Computing (CASC) at LLNL. Dr. Haut conscientiously built the foundation for my PhD research by preceding the mentorship of my PhD with that of my fellow Applied Science & Technology PhD graduate Dr. Sam Olivier, who graduated the same year that I began working with Dr. Haut (2022). I can still remember the overwhelming pride and excitement that I felt immediately following the moment when I learned that Dr. Haut had agreed to mentor my PhD research. Over more than one hundred hours of individual meetings with Dr. Haut, he demonstrated to me his incredible passion for the subjects in which I conducted my PhD research. His curiosity, his patience, and his kindness established the standard by which I conduct myself in a research setting. I experienced many of the most revelatory and exciting moments of my PhD research in my meetings with Dr. Haut, and for those memorable experiences I am deeply grateful.

Seven years ago, a friend recommended that I apply to the PhD program in which he was enrolled. Three years ago, that friend graduated from our PhD program, joined Los Alamos National Laboratory (LANL), and began mentoring my dissertation research, which I built directly on top of his dissertation research. That friend is Dr. Sam Olivier, and over the last three years, Dr. Olivier contributed at least twenty-five hours of his time to answer my questions about his dissertation research, my dissertation research, and the intersection of the two. I am very grateful to Dr. Olivier for his time and for our friendship.

I am also very grateful for the wisdom conferred by Professor David Attwood, co-creator of the Applied Science & Technology (AS&T) PhD Program. Professor Attwood contributed ten hours of his time to prepare me for my qualifying exam and also served as my qualifying

exam Chair. We read journal articles and “discussed” them, by which I mean that Professor Attwood patiently explained everything that I did not understand. Professor Attwood’s enthusiasm for the physics of plasmas, radiation, and lasers is unmatched by anyone else that I met at Berkeley. It is with deep kindness and passion that Professor Attwood stewards AS&T, which is now in its fourth decade under his leadership. I feel very fortunate to have been a PhD student in Professor Attwood’s AS&T.

I am filled with gratitude for the support of hundreds of friends and colleagues at LLNL. I thoroughly enjoyed working closely with the three members of the LLNL MCTP who have been working on the MCTP with me since my first day ten years ago: Shawn Dawson, Dr. Scott McKinley, and Dr. Matt O’Brien. All three provided crucially important research suggestions, as did many other MCTP members, such as Dr. Ryan Bleile, Dr. Nick Gentile, Dr. Evan Gonzalez, Dr. Mike Lambert, Dr. Alex Robinson, and Dr. Rick Vega. My friend and office neighbor Dr. Rose McCallen knew I was ready to take my qualifying exam and start writing my dissertation before anyone else did, not even myself, and thereby helped me graduate at least one semester earlier than I would have if she had not told me so. I enjoyed the intellectual company and friendship of Jason Burmark, Dr. Robert Chen, Mike Collette, Dr. Arjun Gambhir, and especially Tom Stitt, with whom I participated in many wide-ranging technical and non-technical discussions over hundreds of hours commuting, in each other’s offices, in his home, and innumerable other places that we went together. I am thankful for the mentorship of my former Group Leaders, Punita Sinha and Adam Kunen, as well as my current Group Leader, Aaron Black. I am also thankful to have met George Zimmerman and the late Dr. John Castor. I deeply appreciate their technical advice and especially the gentleness with which they communicated it. Carol Musto provided tireless stewardship of my education as a patient mentor, trusted confidant, and loyal friend.

Finally, my research benefited significantly from at least one discussion with each of the following researchers: Professor Dmitriy Anistratov (North Carolina State University), Dr. Matt Cleveland (LANL), Professor Todd Palmer (Oregon State University), Professor Anil Prinja (University of New Mexico), and Dr. Ryan Wollaeger (LANL). Professor Larry Leemis (College of William & Mary) introduced me to the Monte Carlo method in his calculus-based probability course, which was my favorite course that I took as an undergraduate student. Tom Crockett (College of William & Mary) introduced me to high performance computing (HPC) and taught me the HPC skills which lead me to LLNL. My uncle, Dr. Thomas Brophy, whose academic training is in the physics of planetary science, suggested that I study physics, practiced with me for my qualifying exam, discussed my research with me, and made many helpful suggestions for my dissertation and my career.

For my father

Chapter 1

Introduction

I present a novel hybrid method for solving the equations of thermal radiative transfer (TRT). The TRT equations are a useful model for computer simulations of hot matter. I describe the physics underpinning the TRT equations in section 1.1 of this introduction. My goal is to elucidate the limitations of the model so that readers can decide whether the model is appropriate for their application before spending time learning my method. I show the TRT equations in section 1.2 and I give descriptions of existing TRT solution methods in section 1.3. I finish this introduction with a summary of my method in section 1.4 and summaries of some methods which are similar to mine in section 1.5.

A hybrid method for solving the TRT equations combines Monte Carlo methods with deterministic methods in an attempt to utilize the strengths and circumvent the weaknesses of the two approaches. Deterministic methods discretize every dimension of the TRT equations. Discretization converts the TRT equations, which are a system of continuous equations, to a system of discrete equations. The continuous system, when solved in a specified function space, requires an infinite number of degrees-of-freedom to represent its solution. The discrete system has a finite number of degrees-of-freedom, though the number required for an acceptable approximation of the solution to the continuous system can be very large. A discrete solution will converge to the continuous solution in the limit of infinitely many discrete equations. In the deterministic component of my hybrid method, which I describe in Chapter 2, I use a mixed finite element discretization to compute a scattering source for the Monte Carlo component.

Monte Carlo methods use a random sample of photon histories to compute an estimator which converges to the TRT solution in the limit of infinitely many photon histories. A photon history is the sequence of events undergone by a simulation photon. Events include emission, scattering, and absorption. Random numbers and physical properties of the matter determine the quantity and type of events that constitute the histories. In the Monte Carlo component of my hybrid method, which I describe in Chapter 3, I use a random sample of photon histories to compute source terms for the deterministic component.

The system of TRT equations is nonlinear, but can be linearized over a timestep to produce a system of linear equations. The solution of the linear system converges to the

solution of the non-linear system in the limit of infinitely many timesteps. The solution for a finite number of timesteps is often a useful approximation of the TRT solution.

My hybrid method is a novel approach for computing the numerical solution of the linear system. I demonstrate my hybrid method by implementing it and using my implementation to compute a numerical solution to the linear system. I describe the way that I combine the deterministic and Monte Carlo components of my method, along with some details of my implementation, in Chapter 4. Finally, I present solutions that I calculated using my implementation for several problems in Chapters 5 and 6, and I offer some concluding remarks as well as ideas for future work in Chapter 7.

1.1 Physics of Thermal Radiative Transfer

Hot matter radiates. In this context, “radiate” means to emit light, or electromagnetic radiation (they are the same)¹. This fundamental physical phenomenon underpins many physical sciences; it is perhaps most important for observational astronomy. It also influences many life sciences, like biology, because biological organisms use radiation. For example, radiation is how we see. Our eyes are sensitive to a narrow band of radiation called “visible” light. We “see” an object when our eyes detect the visible light emitted, reflected, or scattered by the object. Solar radiation allows us to perceive our surroundings, a task which increases in difficulty when the sun descends beneath the horizon. Moonlight, which is sunlight reflected off the moon, allows us to see even when the sun is obscured by the Earth.

Most radiation we cannot see. If our eyes were sensitive to long wavelength light, our skin would appear bright, even in the darkness of a moonless night. If our eyes were equally sensitive to even longer wavelengths, our clothes would appear bright too, but not as bright as our skin. This is because hot matter radiates, even if the matter is not as hot as the sun, and hotter matter radiates more intensely, which is why our skin would appear brighter than our clothes. A more detailed explanation for this skin-and-clothes example requires the definition of a physical quantity called temperature, which I describe in section 1.1.1. Emission, which is the production of radiation, accompanies temperature.

Hot matter absorbs. Absorption, which removes radiation by transferring its energy to matter, is the opposite of emission, where energy is released by radiation. In his explanation of the photoelectric effect, for which he won the 1921 Nobel Prize in Physics, A. Einstein² argued that radiation absorption causes the photoelectric effect. He said that illuminating matter with radiation exceeding a frequency threshold causes the matter to absorb the radiation. The matter then emits photoelectrons, which are the observable consequence of the photoelectric effect.

Absorption and emission are caused by electrons. A photon is a packet of radiation. The number of electrons bound to an atom strongly influences the way that a photon will interact with it, just like the number of nucleons bound in an atomic nucleus strongly influences the

¹Hereafter, I refer to electromagnetic radiation as simply “radiation”.

²Albert Einstein (1879-1955) was a German physicist.

way that a free neutron will interact with it. The quantity of bound electrons is determined by the atomic number of the element and the ionization level of the matter.

1.1.1 Temperature

In perhaps the most common scientific usage of the term, temperature refers to a single value which parameterizes the distribution of kinetic energies of the particles that constitute matter. If a particle has mass m and velocity magnitude v , then it has kinetic energy $E = mv^2/2$. Kinetic temperature comes from a kinetic theory of gases developed in the 19th century by J.C. Maxwell³ and L. Boltzmann⁴. In the first year of the 20th century, Max Planck⁵ contributed a new meaning: temperature refers to a single value which parameterizes the spectral distribution of radiation specific intensity. A spectral distribution is a distribution that depends on radiation frequency (or wavelength, because photon wavelength λ can be written in terms of photon frequency ν as $\lambda = c/\nu$ where c is the speed of light in vacuum). I explain radiation specific intensity in section 1.2. A third definition, also contributed by Boltzmann, is that temperature refers to a single value which parameterizes the distribution of electron excitation states. Excitation occurs when an electron is promoted to a higher energy level within the atom to which it is bound.

The three definitions of temperature, and the three distributions which they parameterize, correspond to three energy reservoirs: radiation, kinetic motion of the matter particles, and electronic excitation of the matter particles. A flurry of scientific discoveries over the last two centuries revealed myriad mechanisms by which energy transfers between the three reservoirs. Fig. 1.1 depicts the three reservoirs as vertices of a triangle, its edges labeled with examples of processes which transfer energy between the reservoirs. Bremsstrahlung is radiation produced by the deceleration or deflection of a charged particle, such as an electron. Compton scattering transfers energy from a photon to a charged particle in a collision that causes the photon to lose energy and change direction.

The distribution corresponding to the center vertex in Fig. 1.1 is the Planck distribution. It describes the relative occurrence of emission photons of different frequencies. Radiation is an energy reservoir because photons have energy. The energy of a photon is the product of the photon frequency and Planck's constant, $E = h\nu$. The Planck distribution is,

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

See Physics Symbols, Math Symbols, Finite Element Symbols, and Monte Carlo Symbols in the front matter for definitions of the symbols in Eq. (1.1) and all subsequent equations in this dissertation. Eq. (1.1) is consistent with the notion that hot matter radiates, and that matter need not be as hot as the sun to have Planckian emission: B exists for any $T > 0$. Matter

³James Clerk Maxwell (1831-1879) was a Scottish physicist.

⁴Ludwig Boltzmann (1844-1906) was an Austrian physicist.

⁵Max Planck (1858-1947) was a German physicist.

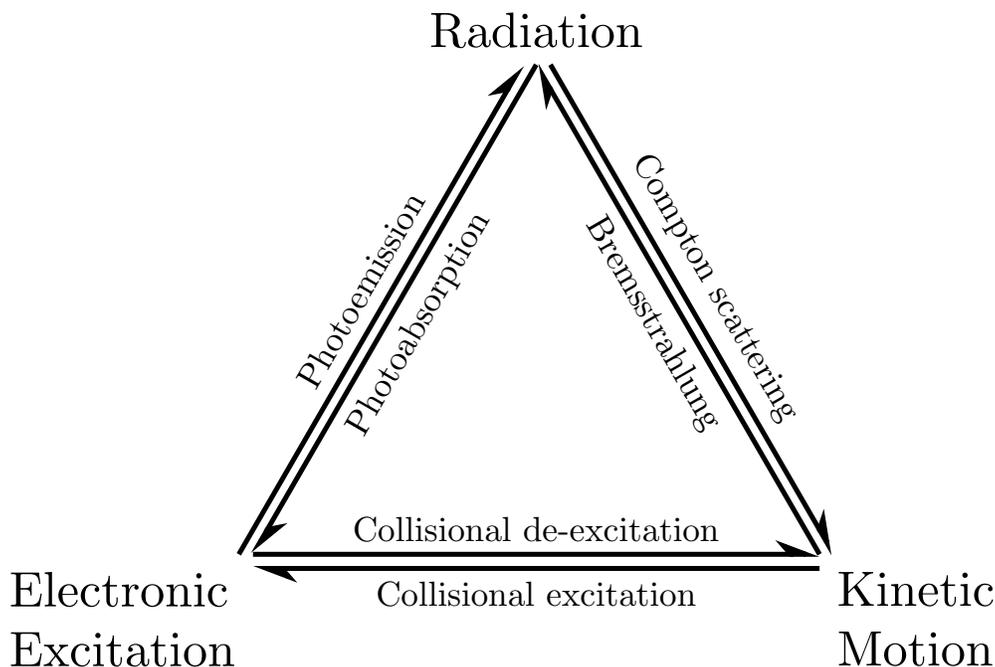


Figure 1.1: Energy reservoir triangle [1].

emits radiation because of emission processes such as those listed on the left and right edges of the triangle in Fig. 1.1. These are microphysical processes involving one or two matter particles. The Planck distribution is a macroscopic representation. Like other results in statistical mechanics, it reduces a system with an enormous number of degrees of freedom (position, momentum, species, and other attributes of every matter particle) to a single degree of freedom (temperature). Thus, Planckian emission is an idealization. No object has emission that is perfectly Planckian. Some objects come very close. Solar emission produces a frequency spectrum that closely matches a Planck distribution at $T = 5788$ Kelvin.

Brightness refers to the radiation specific intensity of an object, meaning one object is brighter than another at a specified frequency if it emits, reflects, or scatters light with greater intensity at that frequency. A plot of the Planck distribution at two different temperatures can illustrate the relative brightness of two objects at different temperatures if one can assume that the objects have emissivities which are approximately Planckian. Earlier, I argued that skin is brighter than clothes, by which I meant that $B_{T_{\text{clothes}}} < B_{T_{\text{skin}}}$. Fig. 1.2 shows a plot of the Planck distribution for room-temperature clothing (295 Kelvin) and skin temperature (310 Kelvin). Thus, to someone with eyes sensitive to low frequency radiation who is observing in the absence of other light sources, skin appears brighter because skin has brighter emission because skin is hotter. If one measured the emission spectra from skin and clothes, the spectra would deviate from the Planckians plotted in Fig. 1.2, but the Planckians may be qualitatively correct. Measured spectra contain emission lines due to

the quantization of electron energy levels. The Planck distribution is continuous and does not include discrete features like emission lines, which would appear as vertical lines in an emission spectrum⁶.

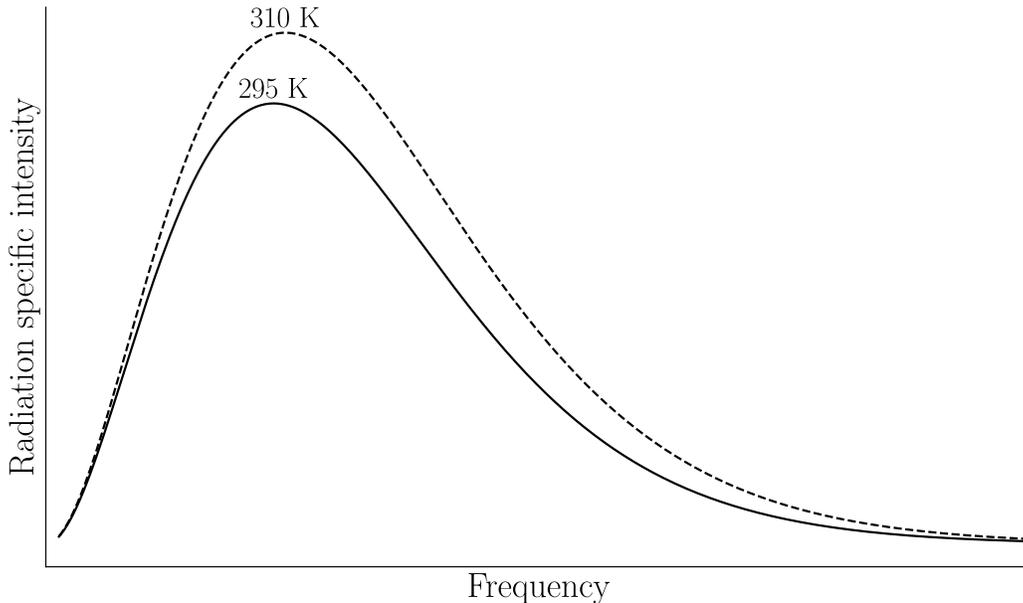


Figure 1.2: Planckians at plausible temperatures for clothing and skin.

The distribution corresponding to the right vertex in Fig. 1.1 is the Maxwell distribution of matter particle velocities,

$$f(v) = 4\pi v^2 \left(\frac{m}{2\pi kT} \right)^{\frac{3}{2}} e^{-\frac{1}{2}mv^2/kT} . \quad (1.2)$$

The left vertex distribution is the Boltzmann distribution for the relative populations of electronic excitation states,

$$\frac{n_2}{n_1} = \frac{g_2}{g_1} e^{-h\nu/kT} . \quad (1.3)$$

The three distributions Eqs. (1.1) to (1.3) have an important common quality besides temperature parameterization: each distribution describes the way that energy is distributed among the constituent particles of the three energy reservoirs. Finally, Eq. (1.3) is insufficient for ionized matter, but it can be augmented by incorporating an equation contributed by M. Saha⁷ called the Saha ionization equation,

$$\frac{n_{i+1}n_e}{n_i} = 2 \frac{Z_{i+1}}{Z_i} \left(\frac{2\pi m_e kT}{h^2} \right)^{3/2} e^{\frac{-x_i}{kT}} . \quad (1.4)$$

The combination of Eq. (1.3) and Eq. (1.4) is called the Saha-Boltzmann distribution.

⁶An example plot showing line emission in an emission spectrum can be seen in Figure 8.1 of [2].

⁷Meghnad Saha (1893-1956) was an Indian physicist.

1.1.2 Hot Dense Plasma

While the constituent particles of the radiation energy reservoir are photons, the constituent particles of the other two reservoirs are ions and free electrons, because the matter that TRT describes is hot enough that molecular bonds have been broken and atoms have been stripped of one or more of their bound electrons. The matter is a hot plasma. Additionally, the plasma is dense, meaning that collisions between the matter particles are the dominant process for populating the excitation states with excited ions. Collisional processes dominate radiative processes, meaning that the center leg of the energy reservoir triangle in Fig. 1.1 dominates the left leg. This restricts the region of applicability to the upper-right portion of the plasma temperature-density plane⁸. Most matter in the observable universe is plasma, but not all plasmas are hot and dense.

The temperature of a plasma is often described using more than a single value. A common approach is to use a two-temperature model, which has separate temperatures for the ions, T_i , and the free electrons, T_e . A two-temperature model can capture plasma dynamics occurring on timescales that are shorter than the timescale required for collisional equilibration of the ions and the free electrons. It can also reproduce the physical phenomenon in which the free electrons gain or lose energy much faster than the ions, because the mass of an electron is much smaller than the mass of an ion. In a fully-ionized hydrogen plasma, where each ion is just a single proton, an electron is about 1,800 times lighter than an ion. Plasmas containing elements which are heavier than hydrogen have even higher ion-electron mass ratios. Finally, external energy sources such as lasers preferentially heat electrons, thereby causing T_e to rise faster than T_i , which is an important physical phenomenon that a two-temperature plasma model can reproduce, but a single temperature plasma model cannot.

Discussions of multi-temperature plasma models sometimes include references to a radiation temperature, T_r . Unlike ions and electrons, photons are not constituent plasma particles. However, the ion and electron temperatures are often used to communicate the amount of energy in the ion and electron fluids, respectively, and a radiation temperature would be useful for communicating the amount of energy in the photon fluid. Such a quantity may be computed by simply solving for T_r in the Stefan-Boltzmann law,

$$u = aT_r^4, \tag{1.5}$$

where u is the energy density of the radiation field, and the radiation constant a is four times the ratio of the Stefan-Boltzmann constant and the speed of light, $a = 4\sigma/c$.

To simplify the presentation of the TRT equations, I assume that the plasma temperature can be described by a single value, and I also refer to this plasma temperature as the “matter temperature” and the “material temperature” in an interchangeable manner. It is this single temperature that we use to evaluate Eqs. (1.1) to (1.4). Thus, the matter emission in TRT is a Planckian at the matter temperature. Furthermore, the partial integro-differential equation coefficients known as opacities, which appear in the TRT equations, can be expressed in terms

⁸An example plane can be seen on the first page of Chapter 8 “Physics of Hot Dense Plasmas” in [2].

of the matter temperature. The opacities also depend on the plasma electron number density (or mass density) and photon frequency. The temperature dependence of the opacities means that there is no need to solve for the populations of the ion excitation states because the validity of a matter temperature permits the populations to be determined by evaluating the Saha-Boltzmann distribution at the matter temperature. Opacities are also sometimes called reaction cross sections and mass attenuation coefficients. The opacities in my description of the TRT equations have dimensions of inverse length.

Astrophysical and laboratory plasmas which are not collisionally dominated, either because they are insufficiently dense or because the radiation is very strong, require a more sophisticated treatment. One approach is to augment the TRT system with additional equations describing the populations of the ion excitation states⁹. One uses the populations to compute the (non-Planckian) emissivity and opacity. Such treatments can be expensive. They exceed the scope of this dissertation.

Cold plasmas, meaning plasmas with temperatures that are so low that the plasma emission is too weak to change the plasma dynamics, may not require TRT. A linear radiation transport equation could be sufficient. However, if the “external” radiation source is strong enough, and the illumination timescale is long enough, the cold plasma could become hot. The likelihood of heating would also depend on the optical thickness of the cold plasma at the frequency of illumination. If there are no external radiation sources, then even linear radiation transport may be unnecessary, since there is no radiation (except for the negligibly weak emission from the cold plasma). Matter which is very cold may not even be a plasma, but rather a gas, liquid, or solid. These cold matter types do not require TRT due to weak emission resulting from the low matter temperature required to sustain neutrality.

Hot dense plasmas are bright sources of extreme ultraviolet and soft x-ray radiation, which is an ideal spectral band for TRT modeling. Extrapolating away from this band in either direction can introduce physics which TRT does not incorporate. At longer wavelengths, the TRT equations may be inadequate because they do not incorporate electromagnetic wave behaviors, like diffraction, which become more evident at longer wavelengths. Some wave effects can be reincorporated; one example is polarization¹⁰. At shorter wavelengths, the TRT equations may be inadequate because they do not consider photonuclear effects.

Finally, because TRT has a matter temperature, it can only describe plasmas containing many particles. A system of very few particles, for which the statistical properties underpinning the kinetic theory of gases are violated, would preclude the use of temperature (it makes no sense to describe the velocities of two ions using a Maxwellian). TRT models a continuous mass of matter, not discrete matter particles, and so a microscopic description which considers the velocity of each particle would be more useful for a plasma comprised of few particles. Examples include the particle-in-cell method and molecular dynamics.

⁹See Chapter 9 “Spectral line transport” in [3].

¹⁰See Chapter 4 “The Representation of Polarized Light in the Equation of Transfer” in [4].

1.2 Equations of Thermal Radiative Transfer

Radiative transfer refers to an equation for computing the radiation specific intensity, I , which is a scalar. It is high dimensional relative to other commonly encountered quantities in computational physics and engineering. The independent variables of continuum mechanics disciplines like solid mechanics and fluid mechanics are often four dimensional through space-time dependence. Examples include position, velocity, energy, temperature, and pressure. Radiation specific intensity is higher dimensional because it includes frequency and angle dependence, which makes it seven dimensional. Frequency contributes a single dimension while angle contributes two. This is because every angle in space can be expressed as a unit vector, $\boldsymbol{\Omega}$, directed at locations on the unit sphere, and two coordinates are required to specify every location on the unit sphere. Thus, we may write $I = I(\mathbf{x}, \boldsymbol{\Omega}, \nu, t)$, which identifies the seven dimensions of the radiation specific intensity phase space.

A useful interpretation of the meaning of I is through its relationship to the photon phase space number density. Let $f(\mathbf{x}, \boldsymbol{\Omega}, \nu, t)$ denote the number of photons in the phase space element $d\mathbf{x} d\boldsymbol{\Omega} d\nu$ at time t . Then $I = ch\nu f$. This relation reveals that I is a type of power, or energy per time. The dimensions of f can be expressed as,

$$[f] = \frac{\#}{\text{cm}^3 \text{ sr Hz}}, \quad (1.6)$$

and the dimensions of I can be expressed as,

$$[I] = \frac{\text{energy}}{\text{cm}^2 \text{ sr Hz s}}. \quad (1.7)$$

The energy in the numerator of Eq. (1.7) is due to the energy of the photons ($E = h\nu$) which constitute the radiation. We can integrate to reduce I to just energy per time, or power. Before integrating we multiply by the absorption opacity σ_a which has dimensions of inverse length. The result is the power deposited in the matter by its absorption of radiation,

$$\int_{\mathcal{D}} \int_{\mathbb{S}^2} \int_0^\infty \sigma_a I d\nu d\boldsymbol{\Omega} d\mathbf{x}. \quad (1.8)$$

The radiative transfer equation says that the rate of change of the radiation specific intensity is equal to the sum of the gains and losses,

$$\left(\frac{1}{c}\right) \frac{\partial I}{\partial t} + \boldsymbol{\Omega} \cdot \nabla I = j - kI. \quad (1.9)$$

The first term is the rate of change, the second is loss due to movement of photons out of the phase space volume element, the third is gain from emission due to an emissivity j , and the fourth is loss from absorption due to an absorptivity k . If a plasma is insufficiently dense, or the radiation is sufficiently strong, such that radiative processes rather than collisional processes are the dominant process for populating the excitation states with excited ions,

then one must solve Eq. (1.9) to determine I . Otherwise, *thermal* radiative transfer can be substituted for radiative transfer, which greatly simplifies the specification of j and k .

The prefix “thermal” in TRT is a constraining qualification. Section 1.1.2 articulates the physical context of this constraint. Mathematically, it means that one can make the substitutions $j = \sigma_a B$ and $k = \sigma_a$, giving

$$\left(\frac{1}{c}\right) \frac{\partial I}{\partial t} + \boldsymbol{\Omega} \cdot \nabla I = \sigma_a (B - I). \quad (1.10a)$$

Eq. (1.10a) is the first TRT equation. Its expression here disregards photon scattering processes, which contribute gain and loss terms for photons scattering into and out of phase space, respectively. Photon scattering can be incorporated by including extra terms [4]. Eq. (1.10a) contains the Planck distribution B , but says nothing about the temperature at which to evaluate it. With what temperature does one evaluate Eq. (1.1) for B ?

Recall the energy reservoir triangle in Fig. 1.1 where we had one radiation energy reservoir (center vertex) and two matter energy reservoirs (left and right vertices). Imagine reducing the length of the triangle’s center edge to zero, thus transforming two vertices into one and merging the matter energy reservoirs. The energy in the matter is now due to both electronic excitation and kinetic motion (and possibly other energy reservoirs). The matter energy reservoir encapsulating all matter energy sources is sometimes referred to as the matter “internal” energy.

Modeling this dyad of energy reservoirs, one reservoir for the radiation and one for the matter, requires an equation for the rate of change of the matter temperature. The matter temperature equation says that the rate of change of the matter temperature is equal to the sum of the gains and losses,

$$\frac{\partial U_m}{\partial t} = \rho c_v \frac{\partial T}{\partial t} = \int_{S^2} \int_0^\infty \sigma_a(\mathbf{x}, \nu', T) \left(I(\mathbf{x}, \boldsymbol{\Omega}', \nu', t) - B(\nu', T) \right) d\nu' d\boldsymbol{\Omega}'. \quad (1.10b)$$

Eq. (1.10b) is the second TRT equation. The two unknowns in Eqs. (1.10a) and (1.10b) are the seven-dimensional radiation intensity $I = I(\mathbf{x}, \boldsymbol{\Omega}, \nu, t)$ and the four-dimensional matter energy density $U_m = U_m(\mathbf{x}, t)$ or matter temperature $T = T(\mathbf{x}, t)$. The TRT system is the system of two equations Eqs. (1.10a) and (1.10b) along with the initial conditions,

$$I(\mathbf{x}, \boldsymbol{\Omega}, \nu, 0) = I^i(\mathbf{x}, \boldsymbol{\Omega}, \nu), \quad (1.10c)$$

$$T(\mathbf{x}, 0) = T^i(\mathbf{x}), \quad (1.10d)$$

and the boundary condition,

$$I(\mathbf{x}, \boldsymbol{\Omega}, \nu, t) = I^b(\mathbf{x}, \boldsymbol{\Omega}, \nu, t) \quad \text{for } x \in \partial\mathcal{D} \text{ and } \boldsymbol{\Omega} \cdot \mathbf{n} < 0. \quad (1.10e)$$

Eq. (1.10e) is sometimes called an “inflow” boundary condition because it specifies the value of the radiation intensity on the domain boundary in the subset of directions which point into the domain. The radiation intensity on the domain boundary in the outward facing directions, designated by $\boldsymbol{\Omega} \cdot \mathbf{n} > 0$, is not specified by the boundary condition because it is governed by the TRT system and therefore must be computed. The TRT system is nonlinear due to the Planck emission term B .

1.3 Methods for Thermal Radiative Transfer

Methods for solving the TRT equations typically linearize the system about a timestep. Linearization produces a system of linear equations which have a solution that is a useful approximation of the TRT solution. Linearization, and then computing the solution of the linear system, is an important component of many iterative methods for solving systems of nonlinear equations, such as Newton-Raphson and fixed-point iteration. My novel hybrid method uses the implicit Monte Carlo linearization of Fleck and Cummings [5]. I describe implicit Monte Carlo (IMC) in section 1.3.1 and the properties of the resulting linear system in section 1.3.2. I also describe deterministic (section 1.3.3), Monte Carlo (section 1.3.4), and moment methods (section 1.3.5) for solving the linear system.

1.3.1 Implicit Monte Carlo Linearization

I designed my novel hybrid method for solving the linear transport equation that arises from the implicit Monte Carlo linearization of Fleck and Cummings [5]. IMC makes approximations which allow one to replace the problem of solving the nonlinear TRT equations with the simpler problem of solving a linear transport equation. Thus, the IMC linearization is the connection between my method, which solves linear transport, and my intended application, which is TRT.

Below, I write the IMC equations using the notation from Wollaber [6]. I then introduce my model problem, which is a linear transport equation, and show the correspondence between terms in the IMC equations and my model problem. The IMC equations are a linearization of the TRT Eqs. (1.10a) and (1.10b) about a single timestep¹¹:

$$\frac{I_{n+1} - I_n}{c\Delta t} + \mathbf{\Omega} \cdot \nabla I_{n+1} + \sigma_n I_{n+1} = \frac{\sigma_n b_n}{\sigma_{p,n}} \frac{1}{4\pi} \int_0^\infty \int_{\mathbb{S}^2} (1 - f_n) \sigma_n I_{n+1} d\Omega' d\nu' + f_n \sigma_{p,n} \frac{\sigma_n b_n}{\sigma_{p,n}} \frac{cU_{r,n}}{4\pi} + (1 - f_n) \frac{\sigma_n b_n}{\sigma_{p,n}} \frac{Q_m}{4\pi} + \frac{Q_r}{4\pi}, \quad (1.11a)$$

$$\frac{U_{m,n+1} - U_{m,n}}{\Delta t} + f_n \sigma_{p,n} cU_{r,n} = \int_0^\infty \int_{\mathbb{S}^2} f_n \sigma_n I_{n+1} d\Omega' d\nu' + f_n Q_m. \quad (1.11b)$$

See Physics Symbols, Math Symbols, Finite Element Symbols, and Monte Carlo Symbols in the front matter for definitions of the symbols in Eqs. (1.11a) and (1.11b) and all subsequent equations in this dissertation. The n subscript in Eqs. (1.11a) and (1.11b) denotes a quantity evaluated at time t_n . The time discretization in Eqs. (1.11a) and (1.11b) replaced the time derivatives $\partial I/\partial t$ and $\partial U_m/\partial t$ in Eqs. (1.10a) and (1.10b) with $(I_{n+1} - I_n)/\Delta t$ and $(U_{m,n+1} - U_{m,n})/\Delta t$, respectively. The quantity Δt is the timestep. The IMC Eqs. (1.11a) and (1.11b) converge to the TRT Eqs. (1.10a) and (1.10b) in the limit of an infinitely small timestep.

¹¹For a good derivation of the IMC equations, read [6] starting at the first equation and ending with equations 24a and 24b. The original IMC presentation is in Fleck and Cummings [5].

Eqs. (1.11a) and (1.11b) are subject to the same initial conditions Eqs. (1.10c) and (1.10d) and boundary condition Eq. (1.10e) as the TRT system of Eqs. (1.10a) and (1.10b). The unknowns in Eqs. (1.10a) and (1.10b) are the intensity I and the material energy density U_m , whereas the unknowns in Eqs. (1.11a) and (1.11b) are these quantities at time t_{n+1} : I_{n+1} and $U_{m,n+1}$. One important difference between the IMC linearization and the TRT system is that $U_{m,n+1}$ can be computed explicitly after solving Eq. (1.11a) for I_{n+1} . Note that “IMC” could be considered a misnomer, because Eqs. (1.11a) and (1.11b) are only semi-implicit. Solving the IMC Eqs. (1.11a) and (1.11b) is equivalent to taking a single step of Newton’s method applied to the nonlinear TRT Eqs. (1.10a) and (1.10b). A fully-implicit method would converge the Newton iteration at every timestep, which would be very expensive, but such a method would be unconditionally stable. The semi-implicit formulation provides just enough stability to be useful, meaning that the timestep size threshold at which IMC becomes unstable is not impractically small. An explicit method would be either unstable or conditionally stable, in which case the restriction on the timestep size required for stability would be severe. Unstable methods can give inaccurate or even completely meaningless results because of exponential or uncontrollable growth of small errors¹².

One notable feature of the IMC equations is the presence of “effective scattering”, which differs from physical scattering. In physical scattering, a photon interacts with an electron through a collision without being absorbed. As a result, the photon emerges from the collision, typically traveling in a different direction and possibly with a different energy than it had prior to the interaction. Even when physical scattering is disregarded, as I have done in this dissertation (because I did not include physical scattering terms in the TRT Eqs. (1.10a) and (1.10b)), the IMC linearization always results in an effective scattering term. The effective scattering term is the first term on the right-hand side of Eq. (1.11a). It is a source term which accounts for the effective scattering of IMC photons into the phase space. If an IMC photon undergoes an effective scattering event, it means that the physical photon it represents was absorbed and re-emitted within the timestep.

The form of the “Fleck factor”,

$$f_n = \frac{1}{1 + \alpha\sigma_{p,n}c\Delta t}, \quad (1.12)$$

shows that optically-thick media and large timesteps cause $f_n \in (0, 1)$ to decrease, increasing the amount of effective scattering $(1 - f_n)$ relative to effective absorption f_n . Effective scattering can be expensive, especially in the thick diffusion limit, which is a regime that I describe in section 1.3.2. My hybrid method is a novel way to mitigate the expense of calculations with a high amount of effective scattering.

Both deterministic and Monte Carlo methods can be used to solve the IMC system of Eqs. (1.11a) and (1.11b). Deterministic methods solve the linear transport Eq. (1.11a) deterministically for I , then evaluate it in a quadrature rule to compute the integral in the

¹²Numerical methods have small errors due to discretization and finite precision arithmetic, which may require rounding the results of arithmetic operations required by the method.

temperature update Eq. (1.11b). By contrast, Monte Carlo methods use random numbers to compute the integral in Eq. (1.11b) by accumulating an estimator for the angle integrated intensity. I define estimators, their accumulation, and other Monte Carlo method concepts in Chapter 3.

The fundamental quantity in deterministic methods is the intensity I , whereas the fundamental quantity in Monte Carlo methods is the energy-weight w , which is the amount of energy represented by an IMC photon. An IMC photon with energy-weight w and frequency ν represents $w/h\nu \in \mathbb{R}^+$ physical photons if $w \in \mathbb{R}^+$. The number of physical photons represented by an IMC photon is not necessarily an integer. It is usually a positive number, though some algorithms allow for $w < 0$.

1.3.2 Linear Transport

Eq. (1.11a) is a linear transport equation. If one integrates both sides of Eq. (1.11a) over all frequencies, then a single timestep of the resulting equation can be expressed more simply as,

$$\boldsymbol{\Omega} \cdot \nabla \psi + \sigma_t \psi = \frac{\sigma_s}{4\pi} \int_{\mathbb{S}^2} \psi \, d\Omega' + q, \quad (1.13a)$$

subject to the boundary condition,

$$\psi(\mathbf{x}, \boldsymbol{\Omega}) = \bar{\psi}(\mathbf{x}, \boldsymbol{\Omega}), \quad \mathbf{x} \in \partial\mathcal{D} \text{ and } \boldsymbol{\Omega} \cdot \mathbf{n} < 0. \quad (1.13b)$$

This is the model problem from [7]. It is the problem that I solve with my hybrid method. Typically, time and frequency are discretized into timesteps and frequency groups, so TRT using the IMC linearization requires solving a linear transport equation like Eq. (1.13a) for all timesteps and frequency groups. The number of timesteps required to reach the physical time of interest depends on the timescale of the physics under study and the accuracy requirements of the application. Thousands of timesteps is not atypical. An ambitious TRT calculation may run for one hundred thousand timesteps, but one million or more timesteps would be unusual for TRT. The number of frequency groups required for an accurate model is problem-dependent and can range from one to tens of thousands. Even though Eq. (1.13a) is a five-dimensional problem, whereas Eq. (1.11a) is seven-dimensional, it still retains sufficient complexity to demonstrate solution methods which could be useful for the full-dimensional problem.

The left-hand sides of Eq. (1.11a) and Eq. (1.13a) are identical but for the time derivative, which is handled by the time-stepping procedure prescribed by IMC. The four terms on the right-hand side of Eq. (1.11a) are effective scattering, effective emission, material emission, and external sources. The first term on the right-hand side of Eq. (1.13a) serves as a proxy for effective scattering. The second term on the right-hand side of Eq. (1.13a) serves as a

proxy for the other three. That is,

$$\begin{aligned} \frac{\sigma_s}{4\pi} \int_{\mathbb{S}^2} \psi \, d\Omega' & \text{ imitates } \frac{\sigma_n b_n}{\sigma_{p,n}} \frac{1}{4\pi} \int_0^\infty \int_{\mathbb{S}^2} (1 - f_n) \sigma_n I_{n+1} \, d\Omega' \, d\nu', \\ q & \text{ imitates } f_n \sigma_{p,n} \frac{\sigma_n b_n}{\sigma_{p,n}} \frac{cU_{r,n}}{4\pi} + (1 - f_n) \frac{\sigma_n b_n}{\sigma_{p,n}} \frac{Q_m}{4\pi} + \frac{Q_r}{4\pi}. \end{aligned}$$

The linear transport Eq. (1.13a) can exhibit properties of either parabolic or hyperbolic equations, depending on the physical regime, which poses challenges for designing effective numerical schemes. In transport-dominated regimes, such as problems where the phase space is almost entirely optically-thin, the behavior of the linear transport Eq. (1.13a) is more hyperbolic, like an advection equation. In optically-thick problems, the behavior is more parabolic, like a diffusion equation.

1.3.3 Deterministic Methods for Linear Transport

Deterministic methods for solving linear transport discretize seven dimensional phase space, compute the solution at the finite set of points in phase space, then interpolate to reconstruct a continuous solution. The set of points can be large because it contains the product of points along each dimension, so a discretization which uses ten points in each dimension will produce a solution vector containing 10^7 entries. This means that the coefficient matrix representing the discretized equations has dimensions $10^7 \times 10^7$, or 10^{14} entries. If one uses the conventional 8-bytes of computer memory to store each entry, then the storage requirement approaches one petabyte, or 10^{15} bytes. This is three orders of magnitude more than the memory capacity of the highest memory capacity computers commonly available today.

Discrete Ordinates (S_N) is a popular class of deterministic methods that uses angular quadrature to discretize angle. The choice of space, energy, and time discretizations can vary among S_N implementations. Common choices are finite volume, multigroup, and backward Euler, respectively. Production S_N methods typically overcome the limitation of insufficient computer memory capacity by employing source iteration (SI) [8]. SI lags the scattering term. This allows the solution along each direction to be computed independently. Furthermore, the global coefficient matrix associated with the spatial discretization for a single direction is block triangular, or nearly so, and is never stored¹³. Instead, the system is solved by picking an angle and solving a small¹⁴ linear system in a boundary element, choosing an adjacent element in the direction of the chosen angle and solving the element-local system, and so on in a process known as a “transport sweep.” This is sometimes called a “wavefront solver” because the solution is determined progressively along a wavefront emanating from the starting element. When the sweep completes, one computes a new estimate of the scattering source, and the iteration cycle is complete. The iteration terminates when a convergence criterion is satisfied.

¹³For example coefficient matrix sparsity plots, see Figure 5.4 (a) and (b) in [7].

¹⁴A linear discontinuous spatial discretization would have 8×8 element-local systems in three spatial dimensions because $2^{\text{dim}} = 2^3 = 8$, quadratic would have 27×27 because $3^{\text{dim}} = 3^3 = 27$, and so on.

SI solves the problem of finite computer memory capacity by obviating the need to store the coefficient matrix. However, it suffers from arbitrarily slow convergence in the thick diffusion limit (TDL)¹⁵. Consider the linear transport equation,

$$L\psi = S\psi + q, \quad (1.14)$$

where $L = \mathbf{\Omega} \cdot \nabla + \sigma_t$ and $S = \frac{\sigma_s}{4\pi} \int \psi \, d\Omega'$. The SI method is

$$L\psi^{(i+1)} = S\psi^{(i)} + q, \quad (1.15)$$

where i is the iteration index. The spectral radius of $L^{-1}S$ is bounded by unity, but gets arbitrarily close to unity in the TDL, meaning that the number of iterations required to converge grows without bound. Perhaps the earliest method used in production calculations in the United States to accelerate the convergence of SI, which remains in use today, is called diffusion synthetic acceleration (DSA) [10, 11]. A method called quasi-diffusion, or Variable Eddington Factor, which converges quickly even in the TDL, was also invented around the same time as DSA, but received less attention perhaps because it originated in the Soviet Union [12]. SI acceleration methods, like DSA, and other methods that converge quickly, like quasi-diffusion, use ideas from the radiation diffusion approximation, which I describe in section 1.3.5. For a comparison of SI acceleration methods, see [13].

1.3.4 Monte Carlo Methods for Linear Transport

Monte Carlo methods for solving linear transport use a random sample of photon histories to compute an estimator for the angle integrated intensity. I define estimators in Chapter 3. A photon history is the sequence of events undergone by a simulation photon. Events include emission, scattering, and absorption. Random numbers and physical properties of the matter determine the quantity and type of events that constitute the histories.

Monte Carlo algorithms sample points in phase space which are used to initialize the parameters of simulation particles in a process called sourcing. After sourcing, the photons are tracked through the matter until they are absorbed, escape the problem geometry through a vacuum boundary, or reach the end of the timestep. Monte Carlo photons used to compute the solution of the linear transport equation arising from the IMC linearization are sometimes called “IMC photons”. If an IMC photon undergoes an effective scattering event, the corresponding physical photon was absorbed and reemitted within the timestep. However, it is important to note that the extent to which an IMC photon—a numerical artifact—can be considered representative of a physical photon is a topic of debate in the IMC method literature.

The arbitrarily slow convergence of source iteration, described in section 1.3.3, has a Monte Carlo equivalent. In the TDL, the probability that an effective scattering event is the

¹⁵The TDL [9] is a numerically challenging physical regime characterized by high optical thickness and high amounts of particle scattering. In Section 1.3.6, I present the asymptotic scaling of the transport problem data which characterizes the TDL, along with a discussion of its consequences.

next event that an IMC photon undergoes becomes arbitrarily close to unity. This causes IMC photon histories to grow without bound. A typical history might have a simulation particle move a tiny distance, undergo an effective scattering event which changes its direction, move a tiny distance, undergo another effective scattering event, and so on, hundreds or thousands or millions of times. Each event in the IMC photon history contributes very little to the Monte Carlo estimator, just like each iteration of SI contributes very little to the S_N solution in the TDL.

Also, just like incorporating ideas from the radiation diffusion approximation can accelerate the convergence of SI by reducing the number of iterations required to converge, incorporating diffusion ideas can accelerate MC calculations by reducing the length of the IMC photon histories. Perhaps the first technique invented, called random walk (RW), allows IMC photons to be moved to the edge of a sphere inscribed in the spatial mesh, provided that certain conditions are satisfied [14]. More recently invented acceleration techniques include implicit Monte Carlo diffusion (IMD) and discrete diffusion Monte Carlo (DDMC). Both replace transport zones with diffusion zones in optically-thick materials and solve a linear system arising from a discretization of the diffusion equation using a Monte Carlo interpretation [15–19].

My novel hybrid method can be viewed as an alternative MC acceleration technique. For my method to be useful, it needs to outperform the aforementioned existing techniques, which have some weaknesses. The use of diffusion descriptions in RW, IMD, and DDMC introduces modeling error because the diffusion equation is an approximation of the transport equation, though the error of the approximation goes to zero in the infinitely optically-thick limit. Regimes with intermediate optical thickness highlight the tradeoff inherent to all three methods: either sacrificing accuracy by using diffusion or sacrificing speed by using transport.

Additionally, the acceleration of RW is diminished by spatial mesh refinement because the maximum distance that an IMC photon can travel in a single RW step is constrained by the size of the maximum sphere that may be circumscribed in the zone. The IMD and DDMC methods have yet to be extended to unstructured meshes; however, researchers at laboratories such as Los Alamos National Laboratory (LANL) are actively working on this. For now, IMD- and DDMC-accelerated IMC calculations cannot use meshes which have been distorted by mesh motion, which is an inevitability when one solves the Lagrangian hydrodynamics equations. The Eulerian hydrodynamics equations do not have mesh motion, but the Eulerian representation introduces error due to mass advection caused by fluid flow across the stationary mesh. If one insists on coupling IMC photonics with a Lagrangian hydrodynamics representation, and also wants IMD or DDMC acceleration, then one must perform an expensive mapping operation, which transfers all of the necessary program state from the unstructured hydrodynamics mesh to a structured radiation transport mesh and back every timestep. The transfer operation is neither free nor exact. Research on extending IMD and DDMC to unstructured meshes aims to efficiently address the challenges inherent to unstructured meshes, rather than relying on mesh transfers to circumvent these issues. This is because the cost and approximation errors associated with the transfer operation are

significant enough to make it impractical.

One notable strength of Monte Carlo for solving the linear transport equation is that it can provide higher solution quality than S_N for a fixed amount of time and resources when calculations require many photon energy groups. Increasing the number of groups increases the runtime of MC much more slowly than S_N , because the S_N problem size is multiplicative in the number of groups. The MC problem size is multiplicative in the number of MC particles and the number of spatial elements, but it is not multiplicative in the number of photon energy groups.

MC also avoids angular discretization errors, known as ray effects, which can create unphysical starburst patterns in S_N solutions. Finally, MC solutions can never be negative, as long as the energy-weights of the IMC photons are non-negative, which is easily ensured by MC algorithms. This contrasts with S_N sweeps, which can produce unphysical negative solutions. Non-negativity follows from the properties of the TRT equations, specifically the fact that the physical photon number density cannot be negative, and so the radiation specific intensity also cannot be negative. Negativity in deterministic methods arises from under-resolution in the discretization, for example when the solution has substantial variation within a mesh element. If there is substantial intra-element solution variation, neither high-order nor low-order methods are accurate, but lower-order methods can avoid negativities. Deterministic methods which use techniques to preserve positivity can be less accurate than nonpositive schemes due to a fundamental tradeoff between positivity and accuracy [20]. Positivity-preserving schemes often try to limit to low-order methods in under-resolved regions, and high-order methods when resolved.

The strengths of Monte Carlo are compensation for its two main weaknesses. The first is random variability, also known as statistical noise. The MC noise can severely diminish solution quality compared to S_N . The second is convergence order. Spatial discretizations for S_N can be arbitrarily high-order, providing favorable convergence orders with respect to the spatial mesh element width. For example, doubling the number of spatial elements would decrease the spatial discretization error for a quadratic method by a factor of four, and a cubic method by a factor of eight. This compares favorably with Monte Carlo convergence order which has an unfavorable dependence on the number of MC photons. For example, increasing the number of MC photons by a factor of four would only reduce the noise in the MC estimator by a factor of two.

Noise can be ameliorated but never eliminated. Many Monte Carlo calculations employ a variance reduction technique called survival biasing, and other variance reduction techniques are often helpful too [21]. Another noise amelioration strategy uses quasi-random numbers instead of pseudorandom numbers [22]. An example of one such low-discrepancy sequence is the Halton sequence [23]. Its use improves the MC convergence order: a factor of two decrease in MC noise would require only a factor of two increase in MC photons, not a factor of four increase. Noise complicates hybrid methods which combine MC with deterministic methods. In the worst cases, a noisy quantity computed with MC is differentiated, which causes noise amplification. Hybrid methods with noise amplification may be unusable.

1.3.5 Moment Methods for Linear Transport

Moment methods are a rich source of useful numerical methods applied to a variety of problems in various scientific fields [24]. Moment methods are derived by taking moments of the governing equations of the system under study. Mathematically, a moment is an integral, where the integration is performed with respect to a variable in the governing equations. Moment methods often take multiple moments and apply a closure because the resulting moment system is typically unclosed. An unclosed system of equations has more unknowns than equations. A closure provides more equations. If a closure provides enough equations to close the system, then the system is closed, meaning that it has as many equations as unknowns.

The term “closure” is sometimes used synonymously with “approximation” because closures often involve introducing approximations to the continuous equations. In part i) below, the closures are indeed approximations. However, in part ii), the closures are not approximations. Instead, they are exact in the continuous equations and only become approximations after discretization. I use the terms “closure approximation” and “exact closure” to distinguish between these two situations.

Two reasons to take moments of an equation are: i) to derive new equations which approximate the original equation by using a closure approximation, and ii) to derive new equations which are an equivalent reformulation of the original equation by using an exact closure. The resulting equations can be easier to solve than the original equation. I demonstrate cases i) and ii) with four examples, two for each case. The fourth example is the system of equations which I solve in the deterministic component of my novel hybrid method.

i) Deriving approximating equations

The first example takes moments to derive the Euler and Navier-Stokes equations of hydrodynamics. The Boltzmann equation for the particle number density $f = f(\mathbf{x}, \mathbf{v}, t)$ is,

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla f + \frac{\mathbf{F}}{m} \cdot \nabla_{\mathbf{v}} f = \left(\frac{\partial f}{\partial t} \right)_{\text{collision}}. \quad (1.16)$$

In Eq. (1.16), m is the particle mass and $\nabla_{\mathbf{v}}$ is the gradient in velocity space. The force term \mathbf{F} accounts for any forces acting on the particles, while the collision term $(\partial f / \partial t)_{\text{collision}}$ describes the effects of collisions. These forces and collisions can occur either between the particles themselves or between the particles and a surrounding medium. In systems with significant collective effects, such as plasmas or self-gravitating systems, inter-particle forces are typically included in the force term. Conversely, in systems without collective effects, inter-particle forces are instead incorporated into the collision term.

An example of a force is the gravitational force, $\mathbf{F} = -m\nabla\Phi$, where m is the particle mass, and $\Phi = -GM/r$ is the gravitational potential for a point mass M at a distance r , with G being the gravitational constant. Another example is the Coulomb force, $\mathbf{F} = q(\mathbf{E} + \mathbf{v} \times \mathbf{B})$,

where \mathbf{E} is the electric field and \mathbf{B} is the magnetic field. If the electric field \mathbf{E} and the magnetic field \mathbf{B} arise from external forces—that is, forces not generated by the particles described by Eq. (1.16)—their effects are included in the force term. Otherwise, their effects are accounted for in the collision term.

Taking the zeroth, first, and second mass-weighted velocity moments of Eq. (1.16) yields a system of five equations. One equation describes mass conservation, three describe momentum conservation, and one describes energy conservation. The system is unclosed because it has more unknowns than equations. Closing the system using a closure approximation which assumes that the material particle velocity distribution is Maxwellian, produces the inviscid Euler equations. An alternative closure approximation known as the Chapman-Enskog expansion produces the Navier-Stokes equations [25]. The Euler and Navier-Stokes equations are referred to as “continuum equations” because they model a continuous mass of matter in which the individual particles which constitute the matter cannot be distinguished. In contrast, Eq. (1.16) models the behavior of individual particles. The continuum equations serve as approximations of the particle-based model.

The second example takes moments to derive the radiation diffusion approximation. First, define the angle integrated intensity ϕ , the current \mathbf{J} , and the pressure \mathbf{P} as the zeroth, first, and second angular moments of the intensity ψ ,

$$\phi(\mathbf{x}) = \int_{\mathbb{S}^2} \psi(\mathbf{x}, \boldsymbol{\Omega}) \, d\Omega, \quad (1.17)$$

$$\mathbf{J}(\mathbf{x}) = \int_{\mathbb{S}^2} \boldsymbol{\Omega} \psi(\mathbf{x}, \boldsymbol{\Omega}) \, d\Omega, \quad (1.18)$$

$$\mathbf{P}(\mathbf{x}) = \int_{\mathbb{S}^2} \boldsymbol{\Omega} \otimes \boldsymbol{\Omega} \psi(\mathbf{x}, \boldsymbol{\Omega}) \, d\Omega. \quad (1.19)$$

Now take the zeroth and first angular moments of Eq. (1.13a) to get,

$$\nabla \cdot \mathbf{J} + \sigma_a \phi = Q_0, \quad (1.20a)$$

$$\nabla \cdot \mathbf{P} + \sigma_t \mathbf{J} = \mathbf{Q}_1, \quad (1.20b)$$

where Q_0 and \mathbf{Q}_1 are the zeroth and first angular moments of the fixed source q . The system of Eqs. (1.20a) and (1.20b) is unclosed because it has ten unknowns and only four equations: one radiation relative mass (energy) conservation equation and three radiation momentum conservation equations. The independent variables ϕ , \mathbf{J} , and \mathbf{P} consist of 1+3+6=10 unknowns because they are a scalar, vector, and symmetric tensor, respectively. The closure approximation that leads to the radiation diffusion approximation assumes that ψ is a linearly anisotropic function that can be written,

$$\psi(\mathbf{x}, \boldsymbol{\Omega}) = \frac{1}{4\pi} (\phi(\mathbf{x}) + 3\boldsymbol{\Omega} \cdot \mathbf{J}(\mathbf{x})). \quad (1.21)$$

Eq. (1.21) is a spherical harmonics expansion truncated at the linear term, thus only the constant and linear terms persist. Evaluating Eq. (1.19) using Eq. (1.21) leads to Eddington's approximation,

$$\mathbf{P} = \frac{1}{3}\phi\mathbf{I}. \quad (1.22)$$

Substituting Eq. (1.22) into Eq. (1.20b) gives,

$$\frac{1}{3}\nabla\phi + \sigma_t\mathbf{J} = \mathbf{Q}_1. \quad (1.23)$$

Derive a boundary condition by defining $J_n^\pm = \int_{\Omega \cdot \mathbf{n} \gtrless 0} \Omega \cdot \mathbf{n} \psi \, d\Omega$ and performing algebraic manipulation,

$$\begin{aligned} \mathbf{J} \cdot \mathbf{n} &= J_n^- + J_n^+ \\ &= 2J_n^- + (J_n^+ - J_n^-) \\ &= 2J_n^- + \int_{\mathbb{S}^2} |\Omega \cdot \mathbf{n}| \psi \, d\Omega. \end{aligned} \quad (1.24)$$

Defining

$$B(\psi) = \int_{\mathbb{S}^2} |\Omega \cdot \mathbf{n}| \psi \, d\Omega, \quad (1.25)$$

and

$$J_{\text{in}} = \int_{\Omega \cdot \mathbf{n} < 0} \Omega \cdot \mathbf{n} \bar{\psi} \, d\Omega, \quad (1.26)$$

and substituting into Eq. (1.24) gives the unclosed boundary condition,

$$\mathbf{J} \cdot \mathbf{n} = B(\psi) + 2J_{\text{in}}. \quad (1.27)$$

Substituting the closure approximation Eq. (1.21) into $B(\psi)$ gives,

$$\begin{aligned} B(\psi) &= \int_{\mathbb{S}^2} |\Omega \cdot \mathbf{n}| \psi \, d\Omega \\ &= \frac{1}{4\pi} \int_{\mathbb{S}^2} |\Omega \cdot \mathbf{n}| (\phi + 3\Omega \cdot \mathbf{J}) \, d\Omega \\ &= \frac{1}{4\pi} \left(\phi \int_{\mathbb{S}^2} |\Omega \cdot \mathbf{n}| \, d\Omega + 3 \int_{\mathbb{S}^2} |\Omega \cdot \mathbf{n}| \Omega \cdot \mathbf{J} \, d\Omega \right) \\ &= \frac{1}{4\pi} (2\pi\phi + 0) \\ &= \frac{\phi}{2}. \end{aligned} \quad (1.28)$$

Substituting the above result into Eq. (1.27) gives the radiation diffusion boundary condition,

$$\mathbf{J} \cdot \mathbf{n} = \frac{\phi}{2} + 2J_{\text{in}}. \quad (1.29)$$

The Eqs. (1.20a), (1.23) and (1.29) together form the radiation diffusion approximation system. Using Eq. (1.23) to solve for \mathbf{J} in terms of ϕ , and substituting this expression for \mathbf{J} into Eq. (1.20a), gives the second order form of the radiation diffusion approximation,

$$-\nabla \cdot \frac{1}{3\sigma_t} \nabla \phi + \sigma_a \phi = Q_0 - \nabla \cdot \frac{\mathbf{Q}_1}{\sigma_t}. \quad (1.30)$$

The first order form has four unknowns and four equations. The second order form has only one equation and one unknown. Radiation diffusion is an approximation of Eqs. (1.13a) and (1.13b) because if ψ is more than linearly anisotropic (e.g. quadratic, cubic, etc.) then the solution to Eqs. (1.29) and (1.30) disagrees with Eq. (1.17). This is not true of the next two examples, which are equivalent reformulations of Eqs. (1.13a) and (1.13b).

The radiation diffusion approximation, when expressed in its first order form, is commonly referred to as the P_1 equations or the P_1 approximation. The term “ P_1 ” refers to the linear term in the spherical harmonics expansion, which is the term at which we truncate the expansion in Eq. (1.21). The P_1 equations can also be written in time-dependent form. If we had taken moments of the unsteady transport equation, and then applied the closure approximation Eq. (1.21), we would have arrived at the unsteady P_1 equations, which is the system of Eqs. (1.20a) and (1.23) along with additional time derivative terms:

$$\frac{1}{c} \frac{\partial \phi}{\partial t} + \nabla \cdot \mathbf{J} + \sigma_a \phi = Q_0, \quad (1.31a)$$

$$\frac{1}{c} \frac{\partial \mathbf{J}}{\partial t} + \frac{1}{3} \nabla \phi + \sigma_t \mathbf{J} = \mathbf{Q}_1. \quad (1.31b)$$

The unsteady radiation diffusion approximation, expressed in its second order form, can be derived from the unsteady P_1 Eqs. (1.31a) and (1.31b) by making the approximation that the time variation of the current \mathbf{J} is much smaller than the spatial gradient of the angle integrated intensity ϕ . That is, the first term in Eq. (1.31b) is much smaller than the second,

$$\frac{1}{c} \frac{\partial \mathbf{J}}{\partial t} \ll \frac{1}{3} \nabla \phi. \quad (1.32)$$

Assuming Eq. (1.32) allows for the elimination of the current by substituting $(1/c)\partial\mathbf{J}/\partial t = 0$ into Eq. (1.31b), solving for \mathbf{J} in terms of ϕ , and then substituting the result into Eq. (1.31a),

$$\frac{1}{c} \frac{\partial \phi}{\partial t} - \nabla \cdot \frac{1}{3\sigma_t} \nabla \phi + \sigma_a \phi = Q_0 - \nabla \cdot \frac{\mathbf{Q}_1}{\sigma_t}. \quad (1.33)$$

Eq. (1.33) is a parabolic equation with a rich literature of numerical method development. An important modeling issue with Eq. (1.33) is that it allows radiation to propagate faster than the speed of light, which is an unphysical consequence of setting $(1/c)\partial\mathbf{J}/\partial t = 0$. More information, as well as solutions to Eq. (1.33) for example radiative transfer problems, may be found in the manuscript by Brunner [26].

ii) Deriving equivalent reformulations

The third and fourth examples, Variable Eddington Factor (VEF) and Second Moment Method (SMM) arise from alternative choices for Eqs. (1.22) and (1.25). Unlike in part i), where closure approximations were used, we now employ exact closures, avoiding any approximations. As a result, the derived equations are equivalent reformulations of the original equations.

Multiplying and dividing by the angle integrated intensity is the choice that leads to VEF. The derivation of the VEF method begins by writing the following exact closure, which is multiplicative in the angle integrated intensity, and is therefore a nonlinear equation,

$$\mathbf{P} = \mathbf{E}\varphi. \quad (1.34)$$

In Eq. (1.34), I switched my notation for the angle integrated intensity from ϕ (phi) to φ (varphi) in order to be consistent with [7]. The quantity \mathbf{E} is called the Eddington tensor,

$$\mathbf{E} = \frac{\int_{\mathbb{S}^2} \boldsymbol{\Omega} \otimes \boldsymbol{\Omega} \psi \, d\Omega}{\int_{\mathbb{S}^2} \psi \, d\Omega}. \quad (1.35)$$

Substituting Eq. (1.34) into Eq. (1.20b) gives,

$$\nabla \cdot (\mathbf{E}\varphi) + \sigma_t \mathbf{J} = \mathbf{Q}_1. \quad (1.36)$$

Write another exact closure to close the boundary functional Eq. (1.25) in the same manner as we did for Eq. (1.34). Specifically, multiply and divide by the angle integrated intensity,

$$B(\psi) = E_b \varphi, \quad (1.37)$$

where

$$E_b = \frac{\int_{\mathbb{S}^2} |\boldsymbol{\Omega} \cdot \mathbf{n}| \psi \, d\Omega}{\int_{\mathbb{S}^2} \psi \, d\Omega} \quad (1.38)$$

is the Eddington boundary factor. Substituting Eq. (1.37) into Eq. (1.27) gives the VEF boundary condition,

$$\mathbf{J} \cdot \mathbf{n} = E_b \varphi + 2J_{\text{in}}. \quad (1.39)$$

The Eqs. (1.20a), (1.36) and (1.39) together form the VEF system. Using Eq. (1.36) to eliminate \mathbf{J} from the system gives the second order form of the VEF equation,

$$-\nabla \cdot \frac{1}{\sigma_t} \nabla \cdot (\mathbf{E}\varphi) + \sigma_a \varphi = Q_0 - \nabla \cdot \frac{\mathbf{Q}_1}{\sigma_t}. \quad (1.40)$$

Instead of multiplying and dividing by the angle integrated intensity, we can add and subtract it, which is the choice that leads to the SMM. The derivation of the SMM begins by writing the following exact closure, which is additive in the angle integrated intensity, and is therefore a linear equation,

$$\mathbf{P} = \mathbf{T} + \frac{1}{3} \mathbf{I}\varphi, \quad (1.41)$$

where

$$\mathbf{T} = \int_{\mathbb{S}^2} \boldsymbol{\Omega} \otimes \boldsymbol{\Omega} \psi \, d\Omega - \frac{1}{3} \mathbf{I} \int_{\mathbb{S}^2} \psi \, d\Omega \quad (1.42)$$

is called the SMM correction tensor. Substituting Eq. (1.41) into Eq. (1.20b) gives,

$$\frac{1}{3} \nabla \varphi + \sigma_t \mathbf{J} = \mathbf{Q}_1 - \nabla \cdot \mathbf{T}. \quad (1.43)$$

Write another exact closure to close the boundary functional Eq. (1.25) in the same way as we did for Eq. (1.41). Specifically, add and subtract the angle integrated intensity,

$$B(\psi) = \beta + \frac{1}{2} \varphi, \quad (1.44)$$

where

$$\beta(\psi) = \int_{\mathbb{S}^2} |\boldsymbol{\Omega} \cdot \mathbf{n}| \psi \, d\Omega - \frac{1}{2} \int_{\mathbb{S}^2} \psi \, d\Omega \quad (1.45)$$

is called the SMM boundary correction factor. Substituting Eq. (1.44) into Eq. (1.27) gives the SMM boundary condition,

$$\mathbf{J} \cdot \mathbf{n} = \frac{1}{2} \varphi + 2J_{\text{in}} + \beta. \quad (1.46)$$

The Eqs. (1.20a), (1.43) and (1.46) together form the SMM system. Using Eq. (1.43) to eliminate \mathbf{J} from the system gives the second order form of the SMM equation,

$$-\nabla \cdot \frac{1}{3\sigma_t} \nabla \varphi + \sigma_a \varphi = Q_0 - \nabla \cdot \frac{\mathbf{Q}_1}{\sigma_t} + \nabla \cdot \frac{1}{\sigma_t} \nabla \cdot \mathbf{T}. \quad (1.47)$$

The solutions to both Eq. (1.40) and Eq. (1.47) agree exactly with Eq. (1.17) because Eqs. (1.34) and (1.41) are exact closures. In practice, we do not know ψ , so implementations of VEF or SMM must simultaneously solve Eq. (1.13a) for ψ to compute the functionals \mathbf{E} and E_b or \mathbf{T} and β , which are required to solve the VEF or SMM equations, respectively.

Discretization causes the VEF and SMM solutions to differ from the transport solution by an amount that is on the order of the discretization error, which goes to zero in the limit of a discretization with infinite resolution. A ‘‘consistent’’ discretization of VEF or SMM is one in which the VEF or SMM solution is equal to the transport solution. Consistent VEF and SMM discretizations have restricted flexibility in discretization choices compared to VEF and SMM discretizations which are not consistent with the transport discretization. Thus, consistency with transport can be achieved, albeit at the expense of flexibility in the moment system discretization.

1.3.6 Thick Diffusion Limit

Both deterministic and Monte Carlo methods can be used to solve the linear transport equation that arises from the IMC linearization. Deterministic and Monte Carlo methods commonly use source iteration (SI) and effective scattering events (ESE), respectively. Both SI and ESE cause significant prolongation of calculation runtimes in the thick diffusion limit (TDL). For SI, this is because the spectral radius of the streaming and collision operator L in Eq. (1.14) approaches unity. For ESE, this is because the effective scattering probability approaches unity. My hybrid method is a novel way to avoid ESE.

The TDL is a physical regime described mathematically by Larsen, Morel, and Miller with the dimensionless scaling parameter $\epsilon \in (0, 1]$, which they used to scale the data in the model problem as follows [9],

$$\sigma_t = 1/\epsilon, \quad (1.48a)$$

$$\sigma_a = \epsilon, \quad (1.48b)$$

$$\sigma_s = \sigma_t - \sigma_a, \quad (1.48c)$$

$$q = \epsilon. \quad (1.48d)$$

The scaled model problem is thus,

$$\boldsymbol{\Omega} \cdot \nabla \psi + \frac{\sigma_t}{\epsilon} \psi = \left(\frac{\sigma_t}{\epsilon} - \epsilon \sigma_a \right) \frac{1}{4\pi} \int_{\mathbb{S}^2} \psi \, d\Omega' + \epsilon q. \quad (1.49)$$

Larsen, Morel, and Miller show that the asymptotic solution to Eq. (1.49) for $\epsilon \ll 1$ is,

$$\psi(\mathbf{x}, \boldsymbol{\Omega}) = \phi(\mathbf{x}) + O(\epsilon), \quad (1.50)$$

where $\phi(\mathbf{x})$ is defined by the equation,

$$-\nabla \cdot \frac{1}{3\sigma_t} \nabla \phi + \sigma_a \phi = q, \quad (1.51)$$

which is the radiation diffusion approximation. Thus, when applied to the continuous transport equation, the TDL analysis provides a mathematical justification for using the radiation diffusion approximation in regions of phase space where $\epsilon \ll 1$. Larsen, Morel, and Miller also found that, when they applied their analysis to the discretized transport equation, they could predict whether the solution of the discretized equation would remain accurate in the TDL. Their analysis provided a theoretical explanation for empirical results obtained with popular discrete ordinates spatial discretizations, such as diamond-difference and linear discontinuous Galerkin.

Accuracy and speed when calculating solutions in the TDL is a requirement of any method for solving the IMC equations, regardless of whether the method is deterministic, Monte Carlo or both (hybrid). Methods must be accurate and fast in the TDL and any method which lacks either quality is insufficient for solving the most difficult TRT problems. Examples of existing deterministic methods satisfying this requirement include DSA-accelerated

SI S_N as well as S_N moment methods. Examples for Monte Carlo include IMC with RW, IMD, or DDMC.

Finally, notice that if we substitute the closure approximation Eq. (1.21), which gave us the radiation diffusion approximation Eq. (1.30), into the Eddington tensor Eq. (1.34) and the Eddington boundary factor Eq. (1.38), we get $\mathbf{E} = (1/3)\mathbf{I}$ and $E_b = 1/2$, respectively. Similarly, if we substitute Eq. (1.21) into the SMM correction tensor Eq. (1.42) and the SMM boundary correction factor Eq. (1.45), we get $\mathbf{T} = \mathbf{0}$ and $\beta = 0$, respectively. The result in both cases is that the moment methods, which are equivalent reformulations of transport, collapse to the radiation diffusion approximation. Thus, when radiation diffusion is a good approximation of the transport solution, such as in the TDL, $\mathbf{E} \approx (1/3)\mathbf{I}$, $E_b \approx 1/2$, $\mathbf{T} \approx \mathbf{0}$, and $\beta \approx 0$.

1.4 Hybrid Second Moment Method Summary

“Hybrid second moment (HSM)” are the most illustrative three words for describing my novel method. “Hybrid” refers to the combination of the Monte Carlo method, which is a stochastic method, with a deterministic method. “Second moment” refers to the SMM, a method that I introduced in section 1.3.5. The SMM requires simultaneous solution of the linear transport equation and the SMM system in an iteration. Fig. 1.3 shows the SMM algorithm. My HSM method uses the Monte Carlo method without scattering events to solve the linear transport equation (left side of Fig. 1.3) and compute the SMM data \mathbf{T} and β as Monte Carlo estimators. I then solve the SMM equations (right side of Fig. 1.3) using a deterministic method, and use the solution φ to compute the scattering source, which I converge in an iteration. Chapter 2 describes the deterministic component of HSM which I use to compute the scattering source, Chapter 3 describes the Monte Carlo component which I use to estimate the SMM data, Chapter 4 describes my implementation of the combined components, and Chapter 5 demonstrates numerical results using HSM to solve linear transport problems.

The diagram illustrates the SMM algorithm. It consists of three main parts: a transport equation on the left, SMM equations on the right, and formulas for computing SMM data at the bottom. Arrows indicate the flow of information: φ is used to compute the scattering source in the transport equation; the transport equation and SMM equations are solved simultaneously; the solution φ is used to compute the SMM data \mathbf{T} and β .

$$\begin{aligned}
 & \mathbf{\Omega} \cdot \nabla \psi + \sigma_t \psi = \frac{\sigma_s}{4\pi} \varphi + q, & \nabla \cdot \mathbf{J} + \sigma_a \varphi = Q_0, \quad \mathbf{x} \in \mathcal{D}, \\
 & \psi(\mathbf{x}, \mathbf{\Omega}) = \bar{\psi}(\mathbf{x}, \mathbf{\Omega}), \quad \mathbf{x} \in \partial \mathcal{D} \text{ and } \mathbf{\Omega} \cdot \mathbf{n} < 0. & \frac{1}{3} \nabla \varphi + \sigma_t \mathbf{J} = \mathbf{Q}_1 - \nabla \cdot \mathbf{T}, \quad \mathbf{x} \in \mathcal{D}, \\
 & & \mathbf{J} \cdot \mathbf{n} = \frac{1}{2} \varphi + 2J_{\text{in}} + \beta, \quad \mathbf{x} \in \partial \mathcal{D}. \\
 & & \mathbf{T}(\psi) = \int_{\mathbb{S}^2} \mathbf{\Omega} \otimes \mathbf{\Omega} \psi \, d\Omega - \frac{1}{3} \mathbf{I} \int_{\mathbb{S}^2} \psi \, d\Omega \\
 & & \beta(\psi) = \int_{\mathbb{S}^2} |\mathbf{\Omega} \cdot \mathbf{n}| \psi \, d\Omega - \frac{1}{2} \int_{\mathbb{S}^2} \psi \, d\Omega
 \end{aligned}$$

Figure 1.3: SMM algorithm [7].

1.5 Similar Hybrid Methods

The fundamental quality of HSM is the use of Monte Carlo to calculate estimators based on a linear transport problem, combined with a deterministic method for computing the solution to a moment system. The moment system could be transport equivalent, such as VEF and SMM, or transport approximant, such as radiation diffusion. The Monte Carlo estimators are used in the moment solve and the deterministic solution is used in the transport solve.

There are several methods which share this fundamental quality with HSM. These similar methods can be divided into two categories: those which iterate and those which do not. I examine the latter in section 1.5.1 Interrupted Monte Carlo and the former in section 1.5.2 Iterated Monte Carlo. My intention is to show how HSM differs.

A third class of methods worth mentioning partition phase space into two non-overlapping regions and solve a linear transport equation in one region and a radiation diffusion approximation in the other. The three most notable methods in this class, all of which I mentioned earlier in this Introduction, are RW [14], IMD [15–17], and DDMC [18, 19]. They are similar to HSM in that they also attempt to improve slow runtimes due to high effective scattering, but they differ in that they do not iterate because the moment system that they solve is closed using Eddington’s approximation. They also differ from HSM by partitioning phase space, which HSM does not.

1.5.1 Interrupted Monte Carlo

Interrupted Monte Carlo methods interrupt the Monte Carlo linear transport solve with a moment solve. The moment solve uses estimators computed during the MC solve. In the first case that I describe below, the authors use the moment solution in the MC solve that they resume after the interruption. In the second case, the authors do not resume the MC solve. Neither iterate.

Cooper and Larsen’s Hybrid VEF

Cooper and Larsen [27] combined Monte Carlo with VEF in a hybrid method for solving the linear transport equation. Their method uses Monte Carlo to compute the angle integrated intensity—which is what nearly all Monte Carlo methods do—but they additionally compute the Eddington tensor. They pause the Monte Carlo calculation once they have an adequate estimate for the Eddington tensor. They resume the calculation after solving the VEF equation and using the VEF solution to compute weight windows. A weight window is a variance reduction technique that splits high-weight particles and kills low-weight particles. The method that Cooper and Larsen use to reduce the noise in their Monte Carlo estimate of the Eddington tensor is running more particles. Specifically, they wait until “a sufficient number of histories have been simulated to ensure good estimates of the Eddington factors” before temporarily halting the Monte Carlo calculation to solve VEF.

My HSM contrasts with Cooper and Larsen’s Hybrid VEF by solving SMM instead of VEF. Also, their method appears to have noise amplification due to twice differentiation of the Eddington tensor in the second order form of the VEF equation that they solve, whereas I avoid noise amplification by solving the first order SMM system with a finite element method. Finally, my method converges the moment solve in an iteration, in contrast to Cooper and Larsen who use an unconverged VEF solution to compute weight windows for Monte Carlo variance reduction.

Novellino and Anistratov’s Hybrid VEF and SMM

Novellino and Anistratov [28, 29] use Monte Carlo to estimate the solution to a 1D linear transport equation while simultaneously accumulating estimators for the VEF and SMM data. When their Monte Carlo calculation finishes, they use a finite volume discretization to compute the solution of the VEF and SMM systems, and then compare the Monte Carlo estimate of the solution and the moment solutions to a reference solution. They find that the moment solution is more accurate than the MC estimate for most of the MC pseudo-random number generator seeds that they tried. This is a great example of a method similar to HSM because the authors are using a Monte Carlo solve to compute estimators for the SMM data, which is exactly what I do, but there are some differences.

For example, Novellino and Anistratov include scattering events and they do not exercise the thick diffusion limit, perhaps because their stated focus is neutronics, not radiative transfer. The authors solve the second order form of the SMM system using a finite volume discretization. Thus, their method appears to have noise amplification due to twice differentiation of the SMM correction tensor, whereas I avoid noise amplification by solving the first order SMM system with a finite element method. The authors do not iterate: they do one Monte Carlo solve followed by one moment solve. I iterate to converge the scattering source.

Finally, Novellino and Anistratov’s choice to solve a 1D linear transport problem differs from my choice to solve a 2D linear transport problem. As I explain in section 4.4.1 Dimensionality, making the assumption that the solution varies only along one spatial dimension can hide numerical problems which arise in the absence of azimuthal symmetry.

1.5.2 Iterated Monte Carlo

Iterated Monte Carlo methods iterate until convergence of some quantity. Each cycle of the iteration includes a Monte Carlo linear transport solve and deterministic solve of a moment system or a diffusion system. The deterministic solve uses estimators computed during the MC solve, and the MC solve uses the solution computed using the deterministic method.

Park, Densmore, Wollaber, Knoll, and Rauenzahn's Hybrid HOLO

Chacón *et al.* [24] published a review of moment methods, which they call high-order/low-order (HOLO) algorithms. In their review, they republished results showing a solution of the unsteady gray TRT equations using Park, Densmore, Wollaber, Knoll, and Rauenzahn's hybrid HOLO method [30]. Figures 2 and 3 in [24] come from Figures 3 and 4 in [30], in which the authors describe using Monte Carlo to hybridize a moment method derived in Park *et al.* [31]. The starting point of their hybrid moment method is the VEF system of equations (11) and (12) in [30]. Their subsequent system of equations (13) and (14) arise after making two changes to the radiation momentum conservation equations. They say,

1. the Eddington tensor is replaced by $1/3$ (i.e., the P_1 approximation), and
2. consistency terms γ_{ij}^\pm are added to match the truncation errors.

We can evaluate γ_{ij}^\pm by substituting the HO solution moments (i.e., E^{HO} , \mathbf{F}^{HO}) into Eq. (14).

That is, they use Monte Carlo to compute a radiation energy E and flux \mathbf{F} , then substitute it into the radiation momentum conservation equations to get expressions for γ_{ij}^\pm . Their expressions are equations (15) and (16), which are

$$\Delta_t F_{ij}^+ + \frac{c^2}{6} (\Delta_x E)_{ij} + c\sigma_{ij} F_{ij}^+ = \gamma_{ij}^+ cE_i, \quad (1.52)$$

$$\Delta_t F_{ij}^- - \frac{c^2}{6} (\Delta_x E)_{ij} + c\sigma_{ij} F_{ij}^- = \gamma_{ij}^- cE_j, \quad (1.53)$$

These expressions are written less concisely as equations (33) and (34) in [31]. The less-concise presentation differs slightly because the Eddington tensor \mathcal{E} has not been replaced. There may also be unintentional differences. For example, the sign on the c^2 term is not flipped. Nonetheless, the heavier notation may be clarifying for someone unfamiliar with their work, like myself. Equations (33) and (34) in [31] are,

$$\gamma_{i,j}^{+,n+1} = -\frac{F_{i,j}^{+,n+1,HO}(1 + c\Delta t\sigma_{i,j}^{n+1}) - F_{i,j}^{+,n,HO} + c^2\Delta t\frac{\vec{n}_{i,j}}{2} \cdot (\nabla \cdot \mathcal{E}^{n+1}E^{n+1,HO})|_{i,j}}{cE_i^{n+1,HO}}, \quad (1.54)$$

$$\gamma_{i,j}^{-,n+1} = -\frac{F_{i,j}^{-,n+1,HO}(1 + c\Delta t\sigma_{i,j}^{n+1}) - F_{i,j}^{-,n,HO} + c^2\Delta t\frac{\vec{n}_{i,j}}{2} \cdot (\nabla \cdot \mathcal{E}^{n+1}E^{n+1,HO})|_{i,j}}{cE_j^{n+1,HO}}, \quad (1.55)$$

The $\Delta_x E$ terms in Eqs. (1.52) and (1.53) and $\nabla \cdot \mathcal{E}^{n+1}E^{n+1,HO}$ terms in Eqs. (1.54) and (1.55) are derivatives that appear to amplify the Monte Carlo noise in E . I cannot find where the authors address noise amplification, but they address the noise itself with two techniques. The first is survival biasing, or continuous energy deposition, which is perhaps the most popular IMC variance reduction technique. The second technique is the replacement of

transport zones with diffusion zones in optically-thick materials. The authors describe the two techniques in section 3.1.1 and 3.2.1 in [30].

My HSM contrasts with Park, Densmore, Wollaber, Knoll, and Rauenzahn’s hybrid HOLO by solving SMM. A quality that the methods share is that both solve a radiation diffusion system modified to include a transport correction which appears as a source term in the radiation momentum conservation equations. Both compute the source term using the Monte Carlo solution. Both solve first order systems. However, their hybrid HOLO method appears to suffer noise amplification due to differentiation of the radiation energy in their expressions for γ_{ij}^{\pm} , whereas I avoid it by using a finite element method to discretize the first order SMM system. They have transport and diffusion zones, which means that they solve different equations in different materials, whereas I solve the same equations everywhere. Finally, they demonstrate their method on the unsteady TRT equations, whereas I demonstrate my method on the steady-state linear transport equation, which is simpler.

Willert’s JFNK-NDA(MC)

Willert [32] introduced a method which Willert calls Jacobian-free Newton-Krylov nonlinear diffusion acceleration Monte Carlo (JFNK-NDA(MC)). Newton’s method is a method for solving systems of nonlinear equations. Newton’s method converges quadratically if the initial guess is close to the solution of the nonlinear system and if the nonlinear functions in the system are sufficiently smooth. Quadratic convergence means that the number of correct digits in the solution approximation doubles with each iteration. Newton’s method requires forming and evaluating the Jacobian matrix of the nonlinear system at each iterate, which may be expensive, impractical, or impossible. Jacobian-free methods attempt to preserve the quadratic convergence of Newton’s method without forming and evaluating the Jacobian matrix. Jacobian-Free Newton-Krylov (JFNK) methods use a Krylov subspace method to solve the resulting linear system of equations. Nonlinear diffusion acceleration (NDA) is a method for accelerating the iterative convergence of SI by solving a diffusion approximation with a consistency term computed using the result of the transport solve. The coarse mesh finite difference (CMFD) method is widely used for solving the diffusion approximation in the NDA framework. In fact, the terms CMFD and NDA are sometimes used interchangeably, even though NDA methods can employ approaches other than CMFD to solve the diffusion approximation. The scattering source is then computed using the solution of the diffusion approximation instead of the angle integrated transport solution. Knoll *et al.* [33] apply JFNK to NDA. Willert’s JFNK-NDA(MC) is a hybrid method which uses Monte Carlo to replace the deterministic transport sweep in [33].

An important quality that my writing about my HSM method in this dissertation shares with Willert’s writing is our expressed desire to avoid noise amplification. I avoid noise amplification by using a finite element discretization of the first order form of the moment system. Willert avoids noise amplification by computing an analytic Jacobian-vector product instead of using a finite difference formula for approximating the Jacobian-vector product. Willert also applies filtering to the noise in the NDA consistency term before using a finite

difference formula for approximating the derivative of the consistency term. The noise after filtering is still amplified by the differentiation, but the result should be less noisy than if the differentiation were performed on the unfiltered quantity. One important way in which our methods differ is the location of the terms that we compute during the transport solve and then use in the deterministic solve: Willert’s consistency term appears on the left-hand side, whereas my SMM correction tensor appears on the right-hand side.

Lam’s DAIMC

Lam [34] introduced a method which Lam calls Diffusion Accelerated Implicit Monte Carlo (DAIMC), for which Lam linearizes the TRT equations differently than Fleck and Cummings, and derives a linear transport equation which does not have effective scattering but requires iteration. Lam solves a diffusion equation to accelerate the convergence of the iteration. Lam uses DAIMC to solve a time-dependent gray 2D TRT system of equations in a single material. Solving the IMC linearization is equivalent to a single step of Newton’s method applied to the nonlinear TRT equations, whereas DAIMC converges the Newton iteration. Finally, Lam used a low-discrepancy sequence instead of a pseudo-random number generator to generate the values with which Lam sampled particle positions and directions and other quantities. This provided Lam with a Monte Carlo estimator uncertainty of $O(N^{-1})$, which is a significant improvement compared to the $O(N^{-1/2})$ uncertainty that arises with pseudo-random number generation.

The strongest similarity between Lam’s work and that of this dissertation is that neither has effective scattering events. The absence of effective scattering makes Lam’s work ideal for radiative transfer in the thick diffusion limit. One notable difference between Lam’s DAIMC and HSM is that Lam’s linearization removes effective scattering from the continuous DAIMC equations. The continuous HSM equations have effective scattering, but the HSM algorithm treats effective scattering using an iteration with a moment solve rather than including effective scattering events.

Pasmann’s iQMC

Pasmann [35] introduced a method which Pasmann calls iterative Quasi-Monte Carlo (iQMC), for which Pasmann uses Monte Carlo to estimate the solution to a energy-dependent steady-state 3D linear transport equation which has both a scattering source term and a fission source term, yet Pasmann’s Monte Carlo neutrons undergo neither scattering events nor fission events. Instead, Pasmann uses the Monte Carlo estimate of the solution from the previous cycle of his iteration to compute scattering and fission sources, and partitions the magnitude of these sources among the weights of his Monte Carlo neutrons. Pasmann iterates until convergence of the two sources. Pasmann uses a Krylov method to accelerate the convergence of the iteration. Pasmann resets the seed every cycle, meaning that his MC neutrons are sourced with the same locations and directions as the previous cycle, so that only their weights differ.

Pasmann also found that averaging, which is an alternative to resetting the seed, outperformed seed resetting, because it allowed for more sampling during the iteration. The procedure resembles the MC eigenvalue iteration process for neutronics criticality calculations in which eigenvalue estimates during a user-specified number of “inactive” cycles are discarded, and estimates during the “active” cycles which follow are averaged, until a convergence criterion involving the average is satisfied.

Finally, Pasmann used a low-discrepancy sequence instead of a pseudo-random number generator to generate the values with which Pasmann sampled particle positions and directions and other quantities. This provided Pasmann with a Monte Carlo estimator uncertainty of $O(N^{-1})$ which is a significant improvement compared to the $O(N^{-1/2})$ uncertainty that arises when one uses random numbers. Pasmann’s work is a great example of a method similar to HSM because Pasmann uses Monte Carlo without scattering events and iterates to converge the scattering source, which is exactly what I do, but there are some differences. One notable difference is that Pasmann does not solve a moment system, nor does Pasmann exercise the thick diffusion limit, perhaps because Pasmann’s stated focus is neutronics, not radiative transfer.

Chapter 2

Deterministic Component of Hybrid Second Moment

The deterministic component of HSM solves the SMM system of equations Eqs. (1.20a), (1.43) and (1.46) that I introduced in section 1.3.5. In Chapter 3, I describe how I use Monte Carlo to estimate the data for the SMM system. In this chapter, I show how I compute the solution of the SMM system by discretizing the SMM system using a mixed finite element method. The most important result in this chapter is Eqs. (2.22a) and (2.22b), which is the weak form for the first-order SMM system introduced in Olivier [7]. I begin this chapter with an overview of Olivier’s discretizations in section 2.1, for which Fig. 2.1 provides a concise summary. I follow with section 2.2, which justifies my choice of the mixed SMM discretization over the others, and summarizes the finite element method calculational procedure. I then build up finite element method concepts in the three sections sections 2.2.3 to 2.2.5 before finally deriving the weak form in section 2.2.6. I follow with a discussion of the convergence of the discretization in section 2.2.7, pseudocode in section 2.2.8, and a discussion of solvers and preconditioners in section 2.2.9. I finish with a brief discourse on the size of the linear system in section 2.2.10.

2.1 Olivier’s Finite Element Moment System Discretizations

Olivier [7] derived a family of high-order finite element discretizations for VEF and SMM on curved meshes. High-order means that Olivier’s discretizations give solutions that vary within a mesh element, instead of piecewise constant solutions which are constant within a mesh element. Olivier achieves this by computing expansion coefficients for solution polynomials of arbitrary degree.

Olivier used the high-order S_N solver in Haut *et al.* [36] to compute the functionals \mathbf{E} and \mathbf{T} and the other functionals of ψ which appear in the VEF and SMM boundary conditions (together they are called the VEF and SMM “data”). Olivier solved the problem of negative

S_N solutions by using the method in Yee *et al.* [37] to ensure positivity so that the VEF and SMM data are well-defined. Haut *et al.* [38] describe a DSA method for accelerated SI that is compatible with their high-order S_N method. Olivier’s methods can be viewed as an alternative to the aforementioned DSA solver. My method can be viewed as a hybridization of Olivier’s methods because I replace S_N with Monte Carlo.

I use Olivier’s mixed finite element method (FEM) discretization of the first order SMM system, which is one of the nine discretizations that Olivier presented in [7]. Fig. 2.1 shows Olivier’s nine discretizations as a 3×3 grid of points. The edges are directed and include labels indicating how the discretizations may be derived from the others. The only point with in-degree zero is the center left point which corresponds to the discontinuous VEF discretization that is the focus of Olivier *et al.* [39]. Furthermore, this one point actually contains three methods which Olivier derived by making three different choices for the “numerical flux” that arises in discontinuous FEM discretizations, as articulated by Arnold *et al.* [40]. The three discontinuous methods that Olivier presents are Interior Penalty (IP), Second Method of Bassi and Rebay (BR2), and Minimal Dissipation Local Discontinuous Galerkin (MDLDG).

The point in Fig. 2.1 corresponding to the Olivier discretization that I use in my method is the center bottom point: Olivier’s mixed discretization of the first order form of the SMM equations. Olivier derives the weak form by linearizing his mixed discretization of the first order form of the VEF equations. In section 2.2.6, I present an alternative derivation starting from the continuous first order SMM equations.

As I mentioned, my novel method contrasts with Olivier’s high-order SMM by using Monte Carlo instead of S_N to compute the SMM data, which turns Olivier’s method into a hybrid method, thus giving rise to the “H” in my HSM method. Olivier’s methods are high-order and handle curved meshes whereas my HSM method is lowest-order and requires straight-edge meshes. Extending HSM to high-order would require estimating polynomial SMM data which varies within a mesh element, instead of piecewise constant SMM data which is constant within a mesh element. Extending HSM to curved meshes would require tracking to curved surfaces because the boundaries of elements with curved edges in three spatial dimensions are curved surfaces.

2.2 Mixed Finite Element Discretization

2.2.1 Justification

Consider Fig. 2.1 once more and ask: which (if any) point in the 3×3 matrix is the best candidate for the deterministic component of a hybrid method for solving TRT? I want to solve transport, not diffusion, so I cannot use the right column. In **Advantages of SMM over VEF**, I explain why I selected the middle column instead of the left column, and in **Advantages of Mixed over Discontinuous and Continuous FEM**, I explain why I selected the bottom row of the middle column.

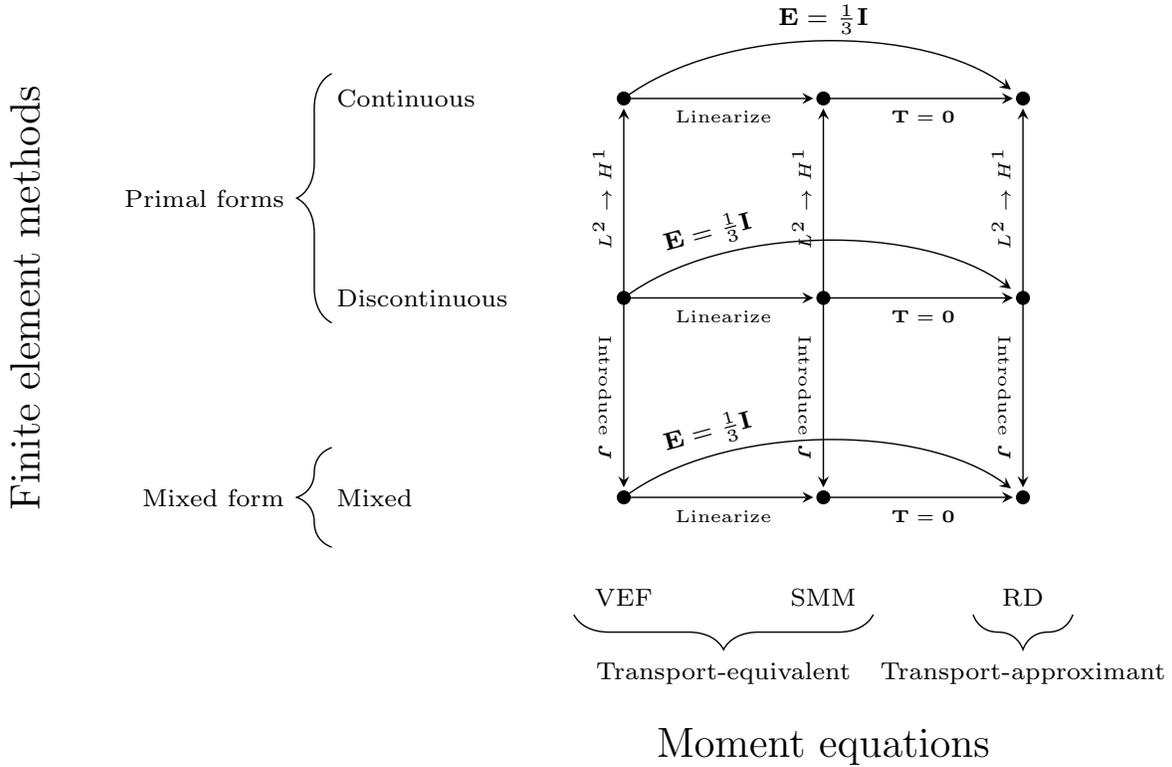


Figure 2.1: Olivier discretization matrix [7].

Advantages of SMM over VEF

The VEF equation is nonlinear. Thus, VEF is a nonlinear reformulation of the linear transport equation. Solving nonlinear equations often requires more computation and/or more storage than solving linear equations, although this is not always true. For example, SI without DSA (or any other acceleration) requires much more computation than VEF in the TDL. Newton's method for solving nonlinear equations converges rapidly, provided that the objective function satisfies certain conditions, but Newton's method may be inconvenient due to the challenges or impracticalities associated with computing and storing the gradient. Olivier used Anderson-accelerated fixed-point iteration, which may be impractical if storing multiple copies of the solution exceeds the computer memory capacity. Furthermore, the linear system that arises is non-symmetric due to the presence of the Eddington tensor inside the divergence. This requires more expensive iterative linear solvers and their associated preconditioners than symmetric and symmetric positive definite (SPD) systems. By definition, a matrix A is symmetric if $A^\top = A$, where A^\top is the transpose of A , which means that the elements of A satisfy $a_{ij} = a_{ji}$ for all i and j . A real matrix A is symmetric positive definite if and only if $A = A^\top$ and $x^\top Ax > 0$ for all vectors $x \neq 0$. I define preconditioning, which

improves the convergence of iterative linear solvers, in section 2.2.9 Linear Solvers and Preconditioners. Linear solvers and their associated preconditioners can leverage the structure of symmetric and SPD coefficient matrices to minimize the number of operations required to solve the corresponding linear system. However, these efficiencies are not applicable to VEF, as the coefficient matrix for the VEF system lacks such advantageous structure. Finally, the left-hand-side of VEF is different for every cycle of the VEF iteration, which means that the coefficient matrix must be recomputed and preconditioned every cycle. The right-hand side of VEF is the same for every cycle.

The SMM equation is linear. Thus, solving it requires solving a linear system of equations rather than a nonlinear system. The operator on the left-hand-side is SPD, which means that the SMM system can be solved using conjugate gradient (CG) with Algebraic Multigrid (AMG) preconditioning. These are efficient iterative methods for preconditioning and solving linear systems. Multigrid methods leverage the fact that smoothers like Jacobi and Gauss-Seidel attenuate high-frequency noise in the approximation of the solution to the linear system more quickly than low-frequency noise, and that coarsening the system converts low frequencies to high frequencies. Multigrid coarsening and refinement requires a mesh concept, whereas Algebraic Multigrid is a generalization of Multigrid which alleviates this requirement. Conjugate Gradient is a method which improves on gradient descent by using conjugate gradients, as can be seen by comparing the circuitous path in Figure 8 to the direct path in Figure 30 of [41].

The left-hand-side operator in SMM is identical to the operator that appears on the left-hand-side of the radiation diffusion approximation. This similarity allows the numerical development efforts aimed at improving the performance and robustness of iterative linear solvers for radiation diffusion to be directly leveraged in SMM. By simply employing the same solvers used for the radiation diffusion system, the SMM system can also be efficiently solved. Finally, the left-hand-side of SMM is fixed for the entire iteration, which means that the coefficient matrix can be computed and preconditioned during the first cycle of the SMM iteration and re-used for all subsequent cycles. The right-hand side of SMM is different for every cycle. Forming a new right-hand side every iteration is less expensive than forming a new coefficient matrix every iteration.

Finally, a Fourier stability analysis of VEF and SMM showed that the two have similar stability and convergence properties [42]. If VEF is theoretically equivalent to SMM, and the latter is less expensive than the former, then it makes practical sense to choose SMM over VEF for use in a hybrid method. Thus, I decide between the left and center columns in Fig. 2.1 by choosing the center column.

Advantages of Mixed over Discontinuous and Continuous FEM

Olivier's discontinuous and continuous discretizations use the second order form of the SMM equation. The correction tensor \mathbf{T} is twice differentiated in the second order form (see Eq. (1.47)). This amplifies the Monte Carlo noise in \mathbf{T} , which is a problem that Olivier never had to contend with because Olivier used a deterministic method (S_N) to compute

\mathbf{T} , whereas my hybrid method uses Monte Carlo to compute \mathbf{T} . The correction tensor \mathbf{T} is once differentiated in the first order form (see Eq. (1.43)). This would appear to amplify the Monte Carlo noise in \mathbf{T} , but using the mixed FEM to discretize the system offloads the derivative from \mathbf{T} to the FEM test function, thus avoiding noise amplification. For this reason, I decide between the three rows of the center column in Fig. 2.1 by choosing the bottom row.

2.2.2 Summary of the Finite Element Method

My novel hybrid method uses a finite element discretization. Here, I outline the concept of finite elements in just a few sentences, then I present details associated with “mixed” finite element formulations. Mixed is a special class of finite element methods to which my hybrid method belongs. Mixed problems require simultaneously solving for two quantities which belong to different finite element spaces. A mixed FEM often results in the need to solve a “saddle point” system. In a saddle point system, the extremum satisfying the discretized system is neither a maximum nor a minimum but rather a saddle point.

The finite element method is a popular spatial discretization choice for numerical solutions of partial differential equations in many scientific fields. The FEM tessellates space using polyhedra that may share faces, edges, and vertices. The domain is the union of the polyhedra in the tessellation,

$$\mathcal{D} = \bigcup_{K \in \mathcal{T}} K, \quad (2.1)$$

where \mathcal{T} denotes the tessellation and K denotes a single polyhedron, which is called a “mesh element” or just “element”. The tessellation is called a “computational mesh” or just “mesh”. The mesh is a purely-geometric entity described by a set of vertices and a set of edges connecting the vertices. The elements that constitute the mesh in discretizations of a single spatial dimension (“1D”) are line segments. Examples of 2D mesh elements are quadrilaterals and triangles. Hexahedra and tetrahedra are examples of 3D mesh elements. A mesh can contain multiple element types, though it is much more common for a mesh to contain only one type of element.

The spatial dependence of the FEM solution to the partial differential equation (PDE) is represented as piecewise polynomials which have domains of definition restricted to individual elements. If a basis is chosen for the polynomial space (e.g. Lagrange interpolating polynomials), then the unknowns on a mesh element are the coefficients of the polynomial representation with respect to the chosen basis. The coefficients are called “degrees-of-freedom”. Examples of basis functions are the Lagrange interpolating polynomials constructed using Gauss-Legendre or Gauss-Lobatto points.

The FEM approximates the spatial dependence of the solution to an integral form of the PDE called the “weak form”. The differential form used to derive the weak form is called the “strong form”. Integrating the weak form produces a large, sparse system of linear equations for the degrees-of-freedom. A sparse matrix is a matrix in which most of

its elements are zero. Matrix-free methods of integrating the weak form, which provide the action of the linear operator without actually forming the coefficient matrix, can be combined with efficient preconditioners and linear solvers to efficiently invert the linear system¹. Methods for representing the linear system, ordered from most scalable to least scalable, include matrix-free, element assembly, partial assembly, and full assembly. Full assembly is also called “global assembly”. Example steps in a typical FEM include:

- i. Choose a mesh, for example a quadrilateral mesh
- ii. Choose basis functions, for example Gauss-Legendre basis functions
- iii. Compute the coefficient matrix by integrating the left-hand side of the weak form
- iv. Compute the right-hand side vector by integrating the right-hand side of the weak form
- v. Solve the linear system, for example using an iterative linear solver with a preconditioner

The weak form in iii. and iv. consists of an equation of the form,

$$L(u, v) = b(v), \quad (2.2)$$

where L is a bilinear form and b is a linear form. I describe the functions u and v , which are called the “trial” and “test” functions, respectively, in section 2.2.5 Finite Element Spaces for the Mixed Problem. Bilinearity means that,

$$L(u_1 + u_2, v) = L(u_1, v) + L(u_2, v), \quad (2.3)$$

and

$$L(u, v_1 + v_2) = L(u, v_1) + L(u, v_2). \quad (2.4)$$

Thus, a bilinear form L is separately linear in each of its arguments. A linear form $b(v)$ satisfies,

$$b(v_1 + v_2) = b(v_1) + b(v_2). \quad (2.5)$$

After choosing a basis for the finite element space, the FEM solution u can be expanded in this basis as,

$$u = \sum_{m=1}^N c_m * u_m, \quad (2.6)$$

where the coefficients c_m are determined from solving a matrix equation with matrix entries $A_{m,n} = L(u_m, u_n)$ and right-hand-side $b(v_m)$.

Boundary conditions (BCs) can be enforced strongly or weakly. Strong enforcement usually means modifying the linear system so that the numerical solution satisfies the boundary condition exactly. For a closed basis, like Gauss-Legendre, this can be achieved by modifying

¹Source iteration, which I described in section 1.3.3, could also be considered a matrix-free method because it inverts a linear system without forming the coefficient matrix.

the representation of the linear system to force the degrees-of-freedom on the boundary to equal the values specified in the boundary condition. This is often only possible for Dirichlet boundary conditions. The Dirichlet BC is called an “essential” BC because it appears in the definition of the finite element space (see section 2.2.5 for examples of finite element spaces).

Weak enforcement means substituting the boundary condition into the weak form, which causes the numerical solution to satisfy the boundary condition only approximately, though the approximation improves under mesh refinement and by increasing the basis function polynomial order. The former is called h -refinement and the latter p -refinement because h and p are often used to represent the characteristic mesh element width and the exponent in the $O(h^{p+1})$ FEM convergence order, respectively. Weak enforcement is common for Neumann BCs. The Neumann BC is called a “natural” BC because the opportunity for substitution of Neumann BCs into the weak form appears in the integration-by-parts formulae which are commonly applied to the weak form. Robin BCs, which are a linear combination of Dirichlet and Neumann BCs, can be enforced weakly.

2.2.3 Finite Element Integration

Finite element integration is performed on a reference element using a change of variables called the inverse mesh transformation $\mathbf{T}^{-1} : K \rightarrow \hat{K}$, where $K \in \mathcal{T}$ and \hat{K} is the reference element. Fig. 2.2 depicts the transformation where K is a quadrilateral element defined in physical-space coordinates $\mathbf{x} = (x, y)$, and \mathbf{T}^{-1} maps to a reference element \hat{K} defined in reference-space coordinates $\xi = \xi, \eta$. Some authors choose to center the reference element at the origin $\xi = (0, 0)$ instead of $\xi = (0.5, 0.5)$, as in Fig. 2.2. The linear system that results from the integration will be the same regardless of the centering, even though \mathbf{T}^{-1} will differ. The reference element could be the unit square for 2D problems or the unit cube for 3D problems, in which case \mathcal{T} would be composed of quadrilaterals or hexahedra, respectively. Then $\mathbf{T}(\boldsymbol{\xi})$ maps points in reference space on the unit square or cube to points in physical space on a quadrilateral or hexahedral element, respectively. The transformation $\mathbf{T}^{-1}(\mathbf{x})$ maps points in physical space to points in reference space. Finite element methods use $\mathbf{T}^{-1}(\mathbf{x})$ to evaluate problem data, originally defined in physical space, within the reference space. The determinant of the Jacobian matrix of the transformation appears in the integrand of the reference-space integrals. The Jacobian matrix is,

$$\mathbf{F} = \frac{\partial \mathbf{T}}{\partial \boldsymbol{\xi}} \in \mathbb{R}^{\dim \times \dim}, \quad (2.7)$$

and $J = |\mathbf{F}|$ is the Jacobian determinant. The Jacobian matrix in 2D is,

$$\mathbf{F} = \begin{pmatrix} \frac{\partial x_1}{\partial \xi_1} & \frac{\partial x_1}{\partial \xi_2} \\ \frac{\partial x_2}{\partial \xi_1} & \frac{\partial x_2}{\partial \xi_2} \end{pmatrix}. \quad (2.8)$$

Reference-space integration is performed with a quadrature rule which evaluates the integrand at a finite number of quadrature points. Thus, the accuracy (and expense) of the finite element method is determined partly by the type and resolution of the mesh, partly by the type and order of the basis functions, and partly by the type and order of the quadrature rule used to integrate the weak form. It is important for the number of quadrature points used in the quadrature rule to be commensurate with the order of the basis function polynomials. A common choice for the quadrature rule order is $p + 1$, where p is the basis function polynomial order.

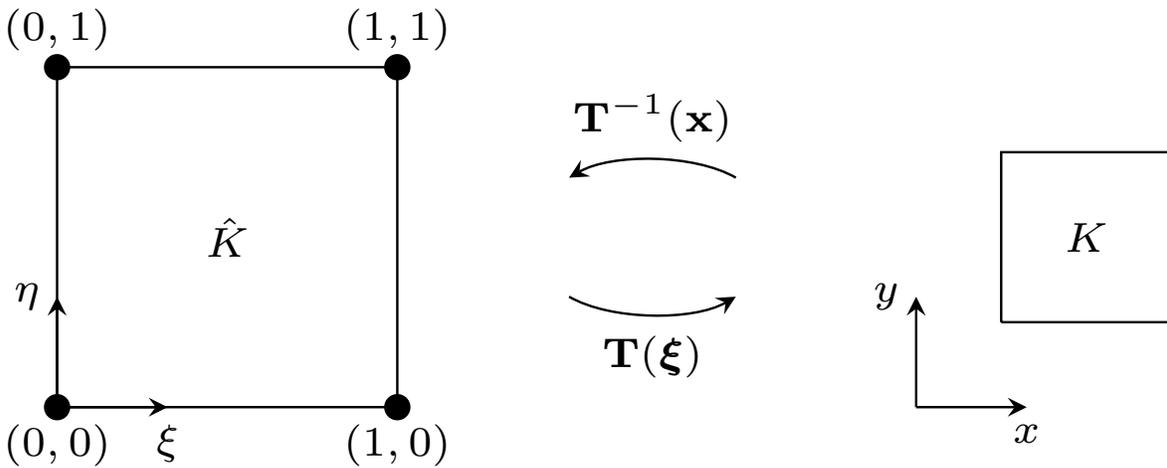


Figure 2.2: The inverse mesh transformation \mathbf{T}^{-1} maps points on the physical-space element K to points on the reference-space element \hat{K} [7].

2.2.4 Sobolev Spaces

A finite element space, which I will define in section 2.2.5, is the space in which one seeks the finite element solution. It is a subspace of a Sobolev space. Note that the first Sobolev space below, $H^1(\mathcal{D})$, does not have relevance for the mixed FEM discretization of SMM unless the SMM system is expressed in a single spatial dimension, because there is no distinction between $H(\text{div}; \mathcal{D})$ and $H^1(\mathcal{D})$ in 1D. More discussion of dimensionality considerations may be found in section 4.4.1.

Denote the space of measurable functions with square-integrable weak gradient as,

$$H^1(\mathcal{D}) = \{u \in L^2(\mathcal{D}) : \int_{\mathcal{D}} \nabla u \cdot \nabla u \, d\mathbf{x} < \infty\}, \quad (2.9)$$

the space of square-integrable functions as,

$$L^2(\mathcal{D}) = \{u : \mathcal{D} \rightarrow \mathbb{R} : \int_{\mathcal{D}} u^2 \, d\mathbf{x} < \infty\}, \quad (2.10)$$

and the space of vector-valued functions with square-integrable divergence as,

$$H(\operatorname{div}; \mathcal{D}) = \{\mathbf{v} \in [L^2(\mathcal{D})]^{\dim} : \nabla \cdot \mathbf{v} \in L^2(\mathcal{D})\}. \quad (2.11)$$

Examination of the weak form guides one to an appropriate choice of Sobolev space. For example, if the weak form includes integrands with gradients, then one would choose $H^1(\mathcal{D})$ because it assures that the integrals in this weak form are computable.

2.2.5 Finite Element Spaces for the Mixed Problem

The order of a PDE is the order of its highest derivative. Any second-order scalar PDE can be converted to a first-order system of two equations by introducing an auxiliary variable. For example,

$$-\nabla \cdot \nabla u = f \quad \mathbf{x} \in \mathcal{D}, \quad (2.12a)$$

$$u = 0 \quad \mathbf{x} \in \partial\mathcal{D}, \quad (2.12b)$$

can be converted by introducing a new unknown v ,

$$\mathbf{v} = \nabla u \quad \mathbf{x} \in \mathcal{D}, \quad (2.13a)$$

$$-\nabla \cdot \mathbf{v} = f \quad \mathbf{x} \in \mathcal{D}, \quad (2.13b)$$

$$u = 0 \quad \mathbf{x} \in \partial\mathcal{D}. \quad (2.13c)$$

Eq. (2.12a) is called the “primal formulation”, and the system of Eqs. (2.13a) and (2.13b) is called the “flux formulation”. The finite element method used to solve the system of Eqs. (2.13a) and (2.13b) for u and \mathbf{v} is referred to as a “mixed” FEM when two different finite element spaces are chosen for u and \mathbf{v} . Mixed finite element methods used in the flux formulation are somewhat more specialized compared to, for instance, continuous or discontinuous finite element methods used in the primal formulation. As a result, they are less commonly used. However, they are covered in textbooks such as Quarteroni and Valli [43] and, in greater detail, Boffi *et al.* [44].

The flux formulation has advantages and disadvantages compared to the primal formulation. The linear system for the flux formulation will be larger, and therefore more expensive to store and invert, because it includes unknowns for both u and \mathbf{v} . However, the flux formulation is first-order, whereas the primal formulation is second-order. This is crucial for my hybrid method, as I aim to avoid differentiation due to its tendency to amplify noise. The SMM data, which I compute using Monte Carlo, inevitably contain statistical noise.

Derivation of the FEM weak form typically involves multiplying by a “test function”, then integrating over an element, then taking the sum over all elements. The function used to approximate the solution is called the “trial function”. Petrov-Galerkin approximations use different basis functions for the test and trial functions, whereas Bubnov-Galerkin approximations use the same basis functions. I use a Bubnov-Galerkin approximation. In the mixed FEM in my hybrid method, the finite element space of my scalar trial and test

functions is the Discrete Galerkin (DG) finite element space, and that of my non-scalar trial and test functions is the Raviart Thomas (RT) finite element space.

The DG space is a discrete subspace of the space of square-integrable functions, $L^2(\mathcal{D})$. Square-integrable means that,

$$\int u^2 \, d\mathbf{x} < \infty. \quad (2.14)$$

Adopting the notation from Olivier [7], the scalar trial function and its associated test function on each element belong to the space of mapped polynomials defined by composition with the inverse mesh transformation,

$$\mathbb{Q}_p(K) = \{u = \hat{u} \circ \mathbf{T}^{-1} : \hat{u} \in \mathcal{Q}_p(\hat{K})\}, \quad (2.15)$$

where $\mathcal{Q}_p(\hat{K})$ denotes the tensor product polynomial space of equal degree in each variable,

$$\mathcal{Q}_p(\hat{K}) = \begin{cases} \mathcal{Q}_{p,p}(\hat{K}), & \dim = 2 \\ \mathcal{Q}_{p,p,p}(\hat{K}), & \dim = 3 \end{cases}. \quad (2.16)$$

The multi-dimensional polynomial spaces $\mathcal{Q}_{m,n}(\hat{K})$ and $\mathcal{Q}_{\ell,m,n}(\hat{K})$ are simply tensor products of univariate polynomial spaces,

$$\mathcal{Q}_{m,n}(\hat{K}^2) = \{p(x)q(y) : p \in \mathcal{P}_m(\hat{K}^1), q \in \mathcal{P}_n(\hat{K}^1)\}, \quad (2.17a)$$

$$\mathcal{Q}_{\ell,m,n}(\hat{K}^3) = \{p(x)q(y)r(z) : p \in \mathcal{P}_\ell(\hat{K}^1), q \in \mathcal{P}_m(\hat{K}^1), r \in \mathcal{P}_n(\hat{K}^1)\}, \quad (2.17b)$$

where \mathcal{P}_k is the univariate polynomial space,

$$\mathcal{P}_k(\hat{K}^1) = \{p : \hat{K}^1 \rightarrow \mathbb{R} : p = \sum_{i=0}^k \alpha_i x^i, \alpha_i \in \mathbb{R}\} = \text{span}\{1, x, x^2, \dots, x^k\}. \quad (2.18)$$

The degree- p DG space is,

$$Y_p = \{u \in L^2(\mathcal{D}) : u|_K \in \mathbb{Q}_p(K), \quad \forall K \in \mathcal{T}\}. \quad (2.19)$$

The degrees-of-freedom in the DG space are not shared by elements, which allows one to use either open or closed points to define the basis functions. This contrasts with the more common Continuous Galerkin (CG) space, which has shared degrees-of-freedom, and therefore is restricted to closed points.

The RT space is the eponymous creation of P. Raviart and J. Thomas [45, 46]. It is a discrete subspace of the space of vector-valued functions with square-integrable divergence, $H(\text{div}; \mathcal{D})$, where

$$H(\text{div}; \mathcal{D}) = \{\mathbf{v} \in [L^2(\mathcal{D})]^2 : \nabla \cdot \mathbf{v} \in L^2(\mathcal{D})\}. \quad (2.20)$$

Vectors in $H(\text{div}; \mathcal{D})$ have normal components which are continuous across element faces, but there is no continuity requirement for the tangential component, which means that the

degrees-of-freedom in the normal direction must be at closed points. Once again adopting the notation from Olivier [7], the 2D RT space is the local polynomial space $\mathcal{Q}_{p+1,p}(\hat{K}) \times \mathcal{Q}_{p,p+1}(\hat{K})$, and the 3D RT space is $\mathcal{Q}_{p+1,p,p}(\hat{K}) \times \mathcal{Q}_{p,p+1,p}(\hat{K}) \times \mathcal{Q}_{p,p,p+1}(\hat{K})$. The 1D RT space is a discrete subspace of $H^1(\mathcal{D})$. That is, there is no distinction between $H(\text{div}; \mathcal{D})$ and $H^1(\mathcal{D})$ in 1D. The degree- p RT space is,

$$RT_p = \{\mathbf{v} \in [L^2(\mathcal{D})]^{\text{dim}} : \mathbf{v}|_K \in \mathbb{D}_p(K) \forall K \in \mathcal{T} \text{ and } [\mathbf{v} \cdot \mathbf{n}] = 0 \forall \mathcal{F} \in \Gamma_0\}. \quad (2.21)$$

For more information about the DG and RT spaces, see the text from which I took all of my finite element language and notation used in this dissertation, which is Chapter 4 “Finite Element Preliminaries” in Olivier [7].

2.2.6 Derivation of the Weak Form

The finite element method weak form for the mixed problem is: find $(\varphi, \mathbf{J}) \in Y_p \times RT_p$ such that,

$$\int u \nabla \cdot \mathbf{J} \, \text{d}\mathbf{x} + \int \sigma_a u \varphi \, \text{d}\mathbf{x} = \int u Q_0 \, \text{d}\mathbf{x}, \quad \forall u \in Y_p, \quad (2.22a)$$

$$\begin{aligned} -\frac{1}{3} \int \nabla \cdot \mathbf{v} \varphi \, \text{d}\mathbf{x} + \int \sigma_t \mathbf{v} \cdot \mathbf{J} \, \text{d}\mathbf{x} + \frac{2}{3} \int_{\Gamma_b} (\mathbf{v} \cdot \mathbf{n})(\mathbf{J} \cdot \mathbf{n}) \, \text{d}s &= \int \mathbf{v} \cdot \mathbf{Q}_1 \, \text{d}\mathbf{x} - \int_{\Gamma_b} \mathbf{v} \cdot \mathbf{T}\mathbf{n} \, \text{d}s \\ + \frac{2}{3} \int_{\Gamma_b} (\mathbf{v} \cdot \mathbf{n})(2J_{\text{in}} + \beta) \, \text{d}s - \int_{\Gamma_0} \llbracket v \rrbracket \cdot \{\{\mathbf{T}\mathbf{n}\}\} \, \text{d}s + \int \nabla_h \mathbf{v} : \mathbf{T} \, \text{d}\mathbf{x} \quad \forall \mathbf{v} \in RT_p, \end{aligned} \quad (2.22b)$$

where the “broken” gradient ∇_h , the jump operator $\llbracket \cdot \rrbracket$, and the average operator $\{\{ \cdot \}\}$ are,

$$(\nabla_h u)|_K = \nabla(u|_K) \quad \forall K \in \mathcal{T}, \quad (2.23)$$

$$\llbracket u \rrbracket = \begin{cases} u_1 - u_2 & \mathcal{F} \in \Gamma_0 \\ u & \mathcal{F} \in \Gamma_b \end{cases}, \quad (2.24)$$

$$\{\{ u \}\} = \begin{cases} \frac{u_1 + u_2}{2} & \mathcal{F} \in \Gamma_0 \\ u & \mathcal{F} \in \Gamma_b \end{cases}. \quad (2.25)$$

In Eq. (2.23), $u|_K$ denotes the restriction of u to the element K . In Eqs. (2.24) and (2.25), Γ_0 is the set of unique faces internal to the mesh, and Γ_b is the set of unique faces on the boundary of the mesh.

To derive Eq. (2.22a), multiply Eq. (1.20a) by the test function $u \in Y_p$, then integrate over \mathcal{D} . Deriving Eq. (2.22b) requires several steps and employs integration-by-parts rules created using the following vector calculus identities:

- Product rule for divergence of a scalar (a) times a vector (\mathbf{F}):

$$\nabla \cdot (a\mathbf{F}) = \nabla a \cdot \mathbf{F} + a\nabla \cdot \mathbf{F}, \quad (2.26)$$

- Divergence theorem:

$$\int_K \nabla \cdot \mathbf{F} \, d\mathbf{x} = \int_{\partial K} \mathbf{F} \cdot \mathbf{n} \, ds, \quad (2.27)$$

- Product rule for divergence of a vector (\mathbf{v}) dotted with a tensor (\mathbf{T}):

$$\nabla \cdot (\mathbf{v} \cdot \mathbf{T}) = \mathbf{v} \cdot (\nabla \cdot \mathbf{T}) + \mathbf{T} : \nabla \mathbf{v}, \quad (2.28)$$

- Double dot product involving vectors (\mathbf{v} , \mathbf{n}) and a tensor (\mathbf{T}):

$$(\mathbf{v} \cdot \mathbf{T}) \cdot \mathbf{n} = \mathbf{v} \cdot (\mathbf{T}\mathbf{n}). \quad (2.29)$$

Combining Eq. (2.26) with Eq. (2.27) gives an integration-by-parts rule that offloads a derivative from a scalar trial function to a vector test function and produces a surface integral as a side effect,

$$\int \nabla a \cdot \mathbf{F} \, d\mathbf{x} = - \int a \nabla \cdot \mathbf{F} \, d\mathbf{x} - \int_{\partial K} a (\mathbf{F} \cdot \mathbf{n}) \, ds. \quad (2.30)$$

Combining Eq. (2.28) with Eq. (2.27) gives an integration-by-parts rule that offloads a derivative from a tensor function to a vector test function and produces a surface integral as a side effect. Subsequent use of Eq. (2.29) in the integrand of the surface integral gives,

$$\int \mathbf{v} \cdot \nabla \cdot \mathbf{T} \, d\mathbf{x} = - \int \mathbf{T} : \nabla \mathbf{v} \, d\mathbf{x} - \int_{\partial K} \mathbf{v} \cdot \mathbf{T}\mathbf{n} \, ds. \quad (2.31)$$

The three steps for deriving Eq. (2.22b) are as follows:

Step 1) Integration over element K

Taking the inner product of both sides of Eq. (1.43) with a test function $\mathbf{v} \in RT_p$ on an arbitrary element K and applying the integration-by-parts rules in Eqs. (2.30) and (2.31) gives,

$$\begin{aligned} & - \frac{1}{3} \int_K \nabla \cdot \mathbf{v} \, \varphi \, d\mathbf{x} + \frac{1}{3} \int_{\partial K} \varphi (\mathbf{v} \cdot \mathbf{n}) \, ds + \int_K \sigma_t \mathbf{v} \cdot \mathbf{J} \, d\mathbf{x} \\ & = \int_K \mathbf{v} \cdot \mathbf{Q}_1 \, d\mathbf{x} - \int_{\partial K} \mathbf{v} \cdot \mathbf{T}\mathbf{n} \, ds + \int_K \nabla \mathbf{v} : \mathbf{T} \, d\mathbf{x} \quad \forall \mathbf{v} \in RT_p. \end{aligned} \quad (2.32)$$

Step 2) Sum over all elements

Summing Eq. (2.32) over all elements $K \in \mathcal{T}$ gives,

$$\begin{aligned} & - \frac{1}{3} \int \nabla \cdot \mathbf{v} \, \varphi \, d\mathbf{x} + \frac{1}{3} \int_{\partial \mathcal{D}} \varphi (\mathbf{v} \cdot \mathbf{n}) \, ds + \int \sigma_t \mathbf{v} \cdot \mathbf{J} \, d\mathbf{x} \\ & = \int \mathbf{v} \cdot \mathbf{Q}_1 \, d\mathbf{x} - \int_{\partial \mathcal{D}} \mathbf{v} \cdot \mathbf{T}\mathbf{n} \, ds - \int_{\Gamma_0} \llbracket v \rrbracket \cdot \{ \mathbf{T}\mathbf{n} \} \, ds + \int \nabla_h \mathbf{v} : \mathbf{T} \, d\mathbf{x} \quad \forall \mathbf{v} \in RT_p. \end{aligned} \quad (2.33)$$

Step 3) Enforce boundary condition

Solving Eq. (1.46) for φ and substituting into the second term in Eq. (2.33) gives the final result, Eq. (2.22b).

2.2.7 Convergence of the Discretization

If the solution to the continuous SMM equations is sufficiently smooth and the mesh is relatively undistorted, then, according to the Brambell-Hilbert Lemma, the chosen finite element spaces Y_p and RT_p will yield a solution of the discretized SMM equations with a numerical error of $O(h^{p+1})$ [47]. In this expression, p is the order of the basis function polynomials and h is the characteristic mesh length,

$$h = \max_{K \in \mathcal{T}} \left(\int_{\hat{K}} J d\xi \right)^{1/\dim}. \quad (2.34)$$

In Eq. (2.34), $J = \det(\mathbf{F})$ is the Jacobian determinant of the element transformation, and \mathbf{F} is the Jacobian matrix of the mapping from reference space to physical space $\mathbf{x} = \mathbf{T}(\xi)$,

$$\mathbf{F} = \frac{\partial \mathbf{T}}{\partial \xi} \in \mathbb{R}^{\dim \times \dim}. \quad (2.35)$$

If the conditions on solution smoothness or mesh distortion are not satisfied, then the Bramble-Hilbert Lemma is inapplicable, and the asymptotic order of the numerical error in the solution could exceed the hypothesized value of $O(h^{p+1})$.

2.2.8 Mixed Finite Element Method Algorithm

Algorithm 1 shows the definition of a function which implements the mixed finite element method solver of the second moment system described in this chapter. Algorithm 1 defines the logic inside the `sm()` call in the HSM algorithm that I will present in Algorithm 3. The most important variable elided from Algorithm 1 is the mesh. The mesh would include a description of the element geometry, material properties, and topology. All are required for integrating the bilinear form, integrating the linear form, and representing the solution.

Line 1 shows the function name and function parameters, all of which appear as source terms on the right-hand side of the weak form in Eqs. (2.22a) and (2.22b). Line 2 assembles the bilinear form, which involves computing the integrals on the left-hand side of the weak form, ultimately leading to the discrete operator. Section 2.2.2 mentions some choices for representing the discrete operator. For example, in the case of full assembly, the variable “A” in Line 2 would represent a coefficient matrix. In contrast, in a matrix-free method, “A” would denote an entity that describes the action of the corresponding linear operator. For matrix-free methods, Line 2 would compute the action of the bilinear form instead of assemble it. Line 3 assembles the linear form, which requires computing the integrals on the right-hand side of the weak form.

Line 4 solves the linear system $A\varphi = b$ in a manner described in section 2.2.9 Linear Solvers and Preconditioners. The solution of the linear system is a vector containing all of the \mathbf{J} degrees-of-freedom stacked on top of all of the φ degrees-of-freedom. The function referenced in Line 4 returns only the φ degrees-of-freedom, thus the \mathbf{J} degrees-of-freedom are discarded. Line 5 returns the solution.

Assembly of the bilinear form in Line 2 requires a loop over elements. For each element, the local “stiffness matrix” is computed by integrating the bilinear form over the element, and then the local stiffness matrix is mapped to the global stiffness matrix using the mesh connectivity. Assembly of the linear form in Line 3 also requires a loop over elements. For each element, the local “load vector” is computed by integrating the source terms multiplied by the basis functions over the element, and then the local load vector is mapped to the global load vector using the mesh connectivity. Finite element methods were first developed and applied to problems in structural mechanics, which gave rise to the “stiffness” and “load” adjectives for describing the coefficient matrix and right-hand-side vector.

Algorithm 1 Mixed Finite Element Method Solve of Second Moment System

```

1: function SM( $Q_0, \mathbf{Q}_1, \mathbf{T}, \beta$ )
2:    $A \leftarrow$  assemble_bilinear_form()
3:    $b \leftarrow$  assemble_linear_form( $Q_0, \mathbf{Q}_1, \mathbf{T}, \beta$ )
4:    $\varphi \leftarrow$  linear_solve( $A, b$ )
5:   return  $\varphi$ 
6: end function

```

2.2.9 Linear Solvers and Preconditioners

Consider the linear system $Ax = b$. Let $r^{(i)} = b - Ax^{(i)}$ denote the residual after i cycles of the iterative linear solver, and let $\|\cdot\|_2$ be the 2-norm of some vector. The 2-norm for a vector $r \in \mathbb{R}^N$ is,

$$\|r\|_2 = \sqrt{\sum_{j=1}^N |r_j|^2}, \quad (2.36)$$

where r_j is the j th entry in the vector r . The mixed FEM SMM system of Eqs. (2.22a) and (2.22b) is not symmetric. It can be symmetrized by multiplying Eq. (2.22a) by negative unity and Eq. (2.22b) by positive three. This allows the solution to the linear system to be computed using the Minimal Residual Method (MINRES) [48], whereas the unsymmetrized system must be solved at greater expense using Generalized Minimal Residual Method (GMRES) [49]. MINRES chooses iterates $x^{(i)}$ which minimize $\|r^{(i)}\|_2$ [50].

The mixed FEM SMM system permits hybridization, which replaces the block system with a smaller system for the associated Lagrange multipliers. The resulting system has fewer unknowns and is also SPD, which means that it can be solved using the previously-mentioned CG method [51, 52], and preconditioned using the previously-mentioned AMG

method [53]. If A is SPD then it defines a norm $\|r\|_{A^{-1}} = (r^T A^{-1} r)^{1/2}$, and CG chooses iterates $x^{(i)}$ which minimize $\|r^{(i)}\|_{A^{-1}}$ [50]. More details on AMG can be found in books such as McCormick [54] and Briggs *et al.* [55]. The hybridized system, provided in [7] and copied here, is: find $(\mathbf{J}, \varphi, \lambda) \in \hat{RT}_p \times Y_p \times \Lambda_p$ such that,

$$\int u \nabla_h \cdot \mathbf{J} \, d\mathbf{x} + \int \sigma_a u \varphi \, d\mathbf{x} = \int u Q_0 \, d\mathbf{x}, \quad \forall u \in Y_p, \quad (2.37a)$$

$$-\frac{1}{3} \int \nabla_h \cdot \mathbf{v} \varphi \, d\mathbf{x} + \int \sigma_t \mathbf{v} \cdot \mathbf{J} \, d\mathbf{x} + \frac{2}{3} \int_{\Gamma_b} (\mathbf{v} \cdot \mathbf{n})(\mathbf{J} \cdot \mathbf{n}) \, ds + \int_{\Gamma_0} \llbracket \mathbf{v} \cdot \mathbf{n} \rrbracket \lambda \, ds = \mathcal{S}, \quad \forall \mathbf{v} \in \hat{RT}_p, \quad (2.37b)$$

$$\int_{\Gamma_0} \mu \llbracket \mathbf{J} \cdot \mathbf{n} \rrbracket \, ds = 0, \quad \forall \mu \in \Lambda_p, \quad (2.37c)$$

where

$$\mathcal{S} = \int \mathbf{v} \cdot \mathbf{Q}_1 \, d\mathbf{x} - \int_{\Gamma_b} \mathbf{v} \cdot \mathbf{Tn} \, ds + \frac{2}{3} \int_{\Gamma_b} \mathbf{v} \cdot \mathbf{n} (2J_{\text{in}} + \beta) \, ds - \int_{\Gamma_0} \llbracket \mathbf{v} \rrbracket \cdot \{\{\mathbf{Tn}\}\} \, ds + \int \nabla_h \mathbf{v} : \mathbf{T} \, d\mathbf{x} \quad (2.38)$$

is the source term for the hybridized discretization and λ are the Lagrange multipliers.

Preconditioning improves the convergence of iterative linear solvers. Left preconditioning means replacing $Ax = b$ with,

$$M^{-1}Ax = M^{-1}b, \quad (2.39)$$

where M approximates A , its inverse M^{-1} is easy to form and easy to apply, and $M^{-1}A$ has a smaller condition number than A . Right preconditioning means replacing $Ax = b$ with,

$$AM^{-1}y = b, \quad (2.40)$$

where $y = Mx$. Left preconditioning modifies both A and b , whereas right preconditioning modifies A while leaving b unchanged. The choice between left and right preconditioning depends on the properties of the coefficient matrix and the properties of the preconditioner. Since the mixed FEM SMM is a block system, block preconditioners are a natural choice for preconditioning the coefficient matrix because they exploit the block structure of the matrix.

The condition number of a coefficient matrix is a quantity $\kappa(A) \in [1, \infty)$ which determines the convergence behavior of iterative solvers used for solving the associated linear system. The condition number is,

$$\kappa(A) = \frac{\sigma_{\max}}{\sigma_{\min}}, \quad (2.41)$$

where σ_{\max} and σ_{\min} are the largest and smallest singular values of A . The singular values of A are the entries of the diagonal matrix Σ in the singular value decomposition $A = U\Sigma V^T$. It is not uncommon for an iterative linear solver to fail to converge when the coefficient matrix has a high condition number. The condition number of the identity matrix is unity. The condition number of a singular matrix is considered to be infinity.

Fig. 2.3 shows the iterative convergence of various linear solvers applied to two linear systems of equal size. The condition numbers of these systems differ by a factor of approximately 75, as detailed in table 2.1. The linear systems are from a linear continuous Galerkin discretization of Poisson’s equation with Dirichlet boundary conditions on a 40×40 mesh of equal-sized squares [56]. The linear systems have 3239 unknowns and about 0.5% of the entries in the coefficient matrix are nonzero. I used a convergence tolerance of 10^{-7} with solvers from the parallel, sparse, iterative linear solver library known as HYPRE [57]. The plots in Fig. 2.3 show the 2-norm of the residual at iteration cycle i , $\|r^{(i)}\|_2$, where $r^{(i)} = b - Ax^{(i)}$. I ran until convergence or cycle 3300, whichever occurred first. The solvers that I plotted in Fig. 2.3 are:

- cg—Conjugate Gradient (described above).
- bicgstab—Biconjugate Gradient Stabilized [58] is a variation on CG which does not require A to be symmetric
- amg—Algebraic Multigrid (described above).
- gmres—Generalized Minimum Residual (described above).
- lgmres—Loose Generalized Minimum Residual [59] is a variation on GMRES that attempts to accelerate GMRES convergence by disrupting the cyclic pattern of directions of the residual vectors at the end of each restart cycle of restarted GMRES.
- flexgmres—Flexible Generalized Minimum Residual [60] is a variation on GMRES that allows changes in the preconditioner at every step.
- hybrid—HYPRE’s “hybrid” solver assumes a strongly diagonally dominant system, and begins iterating a diagonally scaled Krylov solver without preconditioning. If the convergence rate of the solver falls below a threshold, the algorithm switches to a preconditioned Krylov solver. The solver and preconditioner are arbitrary, but I used HYPRE’s default, which is CG preconditioned using AMG.
- amg cg—Algebraic Multigrid Preconditioned Conjugate Gradient uses HYPRE’s CG solver with HYPRE’s AMG as the preconditioner.

Fig. 2.3 demonstrates that iterative solvers do not always converge, and that convergence may require more iterations for matrices with larger condition numbers, even with preconditioning. The curves in Fig. 2.3 which touch the right edge of the plots correspond to solvers that did not converge. For example, CG converged after cycles 1769 in Fig. 2.3 (a), but had not yet converged after 3300 cycles in (b). A solver that converges in fewer iterations than another may not necessarily be faster, as the cost of an iteration depends on the solver and can vary between different solvers.

The smooth curves in Fig. 2.3 correspond to solvers which minimize the 2-norm of the residual, $\|r^{(i)}\|_2$, which is the norm that I chose to plot. If I had plotted the A^{-1} -norm of the

residual, $\|r\|_{A^{-1}}$, then the CG curve would appear smooth instead of choppy. The reason that the curve for HYPRE’s hybrid solver and HYPRE’s AMG-preconditioned CG solver (“hybrid” and “amg cg” in Fig. 2.3, respectively) appear similar is because the hybrid solver spends some initial iterations running a different solver before detecting the slow convergence rate and switching to AMG-preconditioned CG.

In my HSM implementation, which I describe in Chapter 4, I use the parallel, sparse, direct linear solver library known as SuperLU [61]. In practice, a dense, direct linear solver typically requires $O(n^3)$ operations to solve a system of n linear equations with n unknowns. Sparse direct solvers, such as SuperLU, leverage the sparsity of the system to compute the solution in significantly fewer operations and substantially lower memory usage compared to dense direct solvers. The memory savings are achieved through the use of specialized data structures that store only nonzero elements of the matrix.

Each iteration of an iterative linear solver requires only $O(n)$ operations when applied to sparse systems. However, the number of iterations required to converge cannot generally be determined *a priori* and may be quite large, particularly for systems where effective preconditioners are either very difficult to construct or practically unavailable. In such cases, SuperLU may be more efficient. Another tradeoff is that direct solvers give an exact solution (up to machine precision) in a predetermined number of operations, whereas iterative solvers give an approximate solution, and may not converge.

Assume that we can effectively precondition the sparse linear system that we want to solve, and that the system has a very large number of unknowns. Then the relative cost of the aforementioned algorithms is,

$$\text{AMG+CG} < \text{BPC+MINRES} < \text{BPC+GMRES} < \text{SuperLU}, \quad (2.42)$$

where BPC is a block preconditioner. A final consideration when picking a linear solver is the amount of parallelism in the algorithm and whether the parallel work can be efficiently mapped onto a given computer architecture. For example, the coarsening operation fundamental to multigrid produces progressively smaller linear systems which become too small to amortize graphics processor kernel launch overhead, and hybridization necessitates global assembly, which is less scalable than matrix-free methods. An efficient, matrix-free iterative solver combined with block preconditioning of the saddle-point problem may outperform AMG+CG on the hybridized system, particularly on graphics processors [62].

I observe that the expense of my linear solve, which uses the serial implementation of SuperLU, is orders of magnitude less than the expense of my Monte Carlo solve. This is partly because I run a relatively large number of simulation particles, exceeding 1 billion, and partly because the number of unknowns in my linear system is relatively small, never exceeding 1 million. I would switch to an iterative linear solver, or at least parallel SuperLU, if the linear solve became the bottleneck, which could happen if I increase the size of the linear system or decreased the number of simulation particles. Chapter 4 has more details on my HSM implementation, and Chapter 5 lists the exact number of simulation particles and elements that I used in my calculations.

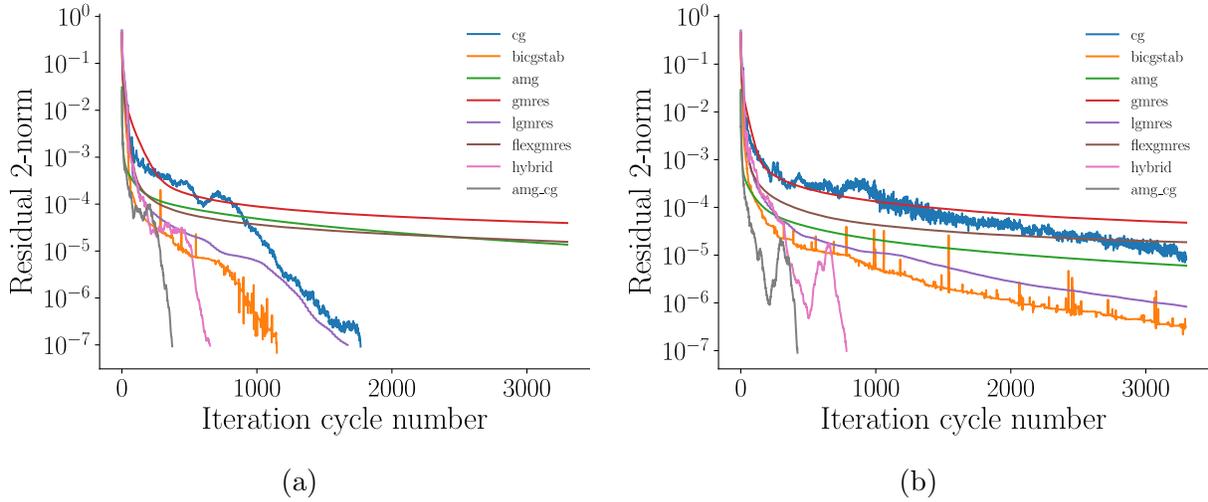


Figure 2.3: Solvers require more iterations to converge when the condition number increases, in this case by a factor of $\kappa_{(b)}/\kappa_{(a)} \approx 75$ [56].

	(a)	(b)	(b) / (a)
σ_{\max}	$6.39 \cdot 10^{11}$	$6.37 \cdot 10^{13}$	99.72
σ_{\min}	$1.05 \cdot 10^5$	$1.38 \cdot 10^5$	1.32
$\kappa(A)$	$6.11 \cdot 10^6$	$4.61 \cdot 10^8$	75.42

Table 2.1: Condition numbers of the linear systems in Fig. 2.3 (a) and (b).

	(a)	(b)
cg	1769	3300
bicgstab	1149	3296
amg	3300	3300
gmres	3300	3300
lgmres	1673	3300
flexgmres	3300	3300
hybrid	654	784
amg cg	374	421

Table 2.2: Iterations required to converge the solvers in Fig. 2.3 (a) and (b). The value 3300 indicates that the solver did not converge to the tolerance of 10^{-7} after 3300 iterations.

2.2.10 Linear System Size

The size of the linear system can be determined using the mesh topology and the order of the basis function polynomials. The trial function for the scalar unknown φ has piecewise polynomial representation. Specifically, we seek the trial function in the degree- p DG space, $\varphi \in Y_p$, meaning that the total number of degrees-of-freedom used to represent φ is simply the number of degrees-of-freedom per element multiplied by the number of elements. Elements do not share Y_p degrees-of-freedom.

The trial function for the vector unknown \mathbf{J} also has piecewise polynomial representation, but we seek the trial function in the degree- p RT space, $\mathbf{J} \in RT_p$. In a quadrilateral mesh of a geometry containing two spatial dimensions, each element has four RT_0 degrees-of-freedom: one on the midpoint of each of the four edges of the quadrilateral. Elements share a subset of their RT_p degrees-of-freedom. If two elements have an edge in common, then the RT_p degrees-of-freedom which correspond to the component of the vector unknown \mathbf{J} that is normal to the common edge are shared.

Chapter 3

Monte Carlo Component of Hybrid Second Moment

The Monte Carlo component of HSM solves the linear transport Eqs. (1.13a) and (1.13b) that I introduced in section 1.3.2. The MC solve in HSM does not have scattering events. Instead, the effect of the scattering source in Eq. (1.13a) is incorporated by solving the SMM system of Eqs. (1.20a), (1.43) and (1.46) that I introduced in section 1.3.5. In Chapter 2, I described the mixed finite element method discretization that I use to solve the SMM system. In this chapter, I show how to use Monte Carlo to estimate the data for the SMM system, and then how to use the solution of the SMM system to incorporate the effect of the scattering source into the Monte Carlo solve.

The most important results in this chapter are the estimators, which are sums, and the procedure for their accumulation. Mathematically, the accumulation procedure is just repeated random variate generation and function evaluation. Random variate generation is divided into “sourcing” and “tracking”, where sourcing randomly selects the initial phase-space points, and tracking randomly moves the points through phase-space. Estimators are the output of the Monte Carlo solve. An estimator is also known as a “tally”, and tallying refers to the accumulation of tallies.

I begin this chapter with two sections in which I introduce mathematics and terminology for describing Monte Carlo simulation. The first is section 3.1 Monte Carlo Simulation of Random Experiments, and the second is section 3.2 Monte Carlo Simulation as Integral Estimation. Random experiments pervade the human experience, and integration is a very common application of Monte Carlo that finds use in many different scientific fields. Both provide insight into Monte Carlo integration of the linear transport Eqs. (1.13a) and (1.13b), which I describe in section 3.3. I describe how I use Monte Carlo to compute the data for the SMM Eqs. (1.20a), (1.43) and (1.46) in section 3.4, and I derive the variance of the transport correction tensor estimator $\hat{\mathbf{T}}$ in section 3.6. The derivation demonstrates an undesirable dependence on the scaling parameter from the TDL regime that I described in section 1.3.6. I conclude this chapter with derivations of lower variance estimators for \mathbf{T} in sections 3.7 and 3.8. An ideal estimator would have a desirable dependence on the TDL

scaling parameter and would be less costly to compute than efficient alternative estimators.

Many definitions, examples, derivations, and much of the notation that I use for probability quantities, such as random variables, probability density functions, expectation, etc. come from an excellent calculus-based introductory probability textbook by L. Leemis [63]. The primary focus is probability, but Leemis includes a secondary focus on Monte Carlo simulation as a way to provide supporting evidence for analytically-derived probabilities and expectations. Leemis does not mention transport. Readers looking for an elaboration of the Monte Carlo transport description that I provide in this chapter may consider the textbook by Lux and Koblinger [64]. Additional references which may also be helpful are the textbook by Spanier and Gelbard [65], the code manual by Kulesza *et al.* [66], and the lecture slides by Brown [21].

3.1 Monte Carlo Simulation of Random Experiments

Monte Carlo simulation is a broad field that resists one definition. One very general definition of MC simulation is: a numerical method that uses random numbers to approximate the probability of one or more possible outcomes in the sample space of a random experiment. The Monte Carlo component of HSM is a Monte Carlo simulation of the system of particles which behave according to the linear transport equation on the left side of Fig. 1.3. In section 3.3 Monte Carlo Integration of Linear Transport, I will show that the Monte Carlo component of HSM is Monte Carlo integration. Thus, it is both a MC simulation and MC integration. While the MC integration formulation is what I will use to derive the Monte Carlo estimators for the SMM data \mathbf{T} and β (which appear on the bottom of Fig. 1.3), I first provide a general description of MC simulation in this section as a way to introduce important terminology.

We can enumerate all possible outcomes of a random experiment, but we cannot predict the outcome, otherwise it would not be random. A random experiment could be flipping two coins, each with the letter H written on one side and T on the other, and observing how many H-side-up coins result. Another random experiment could be directing a laser pulse consisting of two photons at a foil and observing how many of the incident photons pass through the foil without collision.

3.1.1 Discrete Random Variables

Some definitions are required to describe a Monte Carlo simulation of the aforementioned experiments. A random variable is a function X that assigns each element of the sample space $s \in S$ to a real number $X(s) = x$, so $X : S \rightarrow \mathbb{R}$. The support of a random variable is the subset of the range of the random variable which has nonzero probability of being the outcome of an associated random experiment. That is, an element x in the support of X has nonzero probability of occurring as the outcome of the random experiment $P(X = x) \neq 0$, whereas an element y in the range of X may have zero probability of occurring as the

outcome of a random experiment, $P(X = y) = 0$. The support of X is the set of real numbers $\mathcal{A} = \{x | x = X(s), s \in S\}$.

We can denote the sample space for the aforementioned coin flipping experiment as $S = \{TT, HT, TH, HH\}$ and the support $\mathcal{A} = \{0, 1, 2\}$ so that the random variable X is the number of H-side-up coins after flipping,

$$X(TT) = 0, \quad X(HT) = X(TH) = 1, \quad X(HH) = 2. \quad (3.1)$$

The probability of observing an outcome in the set $A \subset \mathcal{A}$ is,

$$P(A) = P(X \in A) = \sum_A f(x), \quad (3.2)$$

where $f(x)$ is the probability mass function (PMF). Assuming that the outcomes in S are equally likely,

$$f(x) = P(X = x) = \frac{\text{number of outcomes in } S \text{ such that } X(s) = x}{\text{total number of outcomes in } S}, \quad (3.3)$$

which gives us,

$$f(x) = \begin{cases} 1/4 & x = 0, \\ 1/2 & x = 1, \\ 1/4 & x = 2. \end{cases} \quad (3.4)$$

The random variable X is discrete because \mathcal{A} is countable. The PMF for the photon transmission experiment would be more complicated because the outcomes are not equally likely.

The coin flipping experiment can be simulated using Monte Carlo by writing a program that instructs a computer to flip two coins. Let N denote the iteration count of a loop enclosing the aforementioned logic. If we write the loop-enclosed logic such that it does not include any early loop exits, then N is the total number of random experiments, i.e. simulated coin flips. Our simulated PMF is,

$$\hat{f}(x) = \frac{1}{N} * \begin{cases} \text{number of random experiments resulting in TT} & x = 0, \\ \text{number of random experiments resulting in HT or TH} & x = 1, \\ \text{number of random experiments resulting in HH} & x = 2. \end{cases} \quad (3.5)$$

where I used the “hat” on $\hat{f}(x)$ to denote that $\hat{f}(x)$ is a Monte Carlo approximation of $f(x)$. Our simulation has the property that,

$$\lim_{N \rightarrow \infty} \hat{f}(x) = f(x). \quad (3.6)$$

Eq. (3.6) is a consequence of the Law of Large Numbers (LLN), which states that the empirical probabilities will converge to the theoretical probabilities as the number of trials

in a random experiment goes to infinity¹. Infinite N is infeasible, but N can still be very large, perhaps even large enough to provide a useful MC approximation. In this case we know $f(x)$, so there is no need to approximate it. If we did not know $f(x)$, then computing $\hat{f}(x)$ would be useful, as long as N is sufficiently large.

The coin flipping simulation is a simple Monte Carlo simulation that demonstrates the “frequentist” approach to probability, which defines probability as the long-run frequency of random events. The definitions needed for describing the coin flipping simulation will remain useful for describing more intricate Monte Carlo methods, such as the Monte Carlo component of HSM. A complication arising from the HSM method compared to the coin flipping simulation is that all the random variables in HSM are continuous rather than discrete.

3.1.2 Continuous Random Variables

Three important probability distributions for describing the HSM method are the uniform, exponential, and Gaussian distributions, all of which are distributions of continuous random variables. A continuous random variable X has an uncountable support \mathcal{A} and probability density function (PDF) satisfying the existence conditions,

$$\int_{\mathcal{A}} f(x) dx = 1 \quad \text{and} \quad f(x) \geq 0 \quad \forall x \in \mathbb{R}. \quad (3.7)$$

Probabilities for continuous random variables are integrals instead of sums,

$$P(A) = P(X \in A) = \int_A f(x) dx. \quad (3.8)$$

Given constants a and b such that $a < b$, we can compute probability by integrating,

$$P(a < X < b) = \int_a^b f(x) dx. \quad (3.9)$$

Note that,

$$P(X = a) = \int_a^a f(x) dx = 0, \quad (3.10)$$

whereas $P(X = a) \neq 0$ for a discrete random variable X for which $a \in \mathcal{A}$.

The first of the three important probability distributions for describing the Monte Carlo component of the HSM method is the uniform probability distribution. Let $X \sim U(a, b)$ denote that the continuous random variable X is uniformly distributed between a and b such that $a < b$. The uniform PDF is,

$$f(x) = \frac{1}{b-a} \quad a < x < b. \quad (3.11)$$

¹The Weak and Strong LLN differ in the type of convergence guaranteed, but both types apply here.

The second important probability distribution is the exponential distribution. Let $X \sim \text{exponential}(\lambda)$ denote that X is exponentially distributed with rate λ . The exponential PDF is,

$$f(x) = \lambda e^{-\lambda x} \quad x > 0. \quad (3.12)$$

The rate parameter λ in Eq. (3.12) is constant, but the exponential distribution can also be parameterized by a non-constant rate. Let $X \sim \text{exponential}(\lambda(x))$ denote that X is exponentially distributed with non-constant rate $\lambda(x)$. The associated PDF is,

$$f(x) = \lambda(x) e^{-\int_0^x \lambda(x') dx'} \quad x > 0. \quad (3.13)$$

The third and final important probability distribution for describing the MC component of HSM is the Gaussian distribution, also known as the Normal distribution. Let $X \sim G(u, \sigma^2)$ denote that X is Gaussian distributed with population mean μ and population variance σ^2 . The Gaussian PDF is,

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad -\infty < x < \infty. \quad (3.14)$$

A random variable is fully characterized by its PDF. Moments, which provide an alternative way to describe a random variable, hold particular significance in Monte Carlo methods. This is because estimating a quantity using Monte Carlo can often be framed as the Monte Carlo estimation of a moment.

3.1.3 Probability Distribution Moment Estimation

Probability distribution moments, which are equivalently referred to as moments of a random variable, are integrals of the PDF. In section 1.3.5 Moment Methods for Linear Transport, I said that a moment is an integral, which is also true here. There, the purpose was to derive approximate or equivalent forms of some PDE. Here, the purpose is different. Eq. (3.6) says that our MC approximation converges to the PDF in the limit as $N \rightarrow \infty$, but it says nothing about the magnitude of the difference between the approximation and the PDF at different N . The second central moment, which is called the “variance”, can provide that information. In probability, taking a moment of the PDF is called computing the “expectation” or “expected value”, which can be denoted as,

$$E[g(X)] = \int_{\mathcal{A}} g(X) f(x) dx, \quad (3.15)$$

where $g(X)$ is some function of the random variable X . Taking $g(X) = X^r$ for some $r \in \mathbb{Z}^+$ gives the r th population moment of X about the origin,

$$E[X^r] = \int_{\mathcal{A}} x^r f(x) dx, \quad (3.16)$$

where $r = 1$ defines the population mean, often denoted μ , so $E[X] = \mu$. Taking $g(X) = (X - \mu)^r$ gives the r th population moment of X about the population mean,

$$E[(X - \mu)^r] = \int_{\mathcal{A}} (x - \mu)^r f(x) dx, \quad (3.17)$$

where $r = 2$ defines the population variance, denoted σ^2 or $\text{Var}[X]$, so $E[(X - \mu)^2] = \sigma^2 = \text{Var}[X]$. An equivalent expression for the variance is $\text{Var}[X] = E[X^2] - \mu^2$ because,

$$\begin{aligned} E[(X - \mu)^2] &= E[X^2 - 2\mu X + \mu^2] \\ &= E[X^2] - 2\mu E[X] + \mu^2 \\ &= E[X^2] - \mu^2. \end{aligned} \quad (3.18)$$

From Eq. (3.15), it is clear that the PDF $f(x)$ is required to compute the expected value, but we often do not know $f(x)$. In those cases, we cannot compute the expected value, but we can estimate it. Consider the population mean,

$$E[X] = \mu = \int_{\mathcal{A}} x f(x) dx. \quad (3.19)$$

We can estimate Eq. (3.19) by computing a function of random variables known as the sample mean, denoted \bar{X} ,

$$\mu \approx \bar{X} = \frac{1}{N} \sum_{i=1}^N X_i. \quad (3.20)$$

In Eq. (3.20), X_i is a random variable which describes the outcome of the i th random experiment. The random variables X_1, \dots, X_N are “independent”, meaning that the outcome of the i th random experiment does not affect the outcome of the j th random experiment for $1 \leq i, j \leq N$ with $i \neq j$. The random variables X_1, \dots, X_N are also “identically distributed”, meaning that they all have the same probability distribution. The qualities of independence and identical distribution frequently occur together, leading to the adjectival acronym independent and identically distributed (iid). Thus, X_1, \dots, X_N are iid random variables.

Computing Eq. (3.20) requires realizing the random variables. A “random variate” is a realization of a random variable². One possible sequence of random variates for the coin flipping example could be 0, 2, 0, 1, 1, so \bar{X} would be 4/5. We could also estimate Eq. (3.19) using a single variate. Is \bar{X} better than X_1 for estimating μ ? One’s intuition might indicate that \bar{X} would outperform a single variate because it incorporates information from multiple variates. We can provide mathematical support to this intuition by examining whether \bar{X} and X_1 are unbiased, consistent, and efficient, because estimators tend to perform better if they have these three qualities.

²Another term is “sample”, which typically refers to a collection of realizations, though an individual realization can also be called a sample

An estimator $\hat{\Lambda}$ is an unbiased estimator of Λ if the expectation of the estimator is the quantity estimated,

$$E[\hat{\Lambda}] = \Lambda. \quad (3.21)$$

An estimator is consistent if it converges in probability to the quantity estimated,

$$\lim_{N \rightarrow \infty} P(|\hat{\Lambda} - \Lambda| < \epsilon) = 1, \quad (3.22)$$

for some arbitrary $\epsilon > 0$. Finally, an estimator $\hat{\Lambda}_1$ is more efficient than $\hat{\Lambda}_2$ if it has smaller variance,

$$\text{Var}[\hat{\Lambda}_1] < \text{Var}[\hat{\Lambda}_2]. \quad (3.23)$$

Because the random variables X_1, \dots, X_N in Eq. (3.20) are identically distributed,

$$E[X_i] = \mu \quad \forall i \in \{1, \dots, N\}, \quad (3.24)$$

and

$$\text{Var}[X_i] = \sigma^2 \quad \forall i \in \{1, \dots, N\}. \quad (3.25)$$

This lets us derive $E[\bar{X}]$ and $\text{Var}[\bar{X}]$, which I show in detail in order to highlight the properties of the $E[\cdot]$ and $\text{Var}[\cdot]$ functionals. The expectation is,

$$\begin{aligned} E[\bar{X}] &= E \left[\frac{1}{N} \sum_{i=1}^N X_i \right] \\ &= \frac{1}{N} E \left[\sum_{i=1}^N X_i \right] \\ &= \frac{1}{N} \sum_{i=1}^N E[X_i] \\ &= \frac{1}{N} \sum_{i=1}^N \mu \\ &= \frac{1}{N} N \mu \\ &= \mu. \end{aligned} \quad (3.26)$$

The variance is,

$$\begin{aligned}
 \text{Var}[\bar{X}] &= \text{Var} \left[\frac{1}{N} \sum_{i=1}^N X_i \right] \\
 &= \frac{1}{N^2} \text{Var} \left[\sum_{i=1}^N X_i \right] \\
 &= \frac{1}{N^2} \sum_{i=1}^N \text{Var}[X_i] \\
 &= \frac{1}{N^2} \sum_{i=1}^N \sigma^2 \\
 &= \frac{1}{N^2} N \sigma^2 \\
 &= \sigma^2/N.
 \end{aligned} \tag{3.27}$$

Thus, \bar{X} has population mean μ and variance σ^2/N . We conclude that \bar{X} is unbiased because $E[\bar{X}] = \mu$.

We can show that \bar{X} converges in probability to μ using Chebyshev's inequality,

$$P \left(|\bar{X} - \mu| < \frac{k\sigma}{\sqrt{N}} \right) \geq 1 - \frac{1}{k^2}. \tag{3.28}$$

Letting $k = \epsilon\sqrt{N}/\sigma$ gives,

$$P(|\bar{X} - \mu| < \epsilon) \geq 1 - \frac{\sigma^2}{N\epsilon^2}. \tag{3.29}$$

Taking the limit of both sides gives,

$$\lim_{N \rightarrow \infty} P(|\bar{X} - \mu| < \epsilon) = 1. \tag{3.30}$$

Thus, \bar{X} is a consistent estimator of μ .

Finally, we can compare the efficiencies of \bar{X} with X_1 by comparing their variances. Given that $\text{Var}[X_1] = \sigma^2$ and,

$$\text{Var}[\bar{X}] = \frac{\sigma^2}{N} < \sigma^2 \quad \text{if } N > 1, \tag{3.31}$$

we conclude that \bar{X} is more efficient than X_1 for $N > 1$. This analysis, which demonstrated that \bar{X} is unbiased and consistent, and that \bar{X} is more efficient than X_1 for $N > 1$, supports our intuition that \bar{X} outperforms X_1 for the task of estimating $E[X] = \mu$.

3.1.4 Central Limit Theorem

The Law of Large Numbers says that our approximation of the PMF for the coin flipping experiment converges to the true PMF as the number of coin flips goes to infinity, as expressed by Eq. (3.6). It also says that the sample mean \bar{X} , which is an estimator of μ , converges to μ as the number of random variables N in the sum goes to infinity. However, the Law of Large Numbers says nothing about the rate by which \bar{X} converges to μ , nor does it say anything about the distribution of \bar{X} for $N < \infty$. The Central Limit Theorem (CLT) describes both.

As before, let X_1, X_2, \dots, X_N be iid random variables with finite mean $\mu = E[X_i] < \infty$ and finite variance $\sigma^2 = \text{Var}[X_i] < \infty$, where $i \in \{1, 2, \dots, N\}$. The CLT states that,

$$\bar{X} \xrightarrow{d} G\left(\mu, \frac{\sigma^2}{N}\right), \quad (3.32)$$

where \xrightarrow{d} denotes convergence in distribution. Thus, the sample mean \bar{X} , which is an estimator of μ , eventually becomes a Gaussian-distributed random variable. The PDF of \bar{X} becomes the Gaussian PDF in Eq. (3.14). From Eq. (3.32), the “standard error” of the estimator \bar{X} is,

$$\text{Standard error of } \bar{X} = \sqrt{\frac{\sigma^2}{N}}. \quad (3.33)$$

I refer to Eq. (3.33) as the “uncertainty” of the estimator. The factor of $N^{-1/2}$ in Eq. (3.33) implies that reducing the uncertainty of \bar{X} by a factor of 2 requires increasing N by a factor of 4. If N is fixed, the only way to reduce the uncertainty of the estimator is to decrease the variance σ^2 of the iid random variables X_i , perhaps by choosing a different set of iid random variables Y_i . Techniques which decrease the variance are called “variance reduction techniques”.

The $1/\sqrt{N}$ convergence of Monte Carlo methods is due to the CLT. Quasi-Monte Carlo, which uses quasi-random numbers instead of pseudorandom numbers, improves the convergence of Monte Carlo from $1/\sqrt{N}$ to $1/N$, but is not widely used in radiative transfer because of characteristics of low-discrepancy sequences which violate assumptions that are valid only for (pseudo-)random number sequences. See J. Spanier’s review in Chapter 3 of the manuscript compiled by Azmy and Sartori [67] for details.

3.2 Monte Carlo Simulation as Integral Estimation

Many popular introductory numerical analysis texts have whole chapters on numerical integration using quadrature methods, but almost none of them mention Monte Carlo integration. A good example of this is the textbook by Burden *et al.* [68]. In a very nice chapter on numerical differentiation and integration, the authors prove the double degree minus one result for Gaussian quadrature, describe error estimation for adaptive quadrature, and show

examples of quadrature rules for multiple integration up to 3D,

$$I = \int_{x_1}^{x_2} \int_{y_1}^{y_2} \int_{z_1}^{z_2} f(x, y, z) dz dy dx . \quad (3.34)$$

Quadrature exceeding 3D may be unwise due to poor scaling of quadrature error relative to Monte Carlo error due to the way that the error in the integral approximation scales with the number of function evaluation points. I explain this important phenomenon in the paragraphs that follow.

First, consider quadrature, which typically approximates the integrand as a polynomial, and then integrates the polynomial exactly. The trapezoid rule approximates the integrand as a linear polynomial by evaluating the integrand at two points x_1 and x_2 and then integrating the linear function described by those points. The integral is equivalent to the area of the eponymous trapezoid formed by legs of length $f(x_1)$ and $f(x_2)$ and bases of length $x_2 - x_1$ and the length of the line segment connecting the function values. Subtracting the expression for the area of the trapezoid from a Taylor series expansion of the integrand at the evaluation points cancels the constant and first derivative term, leaving the quadratic term as the largest contributor to the trapezoid rule approximation error. For N equally-spaced evaluation points, the trapezoid rule approximation error is thus $O(1/N^2)$.

Second, consider Monte Carlo integration, which approximates the integrand as a constant, and then integrates the constant exactly³. The constant is an average of the function sampled at randomly selected points on the domain of integration. Thus, we may estimate the integral,

$$I = \int_a^b f(x) dx \quad a < b, \quad (3.35)$$

using Monte Carlo as,

$$\hat{I} = \frac{b-a}{N} \sum_{i=1}^N f(X_i), \quad (3.36)$$

where $X_i \sim U(a, b)$ are iid random variables. First, note that we will always need to generate random variates in order to compute quantities like the \hat{I} estimator in Eq. (3.37), so henceforth in this dissertation I write estimator equations like Eq. (3.37) using,

$$\hat{I} = \frac{b-a}{N} \sum_{i=1}^N f(x_i), \quad (3.37)$$

where x_i are random variates of the random variable $X \sim U(a, b)$. This is a shorthand expression for the more accurate statement that there are actually N iid random variables $X_i \sim U(a, b)$, where $i = 1, \dots, N$, and that X_1, \dots, X_N are the random variables for which one must generate the random variates in Eq. (3.36).

³The juxtaposition of quadrature approximating the integrand as a polynomial and MC as a constant is nicely illustrated by Figure 2.1 in Nakatsukasa [69].

Also, notice that Eq. (3.36) is nearly identical to the sample mean \bar{X} defined by Eq. (3.20). If the domain of integration is the unit line segment $x \in [a, b] = [0, 1]$, then Eqs. (3.20) and (3.36) are both simple averages: Eq. (3.20) is the average of the random variates and Eq. (3.36) is the average of the function evaluations.

Finally, a comparison of the trapezoid rule error and the Monte Carlo uncertainty shows that reducing the error of the numerical integral by a factor of 4 requires only 2x more evaluations, whereas reducing the uncertainty of the Monte Carlo estimate by a factor of 4 requires 16x more evaluations,

$$\begin{aligned} \text{Monte Carlo uncertainty} &\approx N^{-1/2} \\ \text{Trapezoid rule error} &\approx N^{-2}. \end{aligned} \tag{3.38}$$

It would appear that the trapezoid rule significantly outperforms Monte Carlo. This is definitely true if the integral we wish to compute is just one dimensional. However, integration in k dimensions divides quadrature error exponents by k , but not the MC error. Thus, MC outperforms trapezoid rule for integration exceeding 4D, as identified in Lux and Koblinger [64]. Some complications muddy this analysis. There are higher order quadrature rules that have error exponents smaller than negative 2. There is also an active field of research into quasi-Monte Carlo (QMC), which uses numbers from a low-discrepancy sequence instead of random variates, and thereby can achieve an error exponent of negative unity, which is 2x better than the negative one half error exponent of MC⁴. In addition to increasing N or employing a low-discrepancy sequence, the uncertainty in Monte Carlo methods can also be minimized by reducing the variance of the estimator used to approximate the integral.

3.3 Monte Carlo Integration of Linear Transport

The linear transport Eq. (1.13a) and its boundary condition Eq. (1.13b) are not integrals. It would be useful if we had an integral which was equivalent to Eq. (1.13a) because it would allow us to derive an estimator for the solution to Eq. (1.13a). If we had an estimator for the integral then we could also derive functions of the estimator, like the variance. The approach taken in section 2.4 of Spanier and Gelbard [65] and section 4.III in Lux and Koblinger [64] is to introduce the collision and emission densities, derive a Fredholm integral equation of the second kind (FIESK), then derive an estimator for the solution to the FIESK.

My approach begins with the characteristic equation, which I derive in section 3.3.1. Note that I use the characteristic equation only to derive Monte Carlo estimators. I do not use the Method of Characteristics (MOC), which is a deterministic numerical method based on the characteristic equation [71]. Nor do I use its hybrid successor known as the Random Ray Method (TRRM) [72]. I simply mention [71] and [72] as other cases where the authors used the characteristic equation to develop methods for solving the linear transport equation.

⁴For a discussion of QMC for solving the transport equation, which has thus far seen limited success, see J. Spanier's review in Chapter 3 of the manuscript compiled by Azmy and Sartori [67]. Two examples of recently-developed MC transport methods employing QMC are [35] and [70].

3.3.1 Derivation of Characteristic Equation

The goal in this section is to derive an equation called the characteristic equation, which we will use to derive the estimators in the HSM method. Consider the steady-state, gray, linear transport Eq. (1.13a) rewritten slightly,

$$\boldsymbol{\Omega} \cdot \nabla \psi(\mathbf{x}, \boldsymbol{\Omega}) + \sigma_t \psi(\mathbf{x}, \boldsymbol{\Omega}) = Q(\mathbf{x}, \boldsymbol{\Omega}) \quad \mathbf{x} \in \mathcal{D}, \quad (3.39a)$$

where $\sigma_t = \sigma_t(\mathbf{x})$, $Q(\mathbf{x}, \boldsymbol{\Omega}) = \frac{\sigma_s}{4\pi} \varphi(\mathbf{x}) + q(\mathbf{x}, \boldsymbol{\Omega})$, and $\sigma_s = \sigma_s(\mathbf{x})$. Eq. (3.39a) is subject to the boundary condition,

$$\psi(\mathbf{x}, \boldsymbol{\Omega}) = \bar{\psi}(\mathbf{x}, \boldsymbol{\Omega}), \quad \mathbf{x} \in \partial\mathcal{D} \text{ and } \boldsymbol{\Omega} \cdot \mathbf{n} < 0. \quad (3.39b)$$

In Eq. (3.39a), $\varphi(\mathbf{x})$ is a known quantity that replaces the unknown quantity,

$$\int_{S^2} \psi \, d\Omega', \quad (3.40)$$

which appears in Eq. (1.13a). This simplification of the scattering source is exclusive to methods which do not have scattering events, but rather compute $\varphi(\mathbf{x})$ in an iteration. HSM uses a moment solve to compute $\varphi(\mathbf{x})$. Methods which do not compute $\varphi(\mathbf{x})$ can incorporate the effect of the scattering term with scattering events.

To begin the derivation, observe that in Eq. (3.39a), $\boldsymbol{\Omega} \cdot \nabla \psi = \Omega_x \partial \psi / \partial x + \Omega_y \partial \psi / \partial y + \Omega_z \partial \psi / \partial z$. Now consider the notation change,

$$\mathbf{x} \rightarrow \mathbf{r} - s\boldsymbol{\Omega} = (r_x - s\Omega_x, r_y - s\Omega_y, r_z - s\Omega_z)^T, \quad (3.41)$$

where s is the length of the ray emanating from \mathbf{r} in the negative $\boldsymbol{\Omega}$ direction. The position \mathbf{x} has the following derivatives with respect to the new parameter s ,

$$\frac{dx}{ds} = -\Omega_x \quad \frac{dy}{ds} = -\Omega_y \quad \frac{dz}{ds} = -\Omega_z. \quad (3.42)$$

The total derivative of ψ with respect to the new spatial parameter s is,

$$\begin{aligned} \frac{d\psi}{ds} &= \frac{\partial \psi}{\partial x} \frac{dx}{ds} + \frac{\partial \psi}{\partial y} \frac{dy}{ds} + \frac{\partial \psi}{\partial z} \frac{dz}{ds} \\ &= - \left(\frac{\partial \psi}{\partial x} \Omega_x + \frac{\partial \psi}{\partial y} \Omega_y + \frac{\partial \psi}{\partial z} \Omega_z \right) \\ &= -\boldsymbol{\Omega} \cdot \nabla \psi. \end{aligned} \quad (3.43)$$

Applying the notation change to Eq. (3.39a) gives,

$$-\frac{d}{ds} \psi(\mathbf{r} - s\boldsymbol{\Omega}, \boldsymbol{\Omega}) + \sigma_t \psi(\mathbf{r} - s\boldsymbol{\Omega}, \boldsymbol{\Omega}) = Q(\mathbf{r} - s\boldsymbol{\Omega}, \boldsymbol{\Omega}), \quad (3.44a)$$

where $\sigma_t = \sigma_t(\mathbf{r} - s\boldsymbol{\Omega})$, $Q(\mathbf{r} - s\boldsymbol{\Omega}, \boldsymbol{\Omega}) = \frac{\sigma_s}{4\pi}\varphi(\mathbf{r} - s\boldsymbol{\Omega}) + q(\mathbf{r} - s\boldsymbol{\Omega}, \boldsymbol{\Omega})$, and $\sigma_s = \sigma_s(\mathbf{r} - s\boldsymbol{\Omega})$. The boundary condition is,

$$\psi(\mathbf{r} - s\boldsymbol{\Omega}, \boldsymbol{\Omega}) = \psi_{\text{inc}}(\mathbf{r} - s_0\boldsymbol{\Omega}, \boldsymbol{\Omega}), s_0 \in \partial\mathcal{D} \text{ and } \boldsymbol{\Omega} \cdot \mathbf{n} < 0, \quad (3.44b)$$

where I have replaced the $\bar{\psi}$ notation with ψ_{inc} to make the symbol easier to visually distinguish from ψ , and $s_0(\mathbf{r}, \boldsymbol{\Omega})$ denotes the minimum distance from \mathbf{r} to the boundary $\partial\mathcal{D}$ when traveling in negative $\boldsymbol{\Omega}$ direction (see Fig. 3.1). Mathematically,

$$s_0(\mathbf{r}, \boldsymbol{\Omega}) = \min\{s \mid \mathbf{r} - s\boldsymbol{\Omega} \in \partial\mathcal{D}\}. \quad (3.45)$$

Define an integrating factor,

$$I(s) = e^{-\int_0^s \sigma_t(\mathbf{r} - \eta\boldsymbol{\Omega}) d\eta}, \quad (3.46)$$

which has a derivative of,

$$\frac{dI}{ds} = -\sigma_t(\mathbf{r} - s\boldsymbol{\Omega})I. \quad (3.47)$$

Multiply Eq. (3.44a) by the integrating factor in Eq. (3.46) to get,

$$\begin{aligned} -\frac{d}{ds}\psi I + \sigma_t\psi I &= IQ \\ -\frac{d}{ds}(\psi I) &= IQ \\ -\int_0^{s_0} \frac{d}{ds}(\psi I) ds &= \int_0^{s_0} IQ ds \\ -(\psi(s_0, \boldsymbol{\Omega})I(s_0) - \psi(0, \boldsymbol{\Omega})I(0)) &= \int_0^{s_0} IQ ds \\ \psi(0, \boldsymbol{\Omega})I(0) - \psi(s_0, \boldsymbol{\Omega})I(s_0) &= \int_0^{s_0} IQ ds \\ \psi(\mathbf{r}, \boldsymbol{\Omega}) &= \psi(\mathbf{r} - s_0\boldsymbol{\Omega}, \boldsymbol{\Omega})I(s_0) + \int_0^{s_0} IQ ds \\ \psi(\mathbf{r}, \boldsymbol{\Omega}) &= e^{-\int_0^{s_0} \sigma_t(\mathbf{r} - \eta\boldsymbol{\Omega}) d\eta} \psi_{\text{inc}}(\mathbf{r} - s_0\boldsymbol{\Omega}, \boldsymbol{\Omega}) + \int_0^{s_0} IQ ds. \end{aligned} \quad (3.48)$$

Substituting Eq. (3.46) once more into Eq. (3.48) gives the final result,

$$\psi(\mathbf{r}, \boldsymbol{\Omega}) = e^{-\int_0^{s_0} \sigma_t(\mathbf{r} - \eta\boldsymbol{\Omega}) d\eta} \psi_{\text{inc}}(\mathbf{r} - s_0\boldsymbol{\Omega}, \boldsymbol{\Omega}) + \int_0^{s_0} e^{-\int_0^s \sigma_t(\mathbf{r} - \eta\boldsymbol{\Omega}) d\eta} Q(\mathbf{r} - s\boldsymbol{\Omega}, \boldsymbol{\Omega}) ds. \quad (3.49)$$

Imagine placing a detector at the phase-space location $(\mathbf{r}, \boldsymbol{\Omega})$ denoted by the black dot located at \mathbf{r} in Fig. 3.1. The first term in the characteristic Eq. (3.49) accounts for particles sourced on the domain boundary $\partial\mathcal{D}$ with direction $\boldsymbol{\Omega}$ and exponentially attenuated en route to the detector, and the second term accounts for particles sourced anywhere along s_0 with direction $\boldsymbol{\Omega}$ and exponentially attenuated en route to the detector.

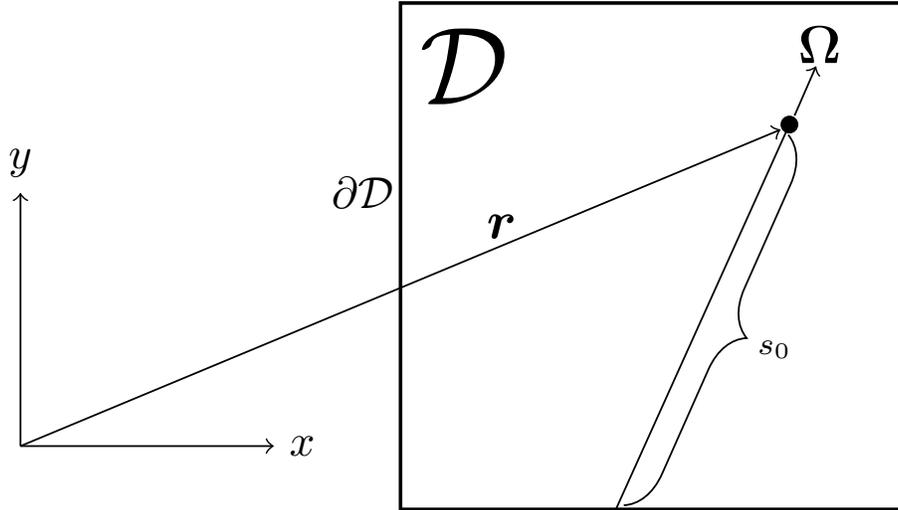


Figure 3.1: Distance to domain boundary $s_0(\mathbf{r})$.

3.3.2 Derivation of $\hat{\phi}$ Estimator

The goal in this section is to derive an estimator for the angle integrated intensity, $\phi(\mathbf{x})$, which I denote $\hat{\phi}$. Why estimate $\phi(\mathbf{x})$ instead of $\psi(\mathbf{x}, \Omega)$? Almost all radiative transfer applications need the angle integrated intensity and almost none need the angular intensity. For example, applications often require the radiation energy absorbed by the matter in a volume, which is the product of the intensity and the absorption opacity integrated over all five dimensions,

$$\int_V \int_{\mathbb{S}^2} \psi(\mathbf{x}, \Omega) \sigma_a(\mathbf{x}) d\Omega d\mathbf{x} = \int_V \phi(\mathbf{x}) \sigma_a(\mathbf{x}) d\mathbf{x}, \quad (3.50)$$

where I used the definition of $\phi(\mathbf{x})$ in Eq. (1.17) and V is the volume of interest. Eq. (3.50) shows that estimating ψ to compute Eq. (3.50) would be less accurate because it would require approximating its angular integral, whereas estimating ϕ to compute Eq. (3.50) does not require subsequent integration. It would also be wasteful to estimate ψ because it would unnecessarily consume computer memory for storing some representation of the solution along the Ω dimensions. A final reason to prefer $\hat{\phi}$ is that we can use $\hat{\phi}$ to compute the SMM correction tensor \mathbf{T} , defined in Eq. (1.42). We do not need to use ψ to compute it.

The derivation of $\hat{\phi}$ follows the following sequence of steps and substeps ending with random variate generation:

1. Formally integrate the characteristic equation over all directions and over the volume enclosed by a single mesh element,
2. Estimate the integral using Monte Carlo,

- a) Change variables such that $s = 0$ defines a point on $\partial\mathcal{D}$ instead of \mathcal{D} ,
- b) Define sampling procedures for the volume and boundary sources,
 - i. Define random variables, a joint PDF, and a function of the random variables such that the expectation of the function is the integral that we want to compute,
 - ii. Define the techniques for generating the random variates of the random variables that we defined.

All the above steps are covered here, except for the random variate generation, which will be discussed in section 3.3.4. Our derivation of $\hat{\phi}$ begins by integrating Eq. (3.49) over angle,

$$\int_{\mathbb{S}^2} \psi(\mathbf{r}, \boldsymbol{\Omega}) d\Omega = \int_{\mathbb{S}^2} e^{-\int_0^{s_0} \sigma_t(\mathbf{r}-\eta\boldsymbol{\Omega}) d\eta} \psi_{\text{inc}}(\mathbf{r} - s_0\boldsymbol{\Omega}, \boldsymbol{\Omega}) d\Omega + \int_{\mathbb{S}^2} \int_0^{s_0} e^{-\int_0^s \sigma_t(\mathbf{r}-\eta\boldsymbol{\Omega}) d\eta} Q(\mathbf{r} - s\boldsymbol{\Omega}, \boldsymbol{\Omega}) ds d\Omega. \quad (3.51)$$

We impose the same spatial discretization concept used in the deterministic component of HSM and described by Eq. (2.1). That is, we impose a mesh. We represent the quantity described by Eq. (3.51) as a piecewise constant function that is single-valued on each mesh element by averaging Eq. (3.51) over an arbitrary element K ,

$$\frac{1}{\text{vol}(K)} \int_K \int_{\mathbb{S}^2} \psi(\mathbf{r}, \boldsymbol{\Omega}) d\Omega d\mathbf{r} = \frac{1}{\text{vol}(K)} \int_K \int_{\mathbb{S}^2} e^{-\int_0^{s_0} \sigma_t(\mathbf{r}-\eta\boldsymbol{\Omega}) d\eta} \psi_{\text{inc}}(\mathbf{r} - s_0\boldsymbol{\Omega}, \boldsymbol{\Omega}) d\Omega d\mathbf{r} + \frac{1}{\text{vol}(K)} \int_K \int_{\mathbb{S}^2} \int_0^{s_0} e^{-\int_0^s \sigma_t(\mathbf{r}-\eta\boldsymbol{\Omega}) d\eta} Q(\mathbf{r} - s\boldsymbol{\Omega}, \boldsymbol{\Omega}) ds d\Omega d\mathbf{r}. \quad (3.52)$$

Eq. (3.52) is an integral equation for the classic piecewise constant Monte Carlo estimator of the angle integrated intensity. The left-hand side of Eq. (3.52) can be approximated using Monte Carlo integration, just like the model integration problem in Eq. (3.35). Eq. (3.52) is a linear transport equation that is more suitable to Monte Carlo integration than the linear transport Eq. (1.13a). If we knew ψ , then the Monte Carlo approximation of Eq. (3.52) could be written as,

$$\frac{1}{\text{vol}(K)} \int_K \int_{\mathbb{S}^2} \psi(\mathbf{r}, \boldsymbol{\Omega}) d\Omega d\mathbf{r} \approx \frac{4\pi}{N} \sum_{n=1}^N \psi(\mathbf{r}^{(n)}, \boldsymbol{\Omega}^{(n)}), \quad (3.53)$$

where $\mathbf{r}^{(n)}$ and $\boldsymbol{\Omega}^{(n)}$ are random variates of uniform iid random variables in space and angle, respectively. We do not know ψ , so the rest of this section describes how to use the right-hand side of Eq. (3.52) to compute the sum in Eq. (3.53). Consider the change of variables,

$$\mathbf{r}' = \mathbf{r} - s\boldsymbol{\Omega}, \quad \boldsymbol{\Omega}' = \boldsymbol{\Omega}, \quad s' = s. \quad (3.54)$$

Let $S \sim \text{exponential}(\sigma_t(\mathbf{r}' + s\boldsymbol{\Omega}))$ be a random variable representing the distance traveled by a MC photon originating from the point \mathbf{r}' and traveling in the direction $\boldsymbol{\Omega}$ before it collides with an electron in the matter. We refer to s as the “path length”. The PDF for the path length s given a fixed position \mathbf{r}' and direction $\boldsymbol{\Omega}$ is,

$$p_{\sigma_t}(s) = \sigma_t(\mathbf{r}' + s\boldsymbol{\Omega})e^{-\int_0^s \sigma_t(\mathbf{r}' + \eta\boldsymbol{\Omega})d\eta}. \quad (3.55)$$

Define the distance traveled in element K as,

$$\tau(\mathbf{r}', \boldsymbol{\Omega}, s; K) = \begin{cases} 0 & s < s_1(\mathbf{r}', \boldsymbol{\Omega}; K), \\ s - s_1(\mathbf{r}', \boldsymbol{\Omega}; K) & s_1(\mathbf{r}', \boldsymbol{\Omega}; K) \leq s \leq s_2(\mathbf{r}', \boldsymbol{\Omega}; K), \\ s_2(\mathbf{r}', \boldsymbol{\Omega}; K) - s_1(\mathbf{r}', \boldsymbol{\Omega}; K) & s_2(\mathbf{r}', \boldsymbol{\Omega}; K) > s, \end{cases} \quad (3.56a)$$

where s_1 and s_2 are the distances to the entry point and exit points of K along $\boldsymbol{\Omega}$, respectively,

$$s_1(\mathbf{r}', \boldsymbol{\Omega}; K) = \min\{s \mid \mathbf{r}' + s\boldsymbol{\Omega} \in \partial K\}, \quad (3.56b)$$

$$s_2(\mathbf{r}', \boldsymbol{\Omega}; K) = \max\{s \mid \mathbf{r}' + s\boldsymbol{\Omega} \in \partial K\}. \quad (3.56c)$$

Fig. 3.2 shows \mathbf{r}' along with three possible absorption locations corresponding to three values of τ . Observe that $\mathbf{r}' \in \partial\mathcal{D}$. The final step in this change of variables $\mathbf{r} \rightarrow \mathbf{r}'$ is to relate the volume elements $d\mathbf{r}$ and $d\mathbf{r}'$.

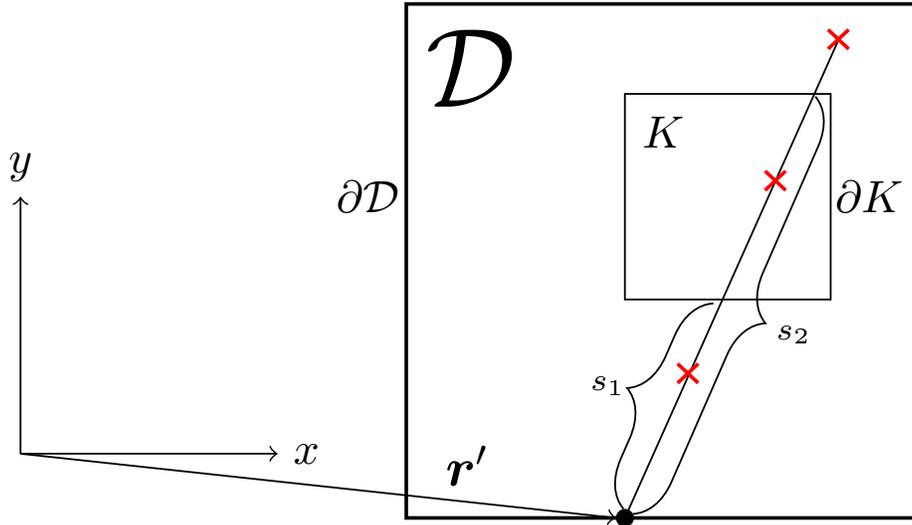


Figure 3.2: Distance to element entry $s_1(\mathbf{r}')$ and exit $s_2(\mathbf{r}')$ and three possible absorption locations marked by **X**'s along the path $\mathbf{r}' + s\boldsymbol{\Omega}$. Here, $\tau = 0$ for the **X** closest to \mathbf{r}' , $\tau = s - s_1$ for the middle **X**, and $\tau = s_2 - s_1$ for the far **X**.

Let $\mathbf{n}(\mathbf{r}')$ denote the surface normal vector at $\mathbf{r}' \in \partial\mathcal{D}$. Consider the volume, made by an infinitesimal area element dA on the boundary $\partial\mathcal{D}$ extruded by distance ds in direction $\boldsymbol{\Omega}$,

$$\{\mathbf{r}' + s\boldsymbol{\Omega} \mid \mathbf{r}' \in dA \subset \partial\mathcal{D}, s \in \hat{s} + ds\}. \quad (3.57)$$

In two spatial dimensions, this volume is a function of the area of the parallelogram in $\text{span}\{\mathbf{n}, \mathbf{n}^\perp\}$, where \mathbf{n}^\perp is the unit vector perpendicular to the surface normal \mathbf{n} , formed from $\mathbf{n} \cdot \mathbf{n}^\perp = 0$ and shown in Fig. 3.3. The volume is,

$$\begin{aligned} ds \, dA \, |\mathbf{n}^\perp \times \boldsymbol{\Omega}| &= ds \, dA \left| \det \begin{pmatrix} n_2 & \Omega_1 \\ -n_1 & \Omega_2 \end{pmatrix} \right| \\ &= ds \, dA \, |n_2\Omega_2 + n_1\Omega_1| \\ &= ds \, dA \, |\boldsymbol{\Omega} \cdot \mathbf{n}|. \end{aligned} \quad (3.58)$$

The subscripts in Eq. (3.58) denote the entry number of the value in the corresponding vector. Also note that,

$$\int_0^s \sigma_t(\mathbf{r} - \eta\boldsymbol{\Omega}) \, d\eta = \int_0^s \sigma_t(\mathbf{r}' + (s - \eta)\boldsymbol{\Omega}) \, d\eta. \quad (3.59)$$

Letting $\tilde{\eta} = s - \eta$, so $d\tilde{\eta} = -d\eta$, we have,

$$\begin{aligned} \int_0^s \sigma_t(\mathbf{r}' + (s - \eta)\boldsymbol{\Omega}) \, d\eta &= - \int_s^0 \sigma_t(\mathbf{r}' + \tilde{\eta}\boldsymbol{\Omega}) \, d\tilde{\eta} \\ &= \int_0^s \sigma_t(\mathbf{r}' + \eta\boldsymbol{\Omega}) \, d\eta. \end{aligned} \quad (3.60)$$

Therefore, the integrand which is expressed in \mathbf{r} coordinates,

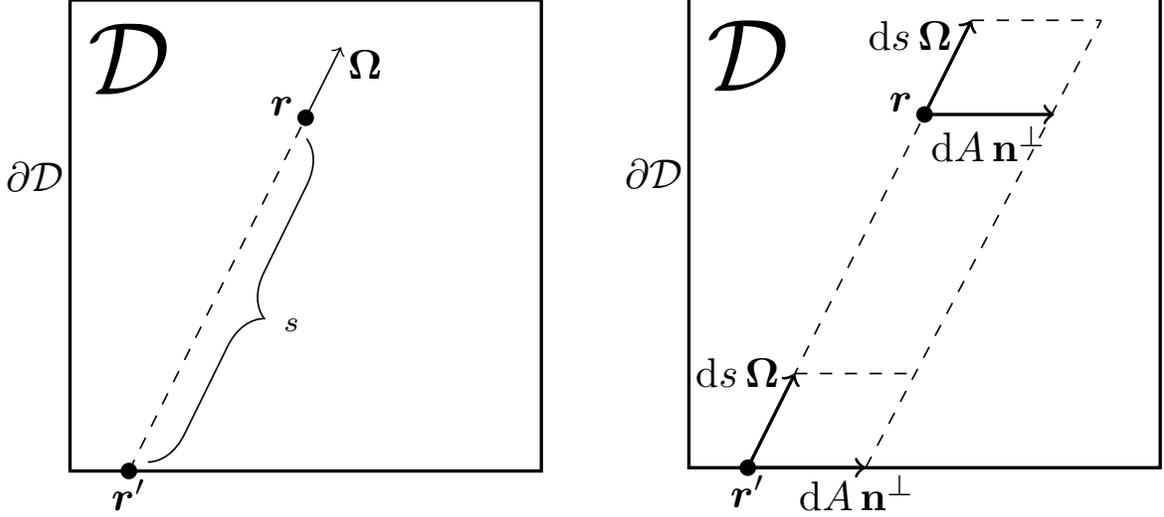
$$e^{-\int_0^{s_0} \sigma_t(\mathbf{r} - \eta\boldsymbol{\Omega}) \, d\eta} \psi_{\text{inc}}(\mathbf{r} - s_0\boldsymbol{\Omega}, \boldsymbol{\Omega}) \, d\mathbf{r} \quad (3.61)$$

can be equivalently represented as an integrand in \mathbf{r}' coordinates,

$$e^{-\int_0^s \sigma_t(\mathbf{r}' + \eta\boldsymbol{\Omega}) \, d\eta} \psi_{\text{inc}}(\mathbf{r}', \boldsymbol{\Omega}) \, |\boldsymbol{\Omega} \cdot \mathbf{n}| \, ds \, dA. \quad (3.62)$$

The change of variables $\mathbf{r} \rightarrow \mathbf{r}'$ generates a factor of $|\boldsymbol{\Omega} \cdot \mathbf{n}|$ in the boundary surface integral because the infinitesimal volume element $d\mathbf{r}$ is equal to $|\boldsymbol{\Omega} \cdot \mathbf{n}| \, ds \, dA$ in \mathbf{r}' coordinates.

It is convenient to treat the two terms on the right-hand side of Eq. (3.52) as distinct sources of MC photons. The term with Q is a volume source and we handle it according to the description in **Volume source contribution to $\hat{\phi}$** . The term with ψ_{inc} is an inflow boundary source and we handle it according to the description in **Boundary source contribution to $\hat{\phi}$** . We then describe how we sample these two sources in **Random Variate Generation**.



(a) Change of variables.

(b) Volume scaling from change of variables.

Figure 3.3: The change of variables $\mathbf{r} \rightarrow \mathbf{r}'$ generates a factor of $|\boldsymbol{\Omega} \cdot \mathbf{n}|$ in the boundary surface integral because $d\mathbf{r} = |\boldsymbol{\Omega} \cdot \mathbf{n}| ds dA$.

Volume source contribution to $\hat{\phi}$

Let $\mathbf{R} \sim U(\mathcal{D})$, $\boldsymbol{\omega} \sim U(\mathbb{S}^2)$, and $S \sim \text{exponential}(\sigma_t(\mathbf{R} + s\boldsymbol{\omega}))$ be random variables representing the initial position, direction, and distance traveled by a MC photon from the volume source, respectively. Define the joint PDF of \mathbf{R} , $\boldsymbol{\omega}$, and S as,

$$p(\mathbf{r}', \boldsymbol{\Omega}, s) = \frac{1}{\text{vol}(\mathcal{D})} \frac{1}{4\pi} p_{\sigma_t}(\mathbf{r}', \boldsymbol{\Omega}, s), \quad (3.63)$$

with domain $\mathcal{A} = \mathcal{D} \times \mathbb{S}^2 \times [0, \infty)$ and $p_{\sigma_t}(\mathbf{r}', \boldsymbol{\Omega}, s)$ defined in Eq. (3.55). Eq. (3.63) is a PDF because,

$$p(\mathbf{r}', \boldsymbol{\Omega}, s) \geq 0 \quad \forall (\mathbf{r}', \boldsymbol{\Omega}, s) \in \mathcal{A}, \quad (3.64a)$$

and

$$\int_{\mathcal{A}} p(\mathbf{r}', \boldsymbol{\Omega}, s) ds d\boldsymbol{\Omega} d\mathbf{r}' = 1. \quad (3.64b)$$

Define a function $f(\mathbf{R}, \boldsymbol{\omega}, S)$ of the random variables \mathbf{R} , $\boldsymbol{\omega}$, and S which represents the contribution of the volume source $Q(\mathbf{r}', \boldsymbol{\Omega})$ to the angle integrated intensity ϕ ,

$$f(\mathbf{R}, \boldsymbol{\omega}, S) = 4\pi \text{vol}(\mathcal{D}) Q(\mathbf{r}', \boldsymbol{\Omega}) \tau(\mathbf{r}', \boldsymbol{\Omega}, s; K). \quad (3.65)$$

If $\psi_{\text{inc}} = 0$ then Eq. (3.52) can now be written in terms of the expectation of $f(\mathbf{R}, \boldsymbol{\omega}, S)$,

$$\frac{1}{\text{vol}(K)} E[f(\mathbf{R}, \boldsymbol{\omega}, S)] = \frac{1}{\text{vol}(K)} \int_{\mathcal{D}} \int_{\mathbb{S}^2} \int_0^{\infty} f(\mathbf{r}', \boldsymbol{\Omega}, s) p(\mathbf{r}', \boldsymbol{\Omega}, s) ds d\boldsymbol{\Omega} d\mathbf{r}', \quad (3.66)$$

where $p(\mathbf{r}', \boldsymbol{\Omega}, s)$ is the joint PDF defined in Eq. (3.63). The Monte Carlo approximation of the expectation in Eq. (3.66) is,

$$E[f(\mathbf{R}, \boldsymbol{\omega}, S)] \approx \frac{1}{N} \sum_{n=1}^N f\left(\mathbf{r}^{(n)}, \boldsymbol{\Omega}^{(n)}, s^{(n)}\right), \quad (3.67)$$

where $(\mathbf{r}^{(n)}, \boldsymbol{\Omega}^{(n)}, s^{(n)})$, $n = 1, \dots, N$, are random variates of the random variables \mathbf{R} , $\boldsymbol{\omega}$, and S . Substituting Eq. (3.65) into Eq. (3.67) gives,

$$\frac{1}{N} \sum_{n=1}^N f\left(\mathbf{r}^{(n)}, \boldsymbol{\Omega}^{(n)}, s^{(n)}\right) = \frac{4\pi \text{vol}(\mathcal{D})}{N} \sum_{n=1}^N Q\left(\mathbf{r}^{(n)}, \boldsymbol{\Omega}^{(n)}\right) \tau\left(\mathbf{r}^{(n)}, \boldsymbol{\Omega}^{(n)}, s^{(n)}; K\right). \quad (3.68)$$

Eq. (3.68) is the contribution of the volume source to the $\hat{\phi}$ estimator. The sum in Eq. (3.68) is over all N particles sourced in the volume \mathcal{D} . We can rewrite Eq. (3.68) as a sum over all volume source particle paths traversed in some element K . Assume that volume source particle n traversed path i in element K and define its weight as,

$$w_i = \frac{4\pi \text{vol}(\mathcal{D})}{N} Q\left(\mathbf{r}^{(n)}, \boldsymbol{\Omega}^{(n)}\right). \quad (3.69)$$

Define the length of path i as,

$$d_i = \tau\left(\mathbf{r}^{(n)}, \boldsymbol{\Omega}^{(n)}, s^{(n)}; K\right). \quad (3.70)$$

Eq. (3.68) can now be rewritten as the sum over paths traversed by volume source particles in K ,

$$\sum_{i=1} w_i d_i. \quad (3.71)$$

Multiplying Eq. (3.71) by the inverse volume in front of the expectation in Eq. (3.66) gives us our final result for the contribution of the volume source to the estimator $\hat{\phi}$ on K ,

$$\frac{1}{\text{vol}(K)} \sum_{i=1} w_i d_i. \quad (3.72)$$

Boundary source contribution to $\hat{\phi}$

Let $\mathbf{R}_b \sim U(\partial\mathcal{D})$, $\boldsymbol{\omega}_h \sim U(\mathbb{S}_h^2)$, and $S_b \sim \text{exponential}(\sigma_t(\mathbf{R}_b + s\boldsymbol{\omega}_h))$ be random variables representing the initial position, direction, and distance traveled by a MC photon from the boundary source, respectively, where \mathbb{S}_h^2 is all directions on the unit hemisphere defined by $\boldsymbol{\Omega} \cdot \mathbf{n} < 0$. Define the joint PDF of \mathbf{R}_b , $\boldsymbol{\omega}_h$, and S_b as,

$$\rho(\mathbf{r}', \boldsymbol{\Omega}, s) = \frac{1}{\text{area}(\partial\mathcal{D})} \frac{1}{2\pi} p_{\sigma_t}(\mathbf{r}', \boldsymbol{\Omega}, s), \quad (3.73)$$

with domain $\mathcal{B} = \partial\mathcal{D} \times \mathbb{S}_h^2 \times [0, \infty)$ and $p_{\sigma_t}(\mathbf{r}', \boldsymbol{\Omega}, s)$ defined in Eq. (3.55). Eq. (3.73) is a PDF because,

$$\rho(\mathbf{r}', \boldsymbol{\Omega}, s) \geq 0 \quad \forall(\mathbf{r}', \boldsymbol{\Omega}, s) \in \mathcal{B}, \quad (3.74a)$$

and

$$\int_{\mathcal{B}} |\boldsymbol{\Omega} \cdot \mathbf{n}| \rho(\mathbf{r}', \boldsymbol{\Omega}, s) ds d\Omega dA = 1. \quad (3.74b)$$

Define a function $g(\mathbf{R}_b, \boldsymbol{\omega}_h, S_b)$ of the random variables $\mathbf{R}_b, \boldsymbol{\omega}_h$, and S_b which represents the contribution of the boundary source $\psi_{\text{inc}}(\mathbf{r}', \boldsymbol{\Omega})$ to the angle integrated intensity ϕ ,

$$g(\mathbf{R}_b, \boldsymbol{\omega}_h, S) = 2\pi \text{area}(\partial\mathcal{D}) \psi_{\text{inc}}(\mathbf{r}', \boldsymbol{\Omega}) \tau(\mathbf{r}', \boldsymbol{\Omega}, s; K). \quad (3.75)$$

If $Q = 0$ then Eq. (3.52) can now be written in terms of the expectation of $g(\mathbf{R}_b, \boldsymbol{\omega}_h, S_b)$,

$$\frac{1}{\text{vol}(K)} E \left[g(\mathbf{R}_b, \boldsymbol{\omega}_h, S) \right] = \int_{\partial\mathcal{D}} \int_{\boldsymbol{\Omega} \cdot \mathbf{n} < 0} \int_0^\infty |\boldsymbol{\Omega} \cdot \mathbf{n}| g(\mathbf{r}', \boldsymbol{\Omega}, s) \rho(\mathbf{r}', \boldsymbol{\Omega}, s) ds d\Omega dA, \quad (3.76)$$

where $\rho(\mathbf{r}', \boldsymbol{\Omega}, s)$ is the joint PDF defined in Eq. (3.73). The Monte Carlo approximation of the expectation in Eq. (3.76) is,

$$E \left[g(\mathbf{R}_b, \boldsymbol{\omega}_h, S) \right] \approx \frac{1}{M} \sum_{m=1}^M g \left(\mathbf{r}_b^{(m)}, \boldsymbol{\Omega}_h^{(m)}, s_b^{(m)} \right), \quad (3.77)$$

where $(\mathbf{r}_b^{(m)}, \boldsymbol{\Omega}_h^{(m)}, s_b^{(m)})$, $m = 1, \dots, M$, are random variates of the random variables $\mathbf{R}_b, \boldsymbol{\omega}_h$, and S_b . Substituting Eq. (3.75) into Eq. (3.77) gives,

$$\begin{aligned} & \frac{1}{M} \sum_{m=1}^M g \left(\mathbf{r}_b^{(m)}, \boldsymbol{\Omega}_h^{(m)}, s_b^{(m)} \right) \\ &= \frac{2\pi \text{area}(\partial\mathcal{D})}{M} \sum_{m=1}^M |\boldsymbol{\Omega}_h^{(m)} \cdot \mathbf{n}| \psi_{\text{inc}} \left(\mathbf{r}_b^{(m)}, \boldsymbol{\Omega}_h^{(m)} \right) \tau \left(\mathbf{r}_b^{(m)}, \boldsymbol{\Omega}_h^{(m)}, s_b^{(m)}; K \right). \end{aligned} \quad (3.78)$$

Eq. (3.78) is the contribution of the boundary source to the $\hat{\phi}$ estimator. The sum in Eq. (3.78) is over all M particles sourced on the surface $\partial\mathcal{D}$. We can rewrite Eq. (3.78) as a sum over all boundary source particle paths traversed in some element K . Assume that boundary source particle m traversed path j in element K and define its weight as,

$$w_j = \frac{2\pi \text{area}(\partial\mathcal{D})}{M} |\boldsymbol{\Omega}_h^{(m)} \cdot \mathbf{n}| \psi_{\text{inc}} \left(\mathbf{r}_b^{(m)}, \boldsymbol{\Omega}_h^{(m)} \right). \quad (3.79)$$

Define the length of path j as,

$$d_j = \tau \left(\mathbf{r}_b^{(m)}, \boldsymbol{\Omega}_h^{(m)}, s_b^{(m)}; K \right). \quad (3.80)$$

Eq. (3.78) can now be rewritten as the sum over paths traversed by boundary source particles in K ,

$$\sum_{j=1} w_j d_j. \quad (3.81)$$

Multiplying Eq. (3.81) by the inverse volume in front of the expectation in Eq. (3.76) gives us our final result for the contribution of the boundary source to the estimator $\hat{\phi}$ on K ,

$$\frac{1}{\text{vol}(K)} \sum_{j=1} w_j d_j. \quad (3.82)$$

The estimator $\hat{\phi}$ is the sum of Eq. (3.72) and Eq. (3.82). Estimators like $\hat{\phi}$ are sometimes called “path-length” estimators because of their dependence on the length of the particle paths. One remaining challenge that we need to address in order to compute $\hat{\phi}$ is the generation of random variates. Section 3.3.4 Random Variate Generation addresses this challenge. The next section shows the derivation of $\hat{\phi}_s$, which is similar to $\hat{\phi}$.

3.3.3 Derivation of $\hat{\phi}_s$ Estimator

The goal in this section is to derive an estimator for $\phi(\mathbf{x})$ on the domain boundary $\partial\mathcal{D}$, which I denote $\hat{\phi}_s$. The subscript s denotes that $\hat{\phi}_s$ is a surface tally. This contrasts with $\hat{\phi}$, derived in section 3.3.2, which is a volume tally. The domain of $\hat{\phi}$ is the mesh elements, whereas the domain of $\hat{\phi}_s$ is mesh element faces. Why estimate $\phi(\mathbf{x})$ on boundary faces? We need $\hat{\phi}_s$ to compute the SMM boundary correction factor β , defined in Eq. (1.45), but there are many other applications which use boundary surface tallies to calculate domain boundary phenomena. One example is energy leakage, which is an important domain boundary phenomenon that one could calculate using a boundary surface tally resembling $\hat{\phi}_s$.

The derivation of $\hat{\phi}_s$ follows the same sequence of steps and substeps as $\hat{\phi}$ except in step 1. Specifically, after integrating the characteristic equation over all directions, we then integrate over the *surface* defined by a single boundary face in the mesh, rather than the *volume* enclosed by a single mesh element. Thus, we begin by averaging Eq. (3.51) over an arbitrary boundary face $\mathcal{F} \in \Gamma_b$,

$$\begin{aligned} \frac{1}{\text{area}(\mathcal{F})} \int_{\mathcal{F}} \int_{\mathbb{S}^2} \psi(\mathbf{r}, \boldsymbol{\Omega}) \, d\Omega \, dF &= \frac{1}{\text{area}(\mathcal{F})} \int_{\mathcal{F}} \int_{\mathbb{S}^2} e^{-\int_0^{s_0} \sigma_t(\mathbf{r}-\eta\boldsymbol{\Omega}) \, d\eta} \psi_{\text{inc}}(\mathbf{r} - s_0\boldsymbol{\Omega}, \boldsymbol{\Omega}) \, d\Omega \, dF \\ &+ \frac{1}{\text{area}(\mathcal{F})} \int_{\mathcal{F}} \int_{\mathbb{S}^2} \int_0^{s_0} e^{-\int_0^s \sigma_t(\mathbf{r}-\eta\boldsymbol{\Omega}) \, d\eta} Q(\mathbf{r} - s\boldsymbol{\Omega}, \boldsymbol{\Omega}) \, ds \, d\Omega \, dF. \end{aligned} \quad (3.83)$$

If we knew ψ , then the Monte Carlo approximation of Eq. (3.83) could be written as,

$$\frac{1}{\text{area}(\mathcal{F})} \int_{\mathcal{F}} \int_{\mathbb{S}^2} \psi(\mathbf{r}, \boldsymbol{\Omega}) \, d\Omega \, dF \approx \frac{4\pi}{N} \sum_{n=1}^N \psi(\mathbf{r}^{(n)}, \boldsymbol{\Omega}^{(n)}), \quad (3.84)$$

where $\mathbf{r}^{(n)}$ and $\boldsymbol{\Omega}^{(n)}$ are random variates of uniform random variables in space and angle, respectively. We do not know ψ , so the rest of this section describes how to use the right-hand side of Eq. (3.83) to compute the sum in Eq. (3.84). We use the change of variables in Eq. (3.54) and the PDF in Eq. (3.55) and define the indicator function for crossing \mathcal{F} as,

$$Y(\mathbf{r}', \boldsymbol{\Omega}, s; \mathcal{F}) = \begin{cases} 0 & s < s_d(\mathbf{r}', \boldsymbol{\Omega}; \mathcal{F}), \\ 1 & s_d(\mathbf{r}', \boldsymbol{\Omega}; \mathcal{F}) \geq s, \end{cases} \quad (3.85a)$$

where s_d is the minimum distance to the boundary face \mathcal{F} along $\boldsymbol{\Omega}$,

$$s_d(\mathbf{r}', \boldsymbol{\Omega}; \mathcal{F}) = \min\{s \mid \mathbf{r}' + s\boldsymbol{\Omega} \in \mathcal{F}, \mathcal{F} \in \Gamma_b\}. \quad (3.85b)$$

Fig. 3.4 shows the distance s_d . The final step in this change of variables $\mathbf{r} \rightarrow \mathbf{r}'$ is to relate the surface elements dF and dA , where the dF is a patch on the domain boundary defined in \mathbf{r} coordinates and dA is a patch on the domain boundary defined in \mathbf{r}' coordinates. Using similar arguments as those given in section 3.3.2 where we related the volume elements $d\mathbf{r}$ and $d\mathbf{r}'$, it can be shown that,

$$dF = \frac{1}{|\boldsymbol{\Omega} \cdot \mathbf{n}|} dA. \quad (3.86)$$

Following the same partitioning into volume source and boundary source estimator contributions given in section 3.3.2, and then combining, one can derive the $\hat{\phi}_s$ estimator,

$$\hat{\phi}_s = \frac{2}{\text{area}(\mathcal{F})} \sum_i \frac{w_i}{|\boldsymbol{\Omega}_i \cdot \mathbf{n}|}, \quad (3.87)$$

where the sum in Eq. (3.87) is over all particles which cross the face $\mathcal{F} \in \Gamma_b$. The sum includes both volume source particles and boundary source particles.

3.3.4 Random Variate Generation

We need to generate the following random variates to estimate $\hat{\phi}$ and $\hat{\phi}_s$:

- $\mathbf{r}^{(n)}$ and $\boldsymbol{\Omega}^{(n)}$ of the $\mathbf{R} \sim U(\mathcal{D})$ and $\boldsymbol{\omega} \sim U(\mathbb{S}^2)$ random variables, respectively, for the volume source particles,
- $\mathbf{r}_b^{(m)}$ and $\boldsymbol{\Omega}_h^{(m)}$ of the $\mathbf{R}_b \sim U(\partial\mathcal{D})$ and $\boldsymbol{\omega}_h \sim U(\mathbb{S}_h^2)$ random variables, respectively, for the boundary source particles, and
- $s^{(n)}$ and $s_b^{(m)}$ of the $S \sim \text{exponential}(\sigma_t(\mathbf{R} + s\boldsymbol{\omega}))$ and $S_b \sim \text{exponential}(\sigma_t(\mathbf{R}_b + s\boldsymbol{\omega}_h))$ random variables for the volume source and boundary source particles, respectively.

We begin by assuming the existence of a pseudo random number generator (PRNG) capable of generating random variates of a $X \sim U(0, 1)$ random variable. The tradeoffs of different PRNG implementations and the associated theory are outside the scope of this dissertation,

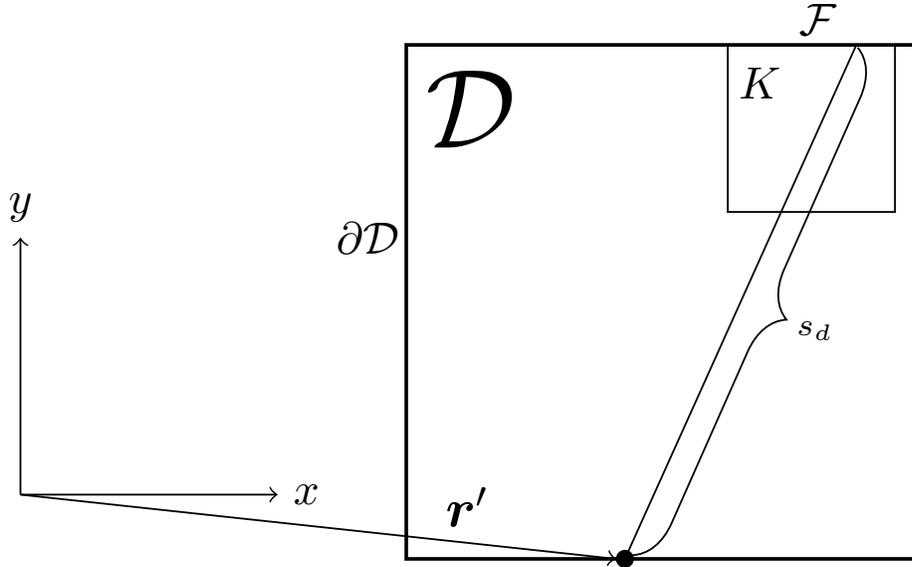


Figure 3.4: Distance to element boundary face s_d .

but can be found in Chapter 2 “Random Numbers” of the monograph by Knuth [73]. For our demonstration of the HSM method, we need two different transformations of $U(0, 1)$ random variates. The first transformation is just scaling and shifting the interval of the $U(0, 1)$ distribution from the unit segment $(0, 1)$ to an arbitrary interval (a, b) . The second transformation is slightly more complicated and involves a function inversion.

Any random variable which is uniform on some interval (a, b) , where a and b are constants such that $a < b$, can be formed by scaling $X \sim U(0, 1)$ by $b - a$ and shifting by a . That is, $Y \sim U(a, b)$ may be written as,

$$Y = Y(X) = (b - a)X + a. \quad (3.88)$$

This is all we need for \mathbf{R} , \mathbf{R}_b , $\boldsymbol{\omega}$, and $\boldsymbol{\omega}_h$. For S and S_b , we will need a slightly more complicated transformation of a $U(0, 1)$ random variable that relies on yet another way of characterizing the distribution of a random variable, the cumulative distribution function (CDF). For a continuous random variable X , the CDF is the function $F(x) = P(X \leq x)$ defined as,

$$F(x) = \int_{-\infty}^x f(x') dx'. \quad (3.89)$$

If $X \sim \text{exponential}(\lambda)$ then the CDF of X is,

$$F(x) = \begin{cases} 0 & x \leq 0, \\ 1 - e^{-\lambda x} & x > 0. \end{cases} \quad (3.90)$$

The inverse of Eq. (3.90) is then,

$$F^{-1}(u) = -\frac{\ln(u)}{\lambda}, \quad (3.91)$$

where $0 < u < 1$ is a random variate of a $U(0, 1)$ random variable and where we used the fact that $(1 - X) \sim U(0, 1)$ if $X \sim U(0, 1)$ to simplify the numerator from $\ln(1 - u)$ to $\ln(u)$ in Eq. (3.91). A random variate x of $X \sim \text{exponential}(\lambda)$ is now computable as,

$$x \leftarrow -\frac{\ln(u)}{\lambda}. \quad (3.92)$$

This transformation is sometimes called the ‘‘inverse-CDF’’ technique, and Eq. (3.92) is an example of the inverse-CDF technique applied to the generation of random variates of an exponential random variable. The inverse-CDF technique relies on the existence of a $U(0, 1)$ PRNG and the invertibility of the CDF.

With Eqs. (3.88) and (3.92), we are now ready to describe the random variate generation techniques for the $\mathbf{R}, \mathbf{R}_b, \boldsymbol{\omega}, \boldsymbol{\omega}_h, S,$ and S_b random variables. Consider first the random variables \mathbf{R} and \mathbf{R}_b for particle positions. The positions in the volume $\mathbf{r}^{(n)}$ are random variates of $\mathbf{R} \sim U(\mathcal{D})$, and the positions on the boundary $\mathbf{r}_b^{(m)}$ are random variates of $\mathbf{R}_b \sim U(\partial\mathcal{D})$. If $\mathcal{D} = [0, 1]^2$, then $\mathbf{R} = (R_x, R_y)$ where $R_x \sim R_y \sim U(0, 1)$ and $\mathbf{R}_b = (R_{bx}, R_{by})$ where,

$$R_{bx} = \begin{cases} 0 & x = 0, \\ 1 & x = 1, \\ U(0, 1) & y = 0, \\ U(0, 1) & y = 1. \end{cases} \quad R_{by} = \begin{cases} U(0, 1) & x = 0, \\ U(0, 1) & x = 1, \\ 0 & y = 0, \\ 1 & y = 1. \end{cases} \quad (3.93)$$

The directions $\boldsymbol{\Omega}^{(n)}$ of particles sourced in the volume are random variates of $\boldsymbol{\omega} \sim U(\mathbb{S}^2)$. Let $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)$ and $\omega_x = \sin \Theta \cos \Phi$, $\omega_y = \sin \Theta \sin \Phi$, and $\omega_z = \cos \Theta$. Then,

$$\Theta = \cos^{-1} \left(U(-1, 1) \right), \quad (3.94a)$$

$$\Phi = U(0, 2\pi). \quad (3.94b)$$

It may seem odd that Eq. (3.94a) first generates random $\cos \theta$ variates on $[-1, 1]$ and then inverts cosine when one could omit the inverse and simply generate θ variates on $[0, \pi]$. No inverse is required for generating random azimuthal angle variates, as can be seen in Eq. (3.94b), which defines the azimuthal angle variates as $U(0, 2\pi)$ variates. The reason for this extra step is that if we generated θ variates on $[0, \pi]$, we would get undesirable clustering of our $\boldsymbol{\Omega}^{(n)}$ variates at the poles of the unit sphere (see Fig. 3.5). This is because the Jacobian determinant relating the area of a patch in Cartesian coordinates to the area of a patch on the surface of the unit sphere is nonlinear in the polar angle θ . Let a be some radius, then

$\boldsymbol{\Omega} = \boldsymbol{\Omega}(\theta, \phi, a) = (a \sin \theta \cos \phi, a \sin \theta \sin \phi, a \cos \theta) = (\Omega_1, \Omega_2, \Omega_3)$. Let $\mathbf{x} = (x_1, x_2, x_3)$ be Cartesian coordinates. The Jacobian matrix is $J_{ij} = \partial \Omega_i / \partial x_j$ and $\det J = a^2 \sin \theta$ therefore $d\Omega = a^2 \sin \theta d\theta d\phi da = \sin \theta d\theta d\phi$ because $a = 1$ for the unit sphere. The function $\sin \theta$ is nonlinear in θ .

The directions $\boldsymbol{\Omega}_h^{(m)}$ of particles sourced on the boundary are random variates of $\boldsymbol{\omega}_b \sim U(\boldsymbol{\Omega} \cdot \mathbf{n} < 0)$. The inequality $\boldsymbol{\Omega} \cdot \mathbf{n} < 0$ defines the hemisphere of domain-inward directions. If $\mathcal{D} = [0, 1]^2$, then $\boldsymbol{\omega}_b$ is defined by the azimuthal half-domains,

$$\Phi = \begin{cases} U(-\frac{\pi}{2}, \frac{\pi}{2}) & x = 0, \\ U(\frac{\pi}{2}, \frac{3\pi}{2}) & x = 1, \\ U(0, \pi) & y = 0, \\ U(\pi, 2\pi) & y = 1. \end{cases} \quad (3.95)$$

The volume source particle distances $s^{(n)}$ and boundary source particle distances $s_b^{(m)}$ are random variates of the random variables,

$$S \sim \text{exponential}(\sigma_t(r^{(n)} + s\boldsymbol{\Omega}^{(n)})), \quad (3.96a)$$

and

$$S_b \sim \text{exponential}(\sigma_t(r_b^{(m)} + s\boldsymbol{\Omega}_h^{(m)})), \quad (3.96b)$$

respectively. In Eqs. (3.96a) and (3.96b), I have made the following substitutions:

$$\mathbf{R} \leftarrow r^{(n)}, \quad (3.97a)$$

and

$$\boldsymbol{\omega} \leftarrow \boldsymbol{\Omega}^{(n)}, \quad (3.97b)$$

as well as

$$\mathbf{R}_b \leftarrow r_b^{(m)}, \quad (3.97c)$$

and

$$\boldsymbol{\omega}_h \leftarrow \boldsymbol{\Omega}_h^{(m)}. \quad (3.97d)$$

The substitutions Eqs. (3.97a) to (3.97d) in Eqs. (3.96a) and (3.96b) emphasize that we generate random variates for the position and direction before generating random variates for the distances. This is because the distance that a photon travels before it collides with an electron in the matter depends on the position where it starts and the direction that it is traveling.

The technique for generating these $s^{(n)}$ and $s_b^{(m)}$ random variates is to express S and S_b as piecewise functions of constant-rate exponential random variables $S_i \sim \text{exponential}(\sigma_i)$, which have PDF,

$$p_i(s) = \sigma_i e^{-\sigma_i s}, \quad s \geq 0. \quad (3.98)$$

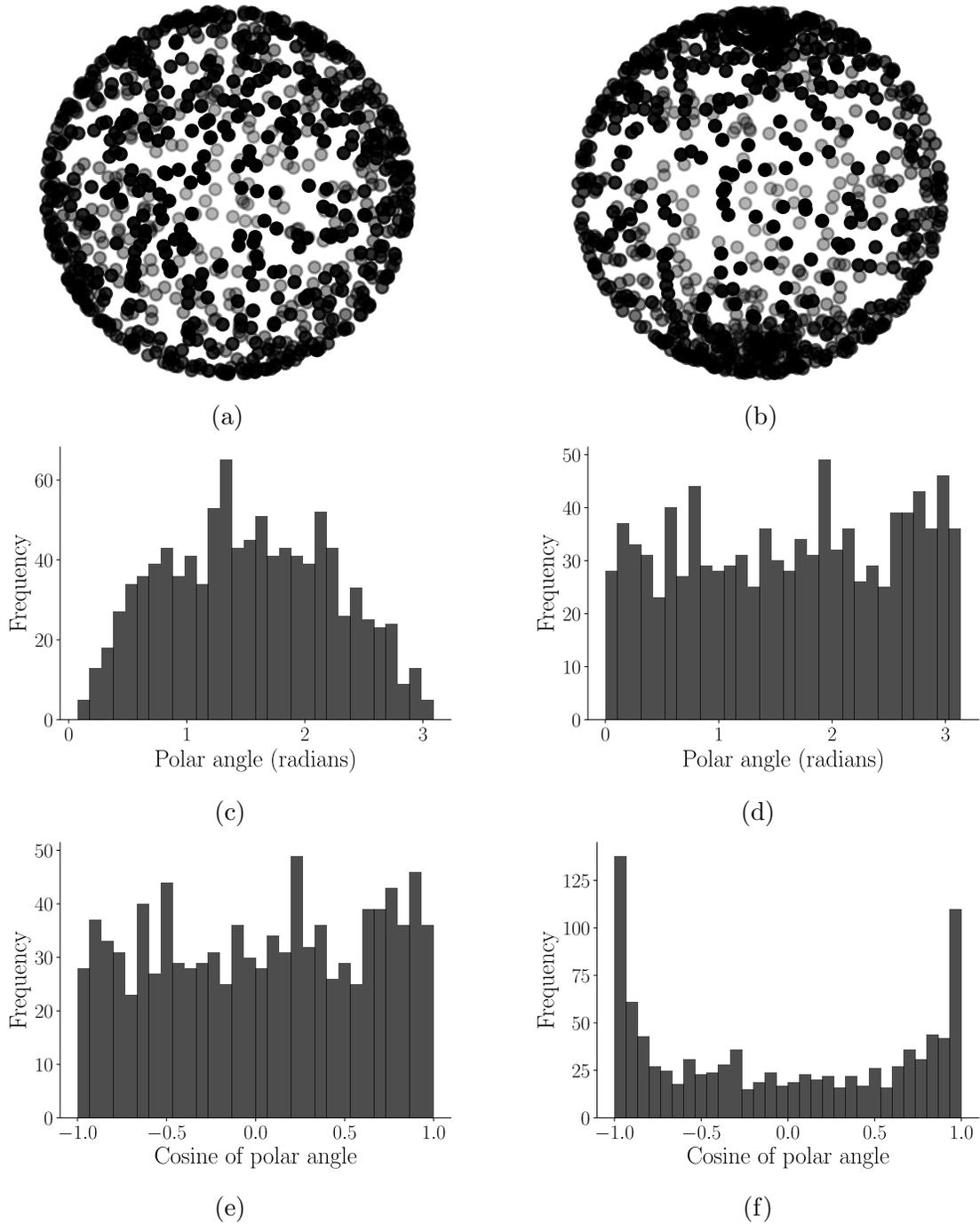


Figure 3.5: One thousand random variates generated from a random variable that is uniform in direction (left column) versus uniform in polar angle (right column). The histogram in (f) shows that the points on the sphere in (b) are highly clustered at the poles. The histograms in (d) and (e) have the same bin heights because I used the same seed to generate the 1000 variates plotted in the left column and the 1000 variates plotted in the right column.

The i subscript in Eq. (3.98) denotes element i along the MC particle direction of travel in the mesh. The σ_i in Eq. (3.98) is the value of σ_t in K_i , where we have assumed that σ_t is piecewise constant and that all discontinuities in σ_t align with mesh boundaries. Let the path lengths \hat{s}_i , $i = 1, \dots, L$ correspond to element face intersection points along the ray defined by position \mathbf{r}' and direction $\mathbf{\Omega}$,

$$\mathbf{r}' + \hat{s}_i \mathbf{\Omega} \in K_i \cap K_{i+1}. \quad (3.99)$$

Fig. 3.6 shows an example where $i \in \{1, 2, 3, 4, 5\}$ because the MC particle traverses five elements on a uniform 3-by-3 mesh of a square domain. With Eqs. (3.98) and (3.99), the random variable $S \sim \text{exponential}(\sigma_t(\mathbf{r}' + s\mathbf{\Omega}))$ for any \mathbf{r}' and $\mathbf{\Omega}$ can now be defined as,

$$S = \begin{cases} S_1 & S_1 < \hat{s}_1, \\ \hat{s}_1 + S_2 & S_1 > \hat{s}_1 \text{ and } \hat{s}_1 + S_2 < \hat{s}_2, \\ \hat{s}_2 + S_3 & \hat{s}_1 + S_2 > \hat{s}_2 \text{ and } \hat{s}_2 + S_3 < \hat{s}_3, \\ \vdots & \vdots \\ \hat{s}_{L-1} + S_L & \hat{s}_{L-2} + S_{L-1} > \hat{s}_{L-1} \text{ and } \hat{s}_{L-1} + S_L < \hat{s}_L, \end{cases} \quad (3.100)$$

and its PDF as,

$$P(\hat{s}_{i-1} \leq S \leq \hat{s}_i) = P(S_i \leq \hat{s}_i - \hat{s}_{i-1}) \prod_{k=1}^{i-1} P(S_k \geq \hat{s}_k). \quad (3.101)$$

Every capital “ S ” in Eqs. (3.100) and (3.101) with a subscript, such as S_i , is an exponential random variable with constant rate parameter σ_i , and PDF defined by Eq. (3.98). Thus, $S_i \sim \text{exponential}(\sigma_i)$, $S_1 \sim \text{exponential}(\sigma_1)$, $S_L \sim \text{exponential}(\sigma_L)$, $S_k \sim \text{exponential}(\sigma_k)$, etc. The factor in front of the product in Eq. (3.101) is just the PDF of $S_i \sim \text{exponential}(\sigma_i)$. The factors in the product are called survival functions. The survival function of a random variable X is $P(X \geq x)$ and is equal to $1 - P(X \leq x) = 1 - F(x)$ where $F(x)$ is the CDF of X . Substituting the PDF from Eq. (3.98) and the CDF from Eq. (3.90) into Eq. (3.101) gives,

$$P(\hat{s}_{i-1} \leq S \leq \hat{s}_i) = (1 - e^{-\sigma_i s}) \prod_{k=1}^{i-1} e^{-\sigma_k \hat{s}_k}. \quad (3.102)$$

Thus, generating $s^{(n)}$ and $s_b^{(m)}$ random variates is equivalent to sampling the exponential distribution defined in the element K_0 , then K_1 , then K_2 , and so on until the particle is absorbed in K_L or escapes the domain boundary.

3.3.5 Summary of Fixed Source Integrals

There are three sources in Eqs. (1.13a) and (1.13b). The scattering source, which is the product of $\sigma_s/(4\pi)$ and φ , varies during the calculation because φ changes with every cycle

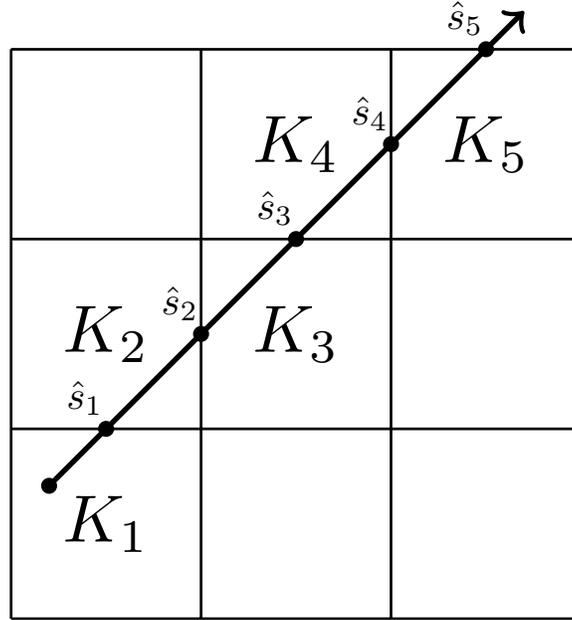


Figure 3.6: A particle sourced in the lower-left of a square domain with a north-easterly direction traverses five elements in a uniform 3-by-3 mesh before escaping out the top boundary.

of the HSM iteration. The volume source q and the boundary source $\bar{\psi}$, which we have alternately referred to as ψ_{inc} in this chapter, are both “fixed” in the sense that they do not vary during the calculation. This section summarizes the integrals that we compute during sourcing of MC particles in the volume of \mathcal{D} and on its boundary $\partial\mathcal{D}$. The process of creating the MC particles by sampling their positions and directions and assigning their weights, which is called “sourcing”, is equivalent to Monte Carlo integration of the following integrals. The description that follows summarizes the aforementioned sourcing process.

The fixed source $q = q(\mathbf{x}, \boldsymbol{\Omega})$ is a volume source. Our method for assigning MC particle weights to the volume source particles satisfies,

$$\lim_{N \rightarrow \infty} \sum_{i=1}^N w_i = \int_{\mathcal{D}} \int_{\mathbb{S}^2} q \, d\Omega \, d\mathbf{x}, \quad (3.103a)$$

where N is the number of MC particles sourced in the volume \mathcal{D} , and

$$w_i = \frac{V}{N} q(x_i, y_i, z_i, \theta_i, \phi_i), \quad (3.103b)$$

with

$$V = \int_{\mathcal{D}} \int_{\mathbb{S}^2} d\Omega \, d\mathbf{x}, \quad (3.103c)$$

defines the weight of particle i . Let $U(a, b)$ be a uniformly-distributed random variate on $[a, b]$. We use Monte Carlo to integrate,

$$\int_{\mathcal{D}} \int_{\mathbb{S}^2} q \, d\Omega \, d\mathbf{x} \approx \frac{V}{N} \sum_{i=1}^N q(x_i, y_i, z_i, \theta_i, \phi_i), \quad (3.103d)$$

$$x_i \leftarrow U(x_{\min}, x_{\max}), \quad y_i \leftarrow U(y_{\min}, y_{\max}), \quad (3.103e)$$

$$z_i \leftarrow U(z_{\min}, z_{\max}), \quad \theta_i \leftarrow \cos^{-1}(U(-1, 1)), \quad \phi_i \leftarrow U(0, 2\pi), \quad (3.103f)$$

where V is defined by Eq. (3.103c), and the assumption that \mathcal{D} is a rectangular prism allowed us to write the position as random variates of uniform distributions. I made the same rectangularity assumption in section 3.3.4 Random Variate Generation, where I let $\mathcal{D} = [0, 1]^2$. Position variate generation in non-rectangular domains can be achieved using acceptance-rejection with an inside-outside indicator function, which determines whether a point lies inside or outside the region of definition of the volume source. Points generated outside the desired region are discarded.

The fixed source $\bar{\psi}(\mathbf{x}, \boldsymbol{\Omega})$ is a boundary source. Our method for assigning MC particle weights to the boundary source particles satisfies,

$$\lim_{M \rightarrow \infty} \sum_{i=1}^M w_i = \int_{\partial\mathcal{D}} \int_{\boldsymbol{\Omega} \cdot \mathbf{n} < 0} |\boldsymbol{\Omega} \cdot \mathbf{n}| \bar{\psi} \, d\Omega \, d\mathbf{x}, \quad (3.104a)$$

where M is the number of MC particles sourced on the surface $\partial\mathcal{D}$, and

$$w_i = \frac{S}{M} |\boldsymbol{\Omega}_i \cdot \mathbf{n}| \bar{\psi}(x_i, y_i, z_i, \theta_i, \phi_i), \quad (3.104b)$$

with

$$S = \int_{\partial\mathcal{D}} \int_{\boldsymbol{\Omega} \cdot \mathbf{n} < 0} d\Omega \, d\mathbf{x}, \quad (3.104c)$$

defines the weight of particle i . We use Monte Carlo to integrate,

$$\int_{\partial\mathcal{D}} \int_{\boldsymbol{\Omega} \cdot \mathbf{n} < 0} |\boldsymbol{\Omega} \cdot \mathbf{n}| \bar{\psi} \, d\Omega \, d\mathbf{x} \approx \frac{S}{M} \sum_{i=1}^M |\boldsymbol{\Omega}_i \cdot \mathbf{n}| \bar{\psi}(x_i, y_i, z_i, \theta_i, \phi_i), \quad (3.104d)$$

where S is defined by Eq. (3.104c), the positions \mathbf{x}_i are sampled uniformly on $\partial\mathcal{D}$, and the directions $\boldsymbol{\Omega}_i$ are sampled uniformly on the hemisphere of the unit sphere defined by $\boldsymbol{\Omega} \cdot \mathbf{n} < 0$.

3.3.6 Monte Carlo Transport Algorithm

Algorithm 2 shows the definition of a function which implements the Monte Carlo solver described in this chapter. The aim of Algorithm 2 is to clearly communicate the main logic. Its style and especially its concision may lead to misunderstandings, which I attempt to

address in the following explanation of its logic. Algorithm 2 is important because it defines the logic inside the `mc()` call in the HSM algorithm that I will present in Algorithm 3.

My pseudocode in Algorithm 2 uses a vector arithmetic syntax inspired by NumPy⁵ [74]. This means that statements in Algorithm 2 manipulate arrays and use masking to operate on array subsets. Vector arithmetic syntax is often concise. For example, the statement $\mathbf{c} = \mathbf{a} + \mathbf{b}$, which describes entry-by-entry addition of vectors \mathbf{a} and \mathbf{b} , is more concise than the C-style indexing statement `for(int i = 0; i < N; i++) c[i] = a[i] + b[i]` because the loop is implied in the vector arithmetic syntax. Vector arithmetic can also expose parallelism that maps well to computer architectures with vector hardware.

In the description that follows, a sequence of “segments” constitutes a Monte Carlo particle “history”. The history depicted in Fig. 3.6 has five segments. The first four are “facet crossings”, meaning that the particle crossed into a new mesh element, and the final segment is an “escape”, meaning that the particle crossed the domain boundary. Another segment that appears in the histories is a “collision”, meaning that the photon represented by the MC particle was absorbed due to an interaction with an electron in the matter.

My MC solver style is based on the event-based Monte Carlo algorithm [75]. This means that every iteration of the loop processes one segment for every particle, in contrast to history-based Monte Carlo, which processes an entire particle history before proceeding to the next particle. Event-based MC can outperform history-based MC on graphics processing unit (GPU) architectures with low kernel launch latencies because the event-based MC has smaller kernels. A small kernel can have higher occupancy⁶ than a large kernel if the large kernel has so much code that it exhausts the number of registers per thread permitted by the GPU instruction set architecture (ISA), as often happens for history-based MC implementations. Event-based kernels have higher occupancy than the equivalent history-based kernel, thus event-based MC can provide the latency hiding required to progress from memory-latency boundedness to memory-bandwidth boundedness [76]. Implementing memory-bandwidth bound algorithms is the dominant strategy by which we can achieve good performance in many scientific applications.

The most important variable elided from Algorithm 2 is the mesh. The mesh would include a description of the element geometry, material properties, and topology. All are required for determining which element a particle is sourced into, distances to ray-surface intersection points, distances to collisions between the photons and the matter, and the piecewise constant domain of definition for the computed estimators. The pervasiveness of the mesh means that, if I had chosen to include it in my pseudocode, it would appear as an argument in many of the function calls in Algorithm 2.

Line 1 shows the function name and function parameters, which have different types: “vs” and “bs” are descriptions of the volume source and boundary source, respectively.

⁵My adoption of NumPy syntax is not an endorsement of Python for Monte Carlo solver implementations.

⁶On an Nvidia GPU, occupancy is a number between 0 and 1 which describes the fraction of the maximum number of warps that can run simultaneously on a streaming multiprocessor (SM). If a kernel achieved 0.5 occupancy then it ran its logic on an SM at half the hardware-defined warp capacity. A warp is 32 threads. An SM is roughly equivalent to a central processing unit (CPU) core. Higher occupancy is better.

A sensible choice for the type of vs and bs would be functions with arbitrary space- and angle-dependence, which would accommodate the $q(\mathbf{x}, \boldsymbol{\Omega})$ or $Q(\mathbf{x}, \boldsymbol{\Omega})$ volume source and the $\bar{\psi}(\mathbf{x}, \boldsymbol{\Omega})$ inflow boundary source. The “scattering_events” parameter is of Boolean⁷ type. Lines 2-3 sample the sources according to Eqs. (3.103d) and (3.104d). Important variables elided from Algorithm 2 that are required during sourcing are the user-defined number of particles to source on the volume and the boundary, which I designated as N and M in Eqs. (3.68) and (3.78), respectively.

Line 4 creates estimator accumulators called “tallies” and initializes them to zero. Line 5 defines the loop conditional to prevent termination until all particles have been absorbed or escaped the domain boundary. In practice, it can be useful to augment the conditional to check that a segment counter, which increments every loop iteration, is less than some user-defined threshold to ensure loop termination.

Lines 6-9 determine which particles cross their elements without collision and which collide with the matter. Lines 10-11 move the particles, thereby generating terms in the estimator sums, which we accumulate. Line 12 checks whether the Monte Carlo solve includes scattering events. The MC solve would exclude scattering events if the linear transport being solved omits the scattering term or if the contribution of the scattering term is included in the volume source, as is the case for HSM. Lines 13-19 determine which particles undergo scattering events and which particles undergo absorption events. A scattering event changes the direction of a particle and an absorption event un-alives a particle. Line 21 un-alives the particles because we determined in the conditional that the particles cannot have scattering events, so the only event they undergo is absorption. Line 24 returns the tallies. Thus, the output of the MC solve is the accumulated estimators.

3.4 Monte Carlo Integration of Second Moment Method

Integrating the SMM using Monte Carlo requires augmenting the Monte Carlo transport technique described in this chapter with two additional considerations: incorporation of information from the moment solve into the Monte Carlo solve, and calculation of information in the Monte Carlo solve for use in the moment solve. The former task requires using Monte Carlo to integrate the scattering source, which I describe in section 3.4.1, and the latter requires estimating the SMM data, which I describe in sections 3.4.2 and 3.4.3.

3.4.1 Derivation of Scattering Source Integral

The method for sampling the scattering source resembles that of the fixed source q except we replace q in Eqs. (3.103a) and (3.103d) with the product of $\sigma_s/(4\pi)$ and φ . The HSM iteration, which I will describe in Chapter 4, involves calling Algorithm 2 in a loop with φ

⁷George Boole (1815-1864) was an English logician.

Algorithm 2 Monte Carlo Integration of Linear Transport

```

1: function MC(vs, scattering_events, bs)
2:   particles  $\leftarrow$  sample_volume_source(vs)
3:   particles  $\leftarrow$  particles + sample_boundary_source(bs)
4:   tallies  $\leftarrow$  0
5:   while any(particles.alive) do
6:     intersection_distances  $\leftarrow$  compute_distances(particles)
7:     collision_distances  $\leftarrow$  sample_distances(particles)
8:     intersection_mask  $\leftarrow$  intersection_distances < collision_distances
9:     collision_mask  $\leftarrow$  intersection_distances > collision_distances
10:    tallies  $\leftarrow$  tallies + move_across_element_boundary(particles[intersection_mask])
11:    tallies  $\leftarrow$  tallies + move_to_collision_location(particles[collision_mask])
12:    if scattering_events then
13:      num_random_variates  $\leftarrow$  count(particles[collision_mask])
14:      random_variates  $\leftarrow$  U(0, 1, num_random_variates)
15:      scattering_mask  $\leftarrow$  random_variates <  $\sigma_s/\sigma_t$ 
16:      new_directions  $\leftarrow$  sample_direction(particles[collision_mask][scattering_mask])
17:      particles[collision_mask][scattering_mask].direction  $\leftarrow$  new_directions
18:      absorption_mask  $\leftarrow$  random_variates >  $\sigma_s/\sigma_t$ 
19:      particles[collision_mask][absorption_mask].alive  $\leftarrow$  false
20:    else
21:      particles[collision_mask].alive  $\leftarrow$  false
22:    end if
23:  end while
24:  return tallies
25: end function

```

iterates that differ every cycle, which means that we need to recompute the particle weights in every MC solve. The other parameters of the particles, which include the position and direction random variates, remain unchanged. We thus reset the pseudo-random number generator seed every cycle of the HSM iteration, which means that MC particles are sourced with the same position and direction every cycle, but with different weights, because $\varphi^{(i)} \neq \varphi^{(i-1)}$, where i is the cycle index of the HSM iteration. Fig. 3.7 shows that the HSM iteration converges when we reset the seed (“Seed reset”) and that the iteration does not converge if we do not reset the seed (“No seed reset”).

Averaging is an alternative to resetting the seed that allows for more sampling during the iteration. The averaging procedure resembles the MC eigenvalue iteration process for neutronics criticality calculations in which eigenvalue estimates during a user-specified number of “inactive” cycles are discarded, and estimates during the subsequent “active” cycles are averaged, until a convergence criterion involving the average is satisfied. Averaging outperformed seed resetting in the iterated Monte Carlo neutronics method of Pasmann [35].

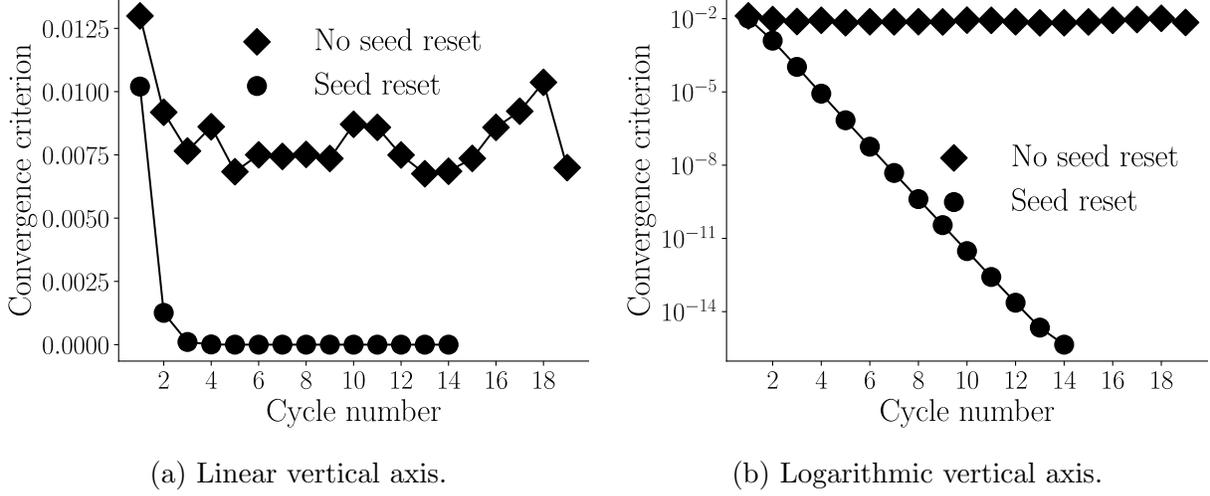


Figure 3.7: Resetting the pseudo-random number generator seed makes the HSM iteration converge to machine precision.

3.4.2 Derivation of $\hat{\mathbf{T}}$ Estimator

The HSM method uses Monte Carlo to estimate the SMM correction tensor \mathbf{T} . We defined \mathbf{T} in Eq. (1.42). We can rewrite Eq. (1.42) using the definitions in Eqs. (1.17) and (1.19) as,

$$\mathbf{T} = \mathbf{P} - \frac{1}{3}\mathbf{I}\phi. \quad (3.105)$$

Also recall that we derived an estimator for ϕ that we called $\hat{\phi}$ in section 3.3.2. Multiplying the characteristic Eq. (3.49) by $\boldsymbol{\Omega} \otimes \boldsymbol{\Omega}$ and proceeding with the derivation will give an estimator for \mathbf{P} instead of ϕ ,

$$\hat{\mathbf{P}} = \frac{1}{\text{vol}(K)} \sum_{i=1} \boldsymbol{\Omega}_i \otimes \boldsymbol{\Omega}_i w_i d_i, \quad (3.106)$$

where the sum in Eq. (3.106) is over all paths traversed by all particles in element K . The sum includes both volume source particles and boundary source particles. Thus, we can estimate the correction tensor \mathbf{T} as,

$$\hat{\mathbf{T}} = \hat{\mathbf{P}} - \frac{1}{3}\mathbf{I}\hat{\phi}. \quad (3.107)$$

3.4.3 Derivation of $\hat{\beta}$ Estimator

The HSM method also uses Monte Carlo to estimate the SMM boundary correction factor β . Our $\hat{\mathbf{T}}$ estimator for \mathbf{T} defined in Eq. (3.107) is a volume tally. Our estimator for β will

be a boundary surface tally. We defined β in Eq. (1.45). We can rewrite Eq. (1.45) using the definitions in Eqs. (1.17) and (1.25) as,

$$\beta = B - \frac{1}{2}\phi_s, \quad (3.108)$$

where the subscript s on ϕ in Eq. (3.108) denotes that the domain of definition is a surface rather than a volume. Eq. (3.87) defines $\hat{\phi}_s$, which is the estimator for ϕ_s that we derived in section 3.3.3. If we multiply the characteristic Eq. (3.49) by $|\boldsymbol{\Omega} \cdot \mathbf{n}|$, integrate over all directions, then average over an arbitrary boundary face $\mathcal{F} \in \Gamma_b$, we will obtain Eq. (3.83) precisely, but with an additional factor of $|\boldsymbol{\Omega} \cdot \mathbf{n}|$ in the integrands. Proceeding with the derivation will result in Eq. (3.87) for $\hat{\phi}_s$, but with an additional factor of $|\boldsymbol{\Omega} \cdot \mathbf{n}|$, which will cancel the one in the denominator of Eq. (3.87). The result is,

$$\hat{B} = \frac{2}{\text{area}(\mathcal{F})} \sum_i w_i, \quad (3.109)$$

where the sum in Eq. (3.109) is over all particles which cross the face $\mathcal{F} \in \Gamma_b$. The sum includes both volume source particles and boundary source particles. Thus, we can estimate the boundary correction factor β as,

$$\hat{\beta} = \hat{B} - \frac{1}{2}\hat{\phi}_s. \quad (3.110)$$

3.5 The Central Limit Theorem and Transport Estimators

This section addresses the question of whether it is appropriate to apply the CLT to the Monte Carlo transport estimators derived in the previous sections. First, recall section 3.1.3 Probability Distribution Moment Estimation, where we defined the sample mean \bar{X} , which is an estimator of μ . In section 3.1.4 Central Limit Theorem, we showed that the uncertainty of the estimator \bar{X} is $\sqrt{\text{Var}[\cdot]/N}$, where $\text{Var}[\cdot]$ is the variance of the estimator and N is the number of random variables X_i , $i \in \{1, \dots, N\}$ in Eq. (3.20). This expression for the uncertainty, which comes from the CLT, requires that the X_i are iid with finite mean and finite variance. Are the random variables which appear in the Monte Carlo transport estimators also iid with finite mean and finite variance?

In section 3.3 Monte Carlo Integration of Linear Transport, we defined the estimators $\hat{\phi}$ and $\hat{\phi}_s$. In section 3.4 Monte Carlo Integration of Second Moment Method, we defined the estimators $\hat{\mathbf{P}}$ and \hat{B} and used them with $\hat{\phi}$ and $\hat{\phi}_s$ to define estimators $\hat{\mathbf{T}} = \hat{\mathbf{P}} - (1/3)\mathbf{I}\hat{\phi}$ and $\hat{\beta} = \hat{B} - (1/2)\hat{\phi}_s$ for the SMM data. All of these estimators are sums corresponding to Monte Carlo integration of the expectation of $p(\mathbf{r}', \boldsymbol{\Omega}, s)$ and $\rho(\mathbf{r}', \boldsymbol{\Omega}, s)$, which are defined in Eqs. (3.63) and (3.73), respectively. The function $p(\mathbf{r}', \boldsymbol{\Omega}, s)$ is a function of the random variables,

$$\mathbf{R} \sim U(\mathcal{D}), \quad (3.111a)$$

$$\boldsymbol{\omega} \sim U(\mathbb{S}^2), \quad (3.111b)$$

$$S \sim \text{exponential}(\sigma_t(\mathbf{R} + s\boldsymbol{\omega})). \quad (3.111c)$$

The function $\rho(\mathbf{r}', \boldsymbol{\Omega}, s)$ is a function of the random variables,

$$\mathbf{R}_b \sim U(\partial\mathcal{D}), \quad (3.112a)$$

$$\boldsymbol{\omega}_h \sim U(\mathbb{S}_h^2), \quad (3.112b)$$

$$S_b \sim \text{exponential}(\sigma_t(\mathbf{R}_b + s\boldsymbol{\omega}_h)). \quad (3.112c)$$

The sums in the estimators are sums over:

- N iid random variables \mathbf{R}_i for $i \in \{1, \dots, N\}$, and
- N iid random variables $\boldsymbol{\omega}_i$ for $i \in \{1, \dots, N\}$, and
- N iid random variables S_i for $i \in \{1, \dots, N\}$, and
- M iid random variables \mathbf{R}_b^j for $j \in \{1, \dots, M\}$, and
- M iid random variables $\boldsymbol{\omega}_h^j$ for $j \in \{1, \dots, M\}$, and
- M iid random variables S_b^j for $j \in \{1, \dots, M\}$, because

we use N particles to integrate the volume source and we use M particles to integrate the boundary source. Note that the S_i are independent of each other, even though S_i depends on \mathbf{R}_i and $\boldsymbol{\omega}_i$. The same is true of the S_b^j , which depend on \mathbf{R}_b^j and $\boldsymbol{\omega}_h^j$.

Since the random variables are iid, the only remaining constraint they must satisfy for the CLT to apply to the estimators is that the mean and variance of the random variables must be finite. As shown in Eqs. (3.111a) to (3.111c) and Eqs. (3.112a) to (3.112c), the two distributions which characterize the random variables are the uniform distribution and the exponential distribution. We will consider these distributions separately.

First, consider the uniform random variable $X \sim U(a, b)$. The mean is $0.5(a + b)$ and the variance is $(1/12)(a + b)^2$, which indicates that the mean and the variance of the uniform distribution are finite as long as the parameters a and b are also finite. The parameters in Eqs. (3.111a), (3.111b), (3.112a) and (3.112b) are $\mathcal{D}, \mathbb{S}^2, \partial\mathcal{D}$, and \mathbb{S}_h^2 , respectively. The unit sphere and its hemisphere are clearly finite. That leaves \mathcal{D} and $\partial\mathcal{D}$, which are finite if the problem definition specifies a finite domain and a finite domain boundary.

Now consider the exponential random variable $X \sim \text{exponential}(\lambda(x))$, where $\lambda(x)$ is the non-constant rate parameter. Its PDF is Eq. (3.13) and its mean is Eq. (3.19). Its variance is $E[f^2] - (E[f])^2$, where f is the PDF of X defined by Eq. (3.13). The mean and variance of X are finite if the domain of f is finite and the rate parameter $\lambda(x)$ is finite. The former is true if \mathcal{D} is finite. The latter is true for radiative transfer because the rate parameter is the total cross section σ_t which we can assume is finite because the physical quantities on which it depends are all finite.

Thus, we conclude that the random variables have finite mean and finite variance as long as the domain \mathcal{D} and its boundary $\partial\mathcal{D}$ are finite.

In summary, the random variables associated with the Monte Carlo transport estimators are iid. Additionally, they have finite mean and finite variance, as long as the domain \mathcal{D} pertaining to the particular problem under study and its boundary $\partial\mathcal{D}$ are both finite. Therefore, the requirement that must be satisfied for the CLT to apply to the estimators is:

$$\text{vol}(\mathcal{D}) < \infty, \quad (3.113a)$$

and

$$\text{area}(\partial\mathcal{D}) < \infty. \quad (3.113b)$$

Consider, for example, a linear transport problem on the unit square, $\mathcal{D} = [0, 1]^2$. This domain and its boundary satisfy Eqs. (3.113a) and (3.113b). Consider using the Monte Carlo transport method with N particles to solve this linear transport problem. By the CLT, the uncertainty of the Monte Carlo estimator for the solution would be $\sqrt{\text{Var}[\cdot]/N}$, where $\text{Var}[\cdot]$ is the variance of the estimator and N is the number of MC particles.

Many transport problems on infinite domains can be accurately approximated as transport problems on finite domains using an appropriate boundary condition. Some infinite-domain problems, such as the ones that follow, can be modeled exactly on a finite domain. Consider an “infinite medium” problem, which has infinite domain, thus violating Eqs. (3.113a) and (3.113b). Assume that the matter is homogeneous and that the fixed source does not have spatial dependence. The problem can be modeled exactly on a cube of finite extent with “reflecting” boundary conditions. Any MC particle which hits the domain boundary (i.e. a face of the cube) has its direction $\mathbf{\Omega}$ negated with respect to the boundary surface normal vector \mathbf{n} (i.e. $\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z, -\mathbf{e}_x, -\mathbf{e}_y,$ or $-\mathbf{e}_z$ for an axis-aligned cube).

A “semi-infinite medium” is infinite in extent along one or two spatial axes, but not all three. It also violates Eqs. (3.113a) and (3.113b). However, it can be modeled in the same manner as the infinite medium, except with reflecting boundaries placed only along the axis or axes corresponding to the infinite extent(s), again assuming homogeneity of the matter. The transformed infinite medium, and the transformed semi-infinite medium, are exact models of the original problems, but the transformed problems no longer violate Eqs. (3.113a) and (3.113b).

3.6 Derivation of $\text{Var}[\hat{\mathbf{T}}]$ in the Thick Diffusion Limit

In section 3.5 The Central Limit Theorem and Transport Estimators, we showed that the uncertainty of a Monte Carlo transport estimator is $\sqrt{\text{Var}[\cdot]/N}$, where $\text{Var}[\cdot]$ is the variance of the estimator and N is the number of MC particles. This result follows from the CLT, which applies provided that Eqs. (3.113a) and (3.113b) is satisfied. In section 5.2 Thick Diffusion Limit Problem, I will show that $\text{Var}[\hat{\phi}] = O(1/\epsilon)$ in the thick diffusion limit, which is an analytic result for which I provide supporting empirical evidence in Fig. 5.8.

The $O(1/\epsilon)$ dependence of the variance is undesirable because it means that, for a fixed number of MC particles, the quality of the HSM estimate of the solution to the linear transport Eq. (1.13a) decreases with optical thickness because the uncertainty increases with optical thickness. In this section, I derive $\text{Var}[\hat{\phi}]$ to show that it is $O(1/\epsilon)$ in the TDL for HSM calculations using $\hat{\mathbf{T}}$. My purpose is to motivate section 3.7, in which I derive a lower variance estimator for \mathbf{T} , which I call $\hat{\mathbf{R}}$. In section 3.7.3, I derive $\text{Var}[\hat{\mathbf{R}}]$ to show that $\text{Var}[\hat{\mathbf{R}}] \ll \text{Var}[\hat{\mathbf{T}}]$ in the TDL.

My derivation of $\text{Var}[\hat{\mathbf{T}}]$ begins by deriving $\text{Var}[\hat{\phi}]$. In section 3.3.2 Derivation of $\hat{\phi}$ Estimator, we showed that $\hat{\phi}$ is the sum of the contributions from two estimators for expectations of functions of random variables, Eqs. (3.65) and (3.75). We can derive the variance of $\hat{\phi}$ by deriving the variance of these functions. Functionals of random variables, such as the expectation, can be applied to functions of random variables because a function of one or more random variables is itself a random variable.

The variance of a random variable, or a function of random variables, is the expectation of the square minus the square of the expectation,

$$\text{Var}[f] = E[f^2] - (E[f])^2. \quad (3.114)$$

Recalling Eq. (3.66), which shows the expectation of f along with a volume factor, the expectation of f^2 is,

$$E[f^2] = \int_{\mathcal{D}} \int_{\mathbb{S}^2} \int_0^\infty f^2 p \, ds \, d\Omega \, d\mathbf{r}', \quad (3.115)$$

where p and f are defined by Eqs. (3.63) and (3.65), respectively. Momentarily disregard the volume and angle integrals and consider only the integral along the particle path,

$$\begin{aligned} \int_0^\infty f^2 p \, ds &= \int_0^\infty \left(4\pi \, \text{vol}(\mathcal{D}) \, Q \, \tau\right)^2 \left(\frac{1}{\text{vol}(\mathcal{D})} \frac{1}{4\pi} p_{\sigma_t}\right) ds \\ &= 4\pi \, \text{vol}(\mathcal{D}) \, Q^2 \int_0^\infty \tau^2 p_{\sigma_t} \, ds, \end{aligned} \quad (3.116)$$

where p_{σ_t} and τ are defined by Eqs. (3.55) and (3.56a), respectively. In a single material problem, the rate in the exponential PDF is constant, and Eq. (3.55) simplifies to,

$$p_{\sigma_t}(s) = \sigma_t e^{-\sigma_t s}. \quad (3.117)$$

Eq. (3.117) in the TDL is,

$$p_{\sigma_t}(s) = \frac{\sigma_t}{\epsilon} e^{-\frac{\sigma_t}{\epsilon} s}, \quad (3.118)$$

where ϵ is the TDL scaling parameter described in section 1.3.6. Now consider just the integral in Eq. (3.116) without its coefficient. Substituting Eq. (3.118) for p_{σ_t} and Eq. (3.56a) for τ into the integral gives,

$$\begin{aligned} \int_0^\infty \tau^2 p_{\sigma_t} \, ds &= \int_0^\infty \tau^2 \left(\frac{\sigma_t}{\epsilon} e^{-\frac{\sigma_t}{\epsilon} s}\right) ds \\ &= \int_{s_1}^{s_2} (s - s_1)^2 \frac{\sigma_t}{\epsilon} e^{-\frac{\sigma_t}{\epsilon} s} \, ds + (s_2 - s_1)^2 \int_{s_2}^\infty \frac{\sigma_t}{\epsilon} e^{-\frac{\sigma_t}{\epsilon} s} \, ds. \end{aligned} \quad (3.119)$$

The second integral in Eq. (3.119) is,

$$\begin{aligned} (s_2 - s_1)^2 \int_{s_2}^{\infty} \frac{\sigma_t}{\epsilon} e^{-\frac{\sigma_t}{\epsilon} s} ds &= (s_2 - s_1)^2 \left[-e^{-\frac{\sigma_t}{\epsilon} s} \right]_{s_2}^{\infty} \\ &= (s_2 - s_1)^2 e^{-\frac{\sigma_t}{\epsilon} s_2}. \end{aligned} \quad (3.120)$$

The first integral in Eq. (3.119) may be computed using integration by parts twice. Applying integration by parts once gives,

$$\begin{aligned} \int_{s_1}^{s_2} (s - s_1)^2 \frac{\sigma_t}{\epsilon} e^{-\frac{\sigma_t}{\epsilon} s} ds &= \left[-(s - s_1)^2 e^{-\frac{\sigma_t}{\epsilon} s} \right]_{s_1}^{s_2} + \int_{s_1}^{s_2} 2(s - s_1) \frac{\sigma_t}{\epsilon} e^{-\frac{\sigma_t}{\epsilon} s} ds \\ &= -(s_2 - s_1)^2 e^{-\frac{\sigma_t}{\epsilon} s_2} + \int_{s_1}^{s_2} 2(s - s_1) \frac{\sigma_t}{\epsilon} e^{-\frac{\sigma_t}{\epsilon} s} ds. \end{aligned} \quad (3.121)$$

Applying integration by parts to the integral in Eq. (3.121) gives,

$$\begin{aligned} \int_{s_1}^{s_2} 2(s - s_1) \frac{\sigma_t}{\epsilon} e^{-\frac{\sigma_t}{\epsilon} s} ds &= \left[2(s - s_1) \frac{-\epsilon}{\sigma_t} e^{-\frac{\sigma_t}{\epsilon} s} \right]_{s_1}^{s_2} - \int_{s_1}^{s_2} 2 \frac{-\epsilon}{\sigma_t} e^{-\frac{\sigma_t}{\epsilon} s} ds \\ &= 2(s_1 - s_2) \frac{\epsilon}{\sigma_t} e^{-\frac{\sigma_t}{\epsilon} s_2} + 2 \frac{\epsilon}{\sigma_t} \left[-\frac{\epsilon}{\sigma_t} e^{-\frac{\sigma_t}{\epsilon} s} \right]_{s_1}^{s_2} \\ &= 2(s_1 - s_2) \frac{\epsilon}{\sigma_t} e^{-\frac{\sigma_t}{\epsilon} s_2} + 2 \frac{\epsilon}{\sigma_t} \left(\frac{\epsilon}{\sigma_t} e^{-\frac{\sigma_t}{\epsilon} s_1} - \frac{\epsilon}{\sigma_t} e^{-\frac{\sigma_t}{\epsilon} s_2} \right) \\ &= 2 \left(\frac{\epsilon}{\sigma_t} \right)^2 e^{-\frac{\sigma_t}{\epsilon} s_1} + \left\{ 2s_1 \left(\frac{\epsilon}{\sigma_t} \right) - 2s_2 \left(\frac{\epsilon}{\sigma_t} \right) - 2 \left(\frac{\epsilon}{\sigma_t} \right)^2 \right\} e^{-\frac{\sigma_t}{\epsilon} s_2}. \end{aligned} \quad (3.122)$$

The first integral in Eq. (3.119) is thus,

$$\begin{aligned} \int_{s_1}^{s_2} (s - s_1)^2 \frac{\sigma_t}{\epsilon} e^{-\frac{\sigma_t}{\epsilon} s} ds &= 2 \left(\frac{\epsilon}{\sigma_t} \right)^2 e^{-\frac{\sigma_t}{\epsilon} s_1} \\ &\quad + \left\{ -(s_2 - s_1)^2 + 2s_1 \left(\frac{\epsilon}{\sigma_t} \right) - 2s_2 \left(\frac{\epsilon}{\sigma_t} \right) - 2 \left(\frac{\epsilon}{\sigma_t} \right)^2 \right\} e^{-\frac{\sigma_t}{\epsilon} s_2}. \end{aligned} \quad (3.123)$$

We can now write Eq. (3.119) as the sum of Eqs. (3.120) and (3.123),

$$\int_0^{\infty} \tau^2 p_{\sigma_t} ds = 2 \left(\frac{\epsilon}{\sigma_t} \right)^2 e^{-\frac{\sigma_t}{\epsilon} s_1} + \left\{ 2s_1 \left(\frac{\epsilon}{\sigma_t} \right) - 2s_2 \left(\frac{\epsilon}{\sigma_t} \right) - 2 \left(\frac{\epsilon}{\sigma_t} \right)^2 \right\} e^{-\frac{\sigma_t}{\epsilon} s_2}. \quad (3.124)$$

Define a new random variable $\zeta_K^{(1)}$ equal to Eq. (3.124),

$$\zeta_K^{(1)} = 2 \left(\frac{\epsilon}{\sigma_t} \right)^2 e^{-\frac{\sigma_t}{\epsilon} s_1} + \left\{ 2s_1 \left(\frac{\epsilon}{\sigma_t} \right) - 2s_2 \left(\frac{\epsilon}{\sigma_t} \right) - 2 \left(\frac{\epsilon}{\sigma_t} \right)^2 \right\} e^{-\frac{\sigma_t}{\epsilon} s_2}. \quad (3.125)$$

We can label the terms and coefficients in Eq. (3.125) to show the order of each term and coefficient in the TDL. Recall that in the TDL, $\epsilon^p < \epsilon^q$ for $p > q$, because $\epsilon \in (0, 1]$. The terms in Eq. (3.125) scale as,

$$\underbrace{2 \left(\frac{\epsilon}{\sigma_t} \right)^2}_{O(\epsilon^2)} \underbrace{e^{-\frac{\sigma_t}{\epsilon} s_1}}_{O(1)} + \underbrace{\left(2s_1 \left(\frac{\epsilon}{\sigma_t} \right) - 2s_2 \left(\frac{\epsilon}{\sigma_t} \right) - 2 \left(\frac{\epsilon}{\sigma_t} \right)^2 \right)}_{O(\epsilon)} \underbrace{e^{-\frac{\sigma_t}{\epsilon} s_2}}_{O(1)}, \quad (3.126)$$

where the exponentials are bounded by 1 because $e^{-x} \in (0, 1]$ for $x \geq 0$. Thus, $\zeta_K^{(1)}$ is $O(\epsilon)$. The same labeling procedure can be applied to Q , which can be written as,

$$Q = (\sigma_t - \sigma_a) \frac{\varphi}{4\pi} + q. \quad (3.127)$$

Eq. (3.127) in the TDL is,

$$\begin{aligned} Q &= \left(\frac{\sigma_t}{\epsilon} - \epsilon \sigma_a \right) \frac{\varphi}{4\pi} + \epsilon q \\ &= \frac{\sigma_t}{\epsilon} \left(1 - \epsilon^2 \frac{\sigma_a}{\sigma_t} \frac{\varphi}{4\pi} + \epsilon^2 \frac{q}{\sigma_t} \right). \end{aligned} \quad (3.128)$$

The terms in Eq. (3.128) scale as,

$$\underbrace{\frac{\sigma_t}{\epsilon}}_{O(\epsilon^{-1})} \underbrace{\left(\underbrace{1}_{O(1)} - \underbrace{\epsilon^2 \frac{\sigma_a}{\sigma_t} \frac{\varphi}{4\pi}}_{O(\epsilon^2)} + \underbrace{\epsilon^2 \frac{q}{\sigma_t}}_{O(\epsilon^2)} \right)}_{O(\epsilon^{-1})}. \quad (3.129)$$

Thus, Q is $O(\epsilon^{-1})$.

Finally, we can rewrite Eq. (3.115) using Eq. (3.116) with Eq. (3.125) substituted,

$$\begin{aligned} E[f^2] &= \int_{\mathcal{D}} \int_{\mathbb{S}^2} \left(\int_0^\infty f^2 p \, ds \right) d\Omega \, d\mathbf{r}' \\ &= \int_{\mathcal{D}} \int_{\mathbb{S}^2} \left(4\pi \, \text{vol}(\mathcal{D}) \, Q^2 \int_0^\infty \tau^2 p_{\sigma_t} \, ds \right) d\Omega \, d\mathbf{r}' \\ &= 4\pi \, \text{vol}(\mathcal{D}) \int_{\mathcal{D}} \int_{\mathbb{S}^2} Q^2 \zeta_K^{(1)} \, d\Omega \, d\mathbf{r}'. \end{aligned} \quad (3.130)$$

We can label the terms in Eq. (3.130) using the orders of $\zeta_K^{(1)}$ and Q determined in Eqs. (3.126) and (3.129), respectively, as follows,

$$\underbrace{\underbrace{4\pi \text{vol}(\mathcal{D})}_{O(1)} \int_{\mathcal{D}} \int_{\mathbb{S}^2} \underbrace{Q^2}_{O(\epsilon^{-2})} \underbrace{\zeta_K^{(1)}}_{O(\epsilon)} d\Omega d\mathbf{r}'}_{O(\epsilon^{-1})}. \quad (3.131)$$

Thus, $E[f^2]$ is $O(\epsilon^{-1})$.

As shown by Eq. (3.114), the variance is the difference of two expectations, of which Eq. (3.130) is only the first. The second is $(E[f])^2$, for which we must consider $E[f]$,

$$E[f] = \int_{\mathcal{D}} \int_{\mathbb{S}^2} \int_0^\infty f p ds d\Omega d\mathbf{r}'. \quad (3.132)$$

The integral along the particle path is,

$$\begin{aligned} \int_0^\infty f p ds &= \int_0^\infty \left(4\pi \text{vol}(\mathcal{D}) Q \tau \right) \left(\frac{1}{\text{vol}(\mathcal{D})} \frac{1}{4\pi} p_{\sigma_t} \right) ds \\ &= Q \int_0^\infty \tau p_{\sigma_t} ds. \end{aligned} \quad (3.133)$$

Substituting Eq. (3.118) for p_{σ_t} and Eq. (3.56a) for τ into the integral gives,

$$\begin{aligned} \int_0^\infty \tau p_{\sigma_t} ds &= \int_0^\infty \tau \left(\frac{\sigma_t}{\epsilon} e^{-\frac{\sigma_t}{\epsilon} s} \right) ds \\ &= \int_{s_1}^{s_2} (s - s_1) \frac{\sigma_t}{\epsilon} e^{-\frac{\sigma_t}{\epsilon} s} ds + (s_2 - s_1) \int_{s_2}^\infty \frac{\sigma_t}{\epsilon} e^{-\frac{\sigma_t}{\epsilon} s} ds. \end{aligned} \quad (3.134)$$

The second integral in Eq. (3.134) is,

$$\begin{aligned} (s_2 - s_1) \int_{s_2}^\infty \frac{\sigma_t}{\epsilon} e^{-\frac{\sigma_t}{\epsilon} s} ds &= (s_2 - s_1) \left[-e^{-\frac{\sigma_t}{\epsilon} s} \right]_{s_2}^\infty \\ &= (s_2 - s_1) e^{-\frac{\sigma_t}{\epsilon} s_2}. \end{aligned} \quad (3.135)$$

The first integral in Eq. (3.134) may be computed using integration by parts,

$$\begin{aligned} \int_{s_1}^{s_2} (s - s_1) \frac{\sigma_t}{\epsilon} e^{-\frac{\sigma_t}{\epsilon} s} ds &= \left[-(s - s_1) e^{-\frac{\sigma_t}{\epsilon} s} \right]_{s_1}^{s_2} + \int_{s_1}^{s_2} e^{-\frac{\sigma_t}{\epsilon} s} ds \\ &= (s_1 - s_2) e^{-\frac{\sigma_t}{\epsilon} s_2} + \left[\frac{-\epsilon}{\sigma_t} e^{-\frac{\sigma_t}{\epsilon} s} \right]_{s_1}^{s_2} \\ &= (s_1 - s_2) e^{-\frac{\sigma_t}{\epsilon} s_2} - \frac{\epsilon}{\sigma_t} e^{-\frac{\sigma_t}{\epsilon} s_2} + \frac{\epsilon}{\sigma_t} e^{-\frac{\sigma_t}{\epsilon} s_1} \\ &= \left(s_1 - s_2 - \frac{\epsilon}{\sigma_t} \right) e^{-\frac{\sigma_t}{\epsilon} s_2} + \frac{\epsilon}{\sigma_t} e^{-\frac{\sigma_t}{\epsilon} s_1}. \end{aligned} \quad (3.136)$$

We can now rewrite Eq. (3.134) as the sum of Eqs. (3.135) and (3.136),

$$\int_0^\infty \tau p_{\sigma_t} ds = \frac{\epsilon}{\sigma_t} \left(e^{-\frac{\sigma_t}{\epsilon} s_1} - e^{-\frac{\sigma_t}{\epsilon} s_2} \right). \quad (3.137)$$

Define a new random variable $\zeta_K^{(0)}$ equal to Eq. (3.137),

$$\zeta_K^{(0)} = \frac{\epsilon}{\sigma_t} \left(e^{-\frac{\sigma_t}{\epsilon} s_1} - e^{-\frac{\sigma_t}{\epsilon} s_2} \right). \quad (3.138)$$

We can label the terms and coefficients in Eq. (3.138) to show the order of each term and coefficient. The terms in Eq. (3.138) scale as,

$$\underbrace{\frac{\epsilon}{\sigma_t}}_{O(\epsilon)} \underbrace{\left(\underbrace{e^{-\frac{\sigma_t}{\epsilon} s_1}}_{O(1)} - \underbrace{e^{-\frac{\sigma_t}{\epsilon} s_2}}_{O(1)} \right)}_{O(1)}. \quad (3.139)$$

Thus, $\zeta_K^{(0)}$ is $O(\epsilon)$. Finally, we can rewrite Eq. (3.132) using Eq. (3.133) with Eq. (3.138) substituted,

$$\begin{aligned} E[f] &= \int_{\mathcal{D}} \int_{\mathbb{S}^2} \left(\int_0^\infty f p ds \right) d\Omega d\mathbf{r}' \\ &= \int_{\mathcal{D}} \int_{\mathbb{S}^2} \left(Q \int_0^\infty \tau p_{\sigma_t} ds \right) d\Omega d\mathbf{r}' \\ &= \int_{\mathcal{D}} \int_{\mathbb{S}^2} Q \zeta_K^{(0)} d\Omega d\mathbf{r}'. \end{aligned} \quad (3.140)$$

We can label the terms in Eq. (3.140) using the orders of $\zeta_K^{(0)}$ and Q determined in Eqs. (3.129) and (3.139), respectively, as follows,

$$\underbrace{\int_{\mathcal{D}} \int_{\mathbb{S}^2} \underbrace{Q}_{O(\epsilon^{-1})} \underbrace{\zeta_K^{(0)}}_{O(\epsilon)} d\Omega d\mathbf{r}'}_{O(1)}. \quad (3.141)$$

Thus, $E[f]$ is $O(1)$. We can now label the terms in the variance Eq. (3.114), which scales as,

$$\text{Var}[f] = \underbrace{E[f^2]}_{O(\epsilon^{-1})} - \underbrace{(E[f])^2}_{O(1)}. \quad (3.142)$$

$\underbrace{\hspace{10em}}_{O(\epsilon^{-1})}$

Thus, the estimator $\hat{\phi}$ has variance $O(1/\epsilon)$ in the TDL. That is, $\text{Var}[\hat{\phi}]$ is $O(1/\epsilon)$. We can use this result along with Eq. (3.107) to determine $\text{Var}[\hat{\mathbf{T}}]$ in the TDL,

$$\begin{aligned}\text{Var}[\hat{\mathbf{T}}] &= \text{Var}\left[\hat{\mathbf{P}} - \frac{1}{3}\mathbf{I}\hat{\phi}\right] \\ &= \text{Var}[\hat{\mathbf{P}}] + \text{Var}\left[\frac{1}{3}\mathbf{I}\hat{\phi}\right] - 2\text{Cov}\left[\hat{\mathbf{P}}, \frac{1}{3}\mathbf{I}\hat{\phi}\right] \\ &= \text{Var}[\hat{\mathbf{P}}] + \frac{1}{9}\mathbf{I}\text{Var}[\hat{\phi}] - 2\text{Cov}\left[\hat{\mathbf{P}}, \frac{1}{3}\mathbf{I}\hat{\phi}\right],\end{aligned}\tag{3.143}$$

where $\text{Cov}[\cdot, \cdot]$ is the covariance of two random variables. I argue that $\text{Var}[\hat{\mathbf{T}}]$ is $O(1/\epsilon)$ in the TDL by arguing that the orders of the three terms in Eq. (3.143) are all $O(1/\epsilon)$:

1. The first term in Eq. (3.143), $\text{Var}[\hat{\mathbf{P}}]$, is $O(1/\epsilon)$ just like $\text{Var}[\hat{\phi}]$. In section 3.4.2, I described how $\hat{\phi}$ and $\hat{\mathbf{T}}$ are nearly the same volume estimator, in the sense that the latter is just the former multiplied by $\boldsymbol{\Omega} \otimes \boldsymbol{\Omega}$. The quantity $\boldsymbol{\Omega} \otimes \boldsymbol{\Omega}$ is $O(1)$, and multiplication of an $O(1/\epsilon)$ quantity by an order unity quantity leaves the order $O(1/\epsilon)$ quantity unchanged.
2. The second term in Eq. (3.143), $\frac{1}{9}\mathbf{I}\text{Var}[\hat{\phi}]$, is $O(1/\epsilon)$. This is because the constant tensor $\frac{1}{9}\mathbf{I}$ is $O(1)$, $\text{Var}[\hat{\phi}]$ is $O(1/\epsilon)$, and the product of the two is $O(1/\epsilon)$.
3. The third term in Eq. (3.143), $2\text{Cov}\left[\hat{\mathbf{P}}, \frac{1}{3}\mathbf{I}\hat{\phi}\right]$, is $O(1/\epsilon)$. The covariance of two random variables X and Y is $\text{Cov}[X, Y] = E[XY] - E[X]E[Y]$. The estimators $\hat{\mathbf{P}}$ and $\frac{1}{3}\mathbf{I}\hat{\phi}$ have asymptotic orders which are equivalent to the order of the function of random variables f defined in Eq. (3.65). Thus, we may express $2\text{Cov}\left[\hat{\mathbf{P}}, \frac{1}{3}\mathbf{I}\hat{\phi}\right]$ as $E[ff] - E[f]E[f] = E[f^2] - (E[f])^2$. This is just the definition of $\text{Var}[f]$, which we found to be $O(1/\epsilon)$.

Thus, $\text{Var}[\hat{\mathbf{T}}]$ is $O(1/\epsilon)$.

3.7 A Lower Variance Estimator for \mathbf{T}

In section 3.6 Derivation of $\text{Var}[\hat{\mathbf{T}}]$ in the Thick Diffusion Limit, we showed that $\text{Var}[\hat{\mathbf{T}}] = O(1/\epsilon)$. This is undesirable because of the way that $\text{Var}[\hat{\mathbf{T}}]$ appears in the expression for the difference of the estimator and the quantity that we are estimating. Specifically,

$$|\hat{\mathbf{T}} - \mathbf{T}| \sim \left(\text{Var}[\hat{\mathbf{T}}]\right)^{1/2} N^{-1/2},\tag{3.144}$$

where N is the number of HSM particles. Setting the right-hand side of Eq. (3.144) to unity, substituting $1/\epsilon$ for $\text{Var}[\hat{\mathbf{T}}]$, and then manipulating the result gives us,

$$\begin{aligned} \left(\text{Var}[\hat{\mathbf{T}}]\right)^{1/2} N^{-1/2} &= 1 \\ \epsilon^{-1/2} N^{-1/2} &= 1 \\ \epsilon^{-1/2} &= N^{1/2} \\ \epsilon^{-1} &= N. \end{aligned} \tag{3.145}$$

Thus, we need $N = O(1/\epsilon)$ HSM particles to maintain a fixed solution quality in the TDL. This means that we need more HSM particles in optically-thick media to maintain a fixed solution quality, and that attempting to maintain a fixed solution quality will cause the HSM method to use significantly more memory and computational time for an acceptably accurate solution in the optically-thick limit $\epsilon \rightarrow 0$.

3.7.1 Derivation of $\hat{\mathbf{R}}$ Estimator

In this section, we derive a new estimator for \mathbf{T} that we call $\hat{\mathbf{R}}$. The advantage of $\hat{\mathbf{R}}$ is that $\text{Var}[\hat{\mathbf{R}}]$ is $O(\epsilon)$ in the TDL. Recomputing Eqs. (3.144) and (3.145) using $\text{Var}[\hat{\mathbf{R}}] = \epsilon$ instead of $\text{Var}[\hat{\mathbf{T}}] = 1/\epsilon$, we observe that we can maintain a fixed solution quality in the TDL using a number of HSM particles that is equal to the optical thickness parameter, $N = O(\epsilon)$, as long as we compute $\hat{\mathbf{R}}$ instead of $\hat{\mathbf{T}}$. The derivation of $\hat{\mathbf{R}}$ uses integration by parts to introduce a factor of σ_t^{-1} into the scattering source. This reduces the magnitude of the scattering source from $O(1/\epsilon)$ to $O(\epsilon)$ in the TDL, while also generating additional source terms that must be computed.

We begin our derivation of $\hat{\mathbf{R}}$ by defining $Q(\mathbf{x}, \boldsymbol{\Omega})$ to be the right-hand side of Eq. (1.13a),

$$Q(\mathbf{x}, \boldsymbol{\Omega}) = \frac{\sigma_s}{4\pi} \varphi(\mathbf{x}) + q(\mathbf{x}, \boldsymbol{\Omega}). \tag{3.146}$$

The steady-state, gray, linear transport Eqs. (1.13a) and (1.13b), rewritten using Eq. (3.146), are:

$$\boldsymbol{\Omega} \cdot \nabla \psi(\mathbf{x}, \boldsymbol{\Omega}) + \sigma_t \psi(\mathbf{x}, \boldsymbol{\Omega}) = Q(\mathbf{x}, \boldsymbol{\Omega}) \quad \mathbf{x} \in \mathcal{D}, \tag{3.147a}$$

$$\psi(\mathbf{x}, \boldsymbol{\Omega}) = \psi_{\text{inc}}(\mathbf{x}, \boldsymbol{\Omega}), \quad \mathbf{x} \in \partial\mathcal{D} \text{ and } \boldsymbol{\Omega} \cdot \mathbf{n} < 0. \tag{3.147b}$$

For simplicity of presentation, consider a homogeneous medium, which is defined by a constant opacity⁸. The characteristic equation defining the solution to Eqs. (3.147a) and (3.147b) is,

$$\psi(\mathbf{r}, \boldsymbol{\Omega}) = e^{-\sigma_t s_0} \psi_{\text{inc}} + \int_0^{s_0} e^{-\sigma_t s} Q \, ds, \tag{3.148}$$

⁸In the general case of a heterogeneous medium, not only is the derivation of $\hat{\mathbf{R}}$ more complicated, but the $\hat{\mathbf{R}}$ that results will include sources defined on the material interfaces.

where $s_0(\mathbf{r}, \boldsymbol{\Omega}) = \min\{s \mid \mathbf{r} - s\boldsymbol{\Omega} \in \partial\mathcal{D}\}$ denotes the minimum distance from \mathbf{r} to the boundary $\partial\mathcal{D}$ when traveling in the direction $-\boldsymbol{\Omega}$ as shown in Fig. 3.1. For $\psi_{\text{inc}} = 0$, Eq. (3.148) is,

$$\psi(\mathbf{r}, \boldsymbol{\Omega}) = \int_0^{s_0} e^{-\sigma_t s} Q \, ds. \quad (3.149)$$

Consider the following manipulation of the right-hand side of Eq. (3.149) which gives a derivative in the integrand,

$$\int_0^{s_0} e^{-\sigma_t s} Q \, ds = -\frac{1}{\sigma_t} \int_0^{s_0} \frac{d}{ds}(e^{-\sigma_t s})Q \, ds, \quad (3.150)$$

and the product rule that we will use to replace the derivative in the integrand of Eq. (3.150),

$$\frac{d}{ds}(e^{-\sigma_t s} Q) = \frac{d}{ds}(e^{-\sigma_t s})Q + e^{-\sigma_t s} \frac{dQ}{ds}. \quad (3.151)$$

Substitute $\frac{dQ}{ds} = -\boldsymbol{\Omega} \cdot \nabla Q$ into Eq. (3.151) and re-arrange,

$$\frac{d}{ds}(e^{-\sigma_t s})Q = \frac{d}{ds}(e^{-\sigma_t s} Q) + e^{-\sigma_t s} \boldsymbol{\Omega} \cdot \nabla Q. \quad (3.152)$$

Use Eq. (3.152) in the right-hand side of Eq. (3.150),

$$\begin{aligned} -\frac{1}{\sigma_t} \int_0^{s_0} \frac{d}{ds}(e^{-\sigma_t s})Q \, ds &= -\frac{1}{\sigma_t} \int_0^{s_0} \frac{d}{ds}(e^{-\sigma_t s} Q) \, ds - \frac{1}{\sigma_t} \int_0^{s_0} e^{-\sigma_t s} \boldsymbol{\Omega} \cdot \nabla Q \, ds \\ &= -\frac{1}{\sigma_t} (e^{-\sigma_t s_0} Q(s_0) - Q(0)) - \frac{1}{\sigma_t} \int_0^{s_0} e^{-\sigma_t s} \boldsymbol{\Omega} \cdot \nabla Q \, ds \\ &= \frac{1}{\sigma_t} Q - \frac{1}{\sigma_t} e^{-\sigma_t s_0} Q(s_0) - \frac{1}{\sigma_t} \int_0^{s_0} e^{-\sigma_t s} \boldsymbol{\Omega} \cdot \nabla Q \, ds. \end{aligned} \quad (3.153)$$

Thus,

$$\psi(\mathbf{r}, \boldsymbol{\Omega}) = \frac{1}{\sigma_t} Q - \frac{1}{\sigma_t} e^{-\sigma_t s_0} Q(s_0) - \frac{1}{\sigma_t} \int_0^{s_0} e^{-\sigma_t s} \boldsymbol{\Omega} \cdot \nabla Q \, ds. \quad (3.154)$$

Using $\sigma_s = \sigma_t - \sigma_a$, we may rewrite Eq. (3.146) as,

$$Q = \frac{\sigma_t - \sigma_a}{4\pi} \varphi + q. \quad (3.155)$$

Multiplying Eq. (3.155) by σ_t^{-1} gives,

$$\frac{1}{\sigma_t} Q = \left(1 - \frac{\sigma_a}{\sigma_t}\right) \frac{\varphi}{4\pi} + \frac{1}{\sigma_t} q. \quad (3.156)$$

Subtracting $\varphi/4\pi$ from Eq. (3.156) gives,

$$\frac{1}{\sigma_t} Q - \frac{\varphi}{4\pi} = -\frac{\sigma_a}{\sigma_t} \frac{\varphi}{4\pi} + \frac{1}{\sigma_t} q, \quad (3.157)$$

which we will use shortly. Subtracting $\varphi/4\pi$ from Eq. (3.154) gives,

$$\psi(\mathbf{r}, \boldsymbol{\Omega}) - \frac{\varphi}{4\pi} = \left(\frac{1}{\sigma_t} Q - \frac{\varphi}{4\pi} \right) - \frac{1}{\sigma_t} e^{-\sigma_t s_0} Q(s_0) - \frac{1}{\sigma_t} \int_0^{s_0} e^{-\sigma_t s} \boldsymbol{\Omega} \cdot \nabla Q \, ds. \quad (3.158)$$

Substituting the right-hand side of Eq. (3.157) for the term in the big parentheses on the right-hand side of Eq. (3.158) gives,

$$\psi(\mathbf{r}, \boldsymbol{\Omega}) - \frac{\varphi}{4\pi} = -\frac{\sigma_a}{\sigma_t} \frac{\varphi}{4\pi} + \frac{1}{\sigma_t} q - \frac{1}{\sigma_t} e^{-\sigma_t s_0} Q(s_0) - \frac{1}{\sigma_t} \int_0^{s_0} e^{-\sigma_t s} \boldsymbol{\Omega} \cdot \nabla Q \, ds. \quad (3.159)$$

Now consider the new linear transport problem,

$$\boldsymbol{\Omega} \cdot \nabla \psi^{(1)} + \sigma_t \psi^{(1)} = -\frac{1}{\sigma_t} \boldsymbol{\Omega} \cdot \nabla Q \quad (3.160a)$$

$$\psi^{(1)} = -\frac{1}{\sigma_t} Q(s_0), \quad s_0 \in \partial \mathcal{D} \text{ and } \boldsymbol{\Omega} \cdot \mathbf{n} < 0. \quad (3.160b)$$

The characteristic equation defining the solution to Eqs. (3.160a) and (3.160b) is,

$$\psi^{(1)} = -\frac{1}{\sigma_t} e^{-\sigma_t s_0} Q(s_0) - \frac{1}{\sigma_t} \int_0^{s_0} e^{-\sigma_t s} \boldsymbol{\Omega} \cdot \nabla Q \, ds, \quad (3.161)$$

which lets us rewrite Eq. (3.159) as,

$$\psi(\mathbf{r}, \boldsymbol{\Omega}) - \frac{\varphi}{4\pi} = -\frac{\sigma_a}{\sigma_t} \frac{\varphi}{4\pi} + \frac{1}{\sigma_t} q + \psi^{(1)}. \quad (3.162)$$

The average of the SMM correction tensor \mathbf{T} over an element K is,

$$R_{K,i,j} = \frac{1}{\text{vol}(K)} \int_K \int_{\mathbb{S}^2} \Omega_i \Omega_j \left(\psi - \frac{\varphi}{4\pi} \right) d\Omega \, d\mathbf{x}. \quad (3.163)$$

Substituting the right-hand side of Eq. (3.162) into Eq. (3.163) gives,

$$\begin{aligned} R_{K,i,j} &= \frac{1}{\text{vol}(K)} \int_K \int_{\mathbb{S}^2} \Omega_i \Omega_j \psi^{(1)} \, d\Omega \, d\mathbf{x} \\ &\quad - \frac{\sigma_a}{\sigma_t} \frac{1}{\text{vol}(K)} \int_K \int_{\mathbb{S}^2} \Omega_i \Omega_j \frac{\varphi}{4\pi} \, d\Omega \, d\mathbf{x} \\ &\quad + \frac{1}{\sigma_t} \frac{1}{\text{vol}(K)} \int_K \int_{\mathbb{S}^2} \Omega_i \Omega_j q \, d\Omega \, d\mathbf{x}. \end{aligned} \quad (3.164)$$

We can use the path length estimators mentioned earlier in this chapter to approximate the first term in Eq. (3.164),

$$R_{K,i,j}^{(1)} = \frac{1}{\text{vol}(K)} \int_K \int_{\mathbb{S}^2} \Omega_i \Omega_j \psi^{(1)} \, d\Omega \, d\mathbf{x}, \quad (3.165)$$

which is just linear transport with source $-\sigma_t^{-1}\Omega \cdot \nabla Q$ and inflow $-\sigma_t^{-1}Q$. The second term in Eq. (3.164) is an integral of the moment system solution φ over the element K ,

$$\begin{aligned} R_{K,i,j}^{(2)} &= -\frac{\sigma_a}{\sigma_t} \frac{1}{\text{vol}(K)} \int_K \int_{\mathbb{S}^2} \Omega_i \Omega_j \frac{\varphi}{4\pi} d\Omega d\mathbf{x} \\ &= -\frac{\sigma_a}{\sigma_t} \frac{1}{\text{vol}(K)} \frac{\delta_{ij}}{3} \int_K \varphi d\mathbf{x}. \end{aligned} \quad (3.166)$$

The third term in Eq. (3.164) is an integral of the fixed source q over the element K ,

$$\begin{aligned} R_{K,i,j}^{(3)} &= \frac{1}{\sigma_t} \frac{1}{\text{vol}(K)} \int_K \int_{\mathbb{S}^2} \Omega_i \Omega_j q d\Omega d\mathbf{x} \\ &= \frac{1}{\sigma_t} \frac{1}{\text{vol}(K)} \frac{4\pi}{3} \delta_{ij} \int_K q d\Omega d\mathbf{x}. \end{aligned} \quad (3.167)$$

Thus, we can use Eqs. (3.165) to (3.167) to write Eq. (3.164) as,

$$R_{K,i,j} = R_{K,i,j}^{(1)} + R_{K,i,j}^{(2)} + R_{K,i,j}^{(3)}. \quad (3.168)$$

where the first two terms vary with each iteration and the third term is fixed.

3.7.2 Implementation Details of $\hat{\mathbf{R}}$ Estimator

To compute $\hat{\mathbf{R}}$, we need to

1. compute $\hat{\mathbf{P}}^{(1)}$ for $Q^{(1)} = -\frac{1}{\sigma_t}\Omega \cdot \nabla Q$ and $\psi_{\text{inc}}^{(1)} = -\frac{1}{\sigma_t}Q$,
2. integrate φ on every element K , and
3. integrate q on every element K .

The integration can be done using a suitable quadrature rule. Computing $\hat{\mathbf{P}}^{(1)}$ can be done using the Monte Carlo transport technique mentioned earlier in this chapter. Implementing the technique requires identifying the sources, which can be partitioned into volume sources and boundary sources as follows:

Volume Sources

Distributing the gradient and the dot product in $Q^{(1)}$ gives,

$$\begin{aligned} Q^{(1)} &= -\frac{1}{\sigma_t}\Omega \cdot \nabla Q \\ &= -\frac{1}{\sigma_t}\Omega \cdot \nabla \left(\frac{\sigma_s}{4\pi}\varphi + q \right) \\ &= -\frac{1}{\sigma_t}\Omega \cdot \left(\frac{\sigma_s}{4\pi}\nabla\varphi + \nabla q \right) \\ &= -\frac{\sigma_s}{\sigma_t} \frac{1}{4\pi}\Omega \cdot \nabla\varphi - \frac{1}{\sigma_t}\Omega \cdot \nabla q. \end{aligned} \quad (3.169)$$

Thus, $-\frac{1}{\sigma_t}\mathbf{\Omega} \cdot \nabla q$ is the fixed volume source and $-\frac{\sigma_s}{\sigma_t} \frac{1}{4\pi}\mathbf{\Omega} \cdot \nabla \varphi$ is the variable volume source.

Boundary Sources

Expanding Q in $\psi_{\text{inc}}^{(1)}$ gives,

$$\begin{aligned}\psi_{\text{inc}}^{(1)} &= -\frac{1}{\sigma_t}Q \\ &= -\frac{1}{\sigma_t} \left(\frac{\sigma_s}{4\pi}\varphi + q \right) \\ &= -\frac{\sigma_s}{\sigma_t} \frac{1}{4\pi}\varphi - \frac{1}{\sigma_t}q.\end{aligned}\tag{3.170}$$

Thus, $-\frac{1}{\sigma_t}q$ is the fixed boundary source and $-\frac{\sigma_s}{\sigma_t} \frac{1}{4\pi}\varphi$ is the variable boundary source.

Summary of Sources

Table 3.1 and table 3.2 summarize the sources for the transformed and original problems, respectively. The rightmost two columns give the order of the sources in the corresponding table locations to the left of the vertical line. The orders are determined by substituting the TDL scaling Eqs. (1.48a) to (1.48d). Let k be the exponent on ϵ in the order expression, so $O(\epsilon^k)$. Notice that $k \geq 0$ for all sources in the transformed problem, meaning that the transformed problem source magnitudes do not grow as $\epsilon \rightarrow 0$. They either shrink or stay constant. This contrasts with the original problem, for which $k \geq 0$ is not true, because of the scattering source. Observe that $(\sigma_s/(4\pi))\varphi$ is $O(1/\epsilon)$ in the TDL, meaning that it grows without bound as $\epsilon \rightarrow 0$. The integration by parts in the derivation of $\hat{\mathbf{R}}$ introduced a factor of σ_t^{-1} into the scattering source. This reduced the magnitude of the scattering source from $O(1/\epsilon)$ to $O(1)$ in the TDL, while also generating other source terms that must be computed.

	Fixed	Variable		
Volume	$-\frac{1}{\sigma_t}\mathbf{\Omega} \cdot \nabla q$	$-\frac{\sigma_s}{\sigma_t} \frac{1}{4\pi}\mathbf{\Omega} \cdot \nabla \varphi$	$O(\epsilon^2)$	$O(1)$
Boundary	$-\frac{1}{\sigma_t}q$	$-\frac{\sigma_s}{\sigma_t} \frac{1}{4\pi}\varphi$	$O(\epsilon^2)$	$O(1)$

Table 3.1: Source values and TDL orders for the transformed problem.

3.7.3 Derivation of $\text{Var}[\hat{\mathbf{R}}]$ in the Thick Diffusion Limit

Here we explain why $\hat{\mathbf{R}}$ may be a superior estimator for \mathbf{T} , which is that $\text{Var}[\hat{\mathbf{R}}] \ll \text{Var}[\hat{\mathbf{T}}]$ in the TDL. The derivation of $\text{Var}[\hat{\mathbf{R}}]$ resembles our derivation of $\text{Var}[\hat{\mathbf{T}}]$ in section 3.6, but the

	Fixed	Variable		
Volume	q	$\frac{\sigma_s}{4\pi}\varphi$	$O(\epsilon)$	$O(1/\epsilon)$
Boundary	0	0	N/A	N/A

Table 3.2: Source values and TDL orders for the original problem.

result will be different, because $\text{Var}[\hat{\mathbf{T}}]$ is $O(1/\epsilon)$, whereas we will see that $\text{Var}[\hat{\mathbf{R}}]$ is $O(\epsilon)$. A quantity that is $O(\epsilon)$ is smaller than a quantity which is $O(1/\epsilon)$ because $\epsilon \in (0, 1]$. In the TDL, $\epsilon \rightarrow 0$, so a $O(\epsilon)$ quantity also goes to 0 whereas a $O(1/\epsilon)$ quantity goes to infinity.

The linear transport problem in the $\hat{\mathbf{R}}$ estimator computation is slightly different than in the $\hat{\mathbf{T}}$ estimator computation. Both require transporting domain boundary and volume sources, but the magnitude of these sources in the $\hat{\mathbf{R}}$ estimator problem is attenuated by a factor of $1/\sigma_t$, which is ϵ/σ_t in the TDL. Table 3.3 shows these sources.

	Boundary Source	Volume Source	Function of Random Variables
$\hat{\mathbf{T}}$	ψ_{inc}	Q	$f = 4\pi \text{vol}(\mathcal{D})Q\tau$
$\hat{\mathbf{R}}$	$\psi_{\text{inc}}^{(1)} = \frac{-1}{\sigma_t}Q$	$Q^{(1)} = \frac{-\epsilon}{\sigma_t}\boldsymbol{\Omega} \cdot \nabla Q$	$h = 4\pi \text{vol}(\mathcal{D})Q^{(1)}\tau$

Table 3.3: Sources and random variable functions for the linear transport problems associated with the original estimator, $\hat{\mathbf{T}}$, and the improved estimator, $\hat{\mathbf{R}}$.

Recall from Eq. (3.129) that Q is $O(1/\epsilon)$ in the TDL. We can use this result to label the terms and coefficients in the $\hat{\mathbf{R}}$ volume source $Q^{(1)}$ to show the order of each term and coefficient. In the TDL, the $\hat{\mathbf{R}}$ volume source $Q^{(1)}$ is,

$$Q^{(1)} = \frac{-\epsilon}{\sigma_t}\boldsymbol{\Omega} \cdot \nabla Q, \quad (3.171)$$

and the terms in Eq. (3.171) scale as,

$$\underbrace{\frac{-\epsilon}{\sigma_t}}_{O(\epsilon)} \underbrace{\boldsymbol{\Omega}}_{O(1)} \cdot \underbrace{\nabla Q}_{O(\epsilon^{-1})}. \quad (3.172)$$

Thus, $Q^{(1)}$ is $O(1)$. Following section 3.6, we compute the expectation of the square first, then the square of the expectation. Consider the function $h = 4\pi \text{vol}(\mathcal{D})Q^{(1)}\tau$. The expectation of h^2 is,

$$E[h^2] = \int_{\mathcal{D}} \int_{\mathbb{S}^2} \int_0^\infty h^2 p \, ds \, d\boldsymbol{\Omega} \, d\mathbf{r}'. \quad (3.173)$$

The integral along the particle path is,

$$\begin{aligned} \int_0^\infty h^2 p \, ds &= \int_0^\infty \left(4\pi \, \text{vol}(\mathcal{D}) \, Q^{(1)} \, \tau \right)^2 \left(\frac{1}{\text{vol}(\mathcal{D})} \frac{1}{4\pi} p_{\sigma_t} \right) \, ds \\ &= 4\pi \, \text{vol}(\mathcal{D}) \left(Q^{(1)} \right)^2 \int_0^\infty \tau^2 p_{\sigma_t} \, ds. \end{aligned} \quad (3.174)$$

We can rewrite Eq. (3.173) using Eq. (3.174) with Eq. (3.125) substituted,

$$\begin{aligned} E[h^2] &= \int_{\mathcal{D}} \int_{\mathbb{S}^2} \left(\int_0^\infty h^2 p \, ds \right) \, d\Omega \, d\mathbf{r}' \\ &= \int_{\mathcal{D}} \int_{\mathbb{S}^2} \left(4\pi \, \text{vol}(\mathcal{D}) \left(Q^{(1)} \right)^2 \int_0^\infty \tau^2 p_{\sigma_t} \, ds \right) \, d\Omega \, d\mathbf{r}' \\ &= 4\pi \, \text{vol}(\mathcal{D}) \int_{\mathcal{D}} \int_{\mathbb{S}^2} \left(Q^{(1)} \right)^2 \zeta_K^{(1)} \, d\Omega \, d\mathbf{r}'. \end{aligned} \quad (3.175)$$

We can label the terms in Eq. (3.175) using the orders of $\zeta_K^{(1)}$ and $Q^{(1)}$ determined in Eqs. (3.126) and (3.172), respectively, as follows,

$$\underbrace{\underbrace{\underbrace{4\pi \, \text{vol}(\mathcal{D})}_{O(1)} \int_{\mathcal{D}} \int_{\mathbb{S}^2} \underbrace{\left(Q^{(1)} \right)^2}_{O(1)} \underbrace{\zeta_K^{(1)}}_{O(\epsilon)} \, d\Omega \, d\mathbf{r}'}_{O(\epsilon)}}_{O(\epsilon)}. \quad (3.176)$$

Thus, $E[h^2]$ is $O(\epsilon)$. The expectation of h is,

$$E[h] = \int_{\mathcal{D}} \int_{\mathbb{S}^2} \int_0^\infty h p \, ds \, d\Omega \, d\mathbf{r}'. \quad (3.177)$$

The integral along the particle path is,

$$\begin{aligned} \int_0^\infty h p \, ds &= \int_0^\infty \left(4\pi \, \text{vol}(\mathcal{D}) \, Q^{(1)} \, \tau \right) \left(\frac{1}{\text{vol}(\mathcal{D})} \frac{1}{4\pi} p_{\sigma_t} \right) \, ds \\ &= Q^{(1)} \int_0^\infty \tau p_{\sigma_t} \, ds. \end{aligned} \quad (3.178)$$

We can rewrite Eq. (3.177) using Eq. (3.178) with Eq. (3.138) substituted,

$$\begin{aligned} E[h] &= \int_{\mathcal{D}} \int_{\mathbb{S}^2} \left(\int_0^\infty h p \, ds \right) \, d\Omega \, d\mathbf{r}' \\ &= \int_{\mathcal{D}} \int_{\mathbb{S}^2} \left(Q^{(1)} \int_0^\infty \tau p_{\sigma_t} \, ds \right) \, d\Omega \, d\mathbf{r}' \\ &= \int_{\mathcal{D}} \int_{\mathbb{S}^2} Q^{(1)} \zeta_K^{(0)} \, d\Omega \, d\mathbf{r}'. \end{aligned} \quad (3.179)$$

We can label Eq. (3.179) using the orders of $\zeta_K^{(0)}$ and $Q^{(1)}$ determined in Eqs. (3.139) and (3.172), respectively, as follows,

$$\underbrace{\int_{\mathcal{D}} \int_{\mathbb{S}^2} \underbrace{Q^{(1)}}_{O(1)} \underbrace{\zeta_K^{(0)}}_{O(\epsilon)} d\Omega d\mathbf{r}'}_{O(\epsilon)}. \quad (3.180)$$

Thus, $E[h]$ is $O(\epsilon)$. We can now label the variance as,

$$\text{Var}[h] = \underbrace{E[h^2]}_{O(\epsilon)} - \underbrace{(E[h])^2}_{O(\epsilon^2)}. \quad (3.181)$$

Thus, the estimator $\hat{\phi}$ for the $\hat{\mathbf{R}}$ transport problem has variance $O(\epsilon)$ in the TDL. It follows that the estimator $\hat{\mathbf{P}}^{(1)}$ for $\mathbf{R}_{K,i,j}^{(1)}$ defined by Eq. (3.165) has variance $O(\epsilon)$ as well.

Finally, we consider the two additional terms $R_{K,i,j}^{(2)}$ and $R_{K,i,j}^{(3)}$ defined by Eqs. (3.166) and (3.167), respectively. As described in section 3.7.2, we do not solve a transport problem to compute them, but rather compute deterministic integrals. In that sense, they are constants, and $\text{Var}[c] = 0$ for any constant c , so they do not contribute to $\text{Var}[\hat{\mathbf{R}}]$. Thus, $\text{Var}[\hat{\mathbf{R}}]$ is $O(\epsilon)$. This is the reason to prefer $\hat{\mathbf{R}}$ over $\hat{\mathbf{T}}$ for estimating \mathbf{T} . The former has variance which decreases with optical thickness and the latter has variance which increases with optical thickness. My HSM implementation for this dissertation, which I describe in Chapter 4 and section 4.4, uses $\hat{\mathbf{T}}$. Future implementations may consider using $\hat{\mathbf{R}}$ instead.

3.8 Deviatoric Estimator for \mathbf{T}

In section 3.7, we derived an estimator for \mathbf{T} which we called $\hat{\mathbf{R}}$. While $\hat{\mathbf{R}}$ accomplishes the goal of dramatically reducing the estimator variance in the TDL, $\hat{\mathbf{R}}$ requires more sources. In a homogeneous medium with no inflow, estimating \mathbf{T} by computing the $\hat{\mathbf{T}}$ estimator defined in section 3.4.2 requires only two sources (see table 3.2). Computing $\hat{\mathbf{R}}$ to estimate \mathbf{T} requires twice as many sources, and the four required sources depend not only on q and φ , but also on their derivatives (see table 3.1). Finally, in a heterogeneous medium, the use of integration by parts to derive $\hat{\mathbf{R}}$ produces surface sources on material interfaces. This is because the opacity, which is a material property, is discontinuous across material interfaces.

In the worst case, which would be when every element contains a different material, $\hat{\mathbf{R}}$ would require sources on both sides of every face in Γ_0 (the set of unique faces internal to the mesh). The worst case also happens to be a common case: a radiation-hydrodynamics model of a system with significant, turbulent advection can give rise to a mesh containing unique materials in every element, even for an initial condition with only two unique materials and a small interface between them. This situation is virtually unavoidable, especially in

important physical applications, such as supernovae calculations and calculations of laser fusion experiments (see Appendix A). For a three-dimensional spatial mesh formed by the product of an equal number of hexahedral elements in each direction, the number of unique internal faces in the mesh is,

$$|\Gamma_0| = 3n^2(n - 1), \quad (3.182)$$

where n is the number of elements in each direction (such that the total number of elements $|\mathcal{T}|$ is n^3). Thus, the number of element face sources required to compute $\hat{\mathbf{R}}$ is $O(n^3)$, which is the same order as $|\mathcal{T}|$.

For a very coarse mesh with only $n = 10$ elements along each axis, computing the $\hat{\mathbf{R}}$ estimator would require $O(10^3)$ element internal face sources, and for a mesh with $n = 1000$, the $\hat{\mathbf{R}}$ estimator would require on the order of 1 billion element internal face sources. This would be very costly and perhaps even infeasible. As a result, it becomes necessary to consider an alternative to $\hat{\mathbf{R}}$ for estimating \mathbf{T} . The objective is to preserve the $O(\epsilon)$ variance property of $\hat{\mathbf{R}}$ while removing the dependence on material interface sources. In this section, I introduce an estimator that meets these criteria. I refer to this alternative as the “deviatoric” estimator, as it based on calculating the deviation of the solution from isotropy.

3.8.1 Derivation of Deviatoric Estimator

Define ψ , which is the solution of the linear transport Eq. (3.39a), to be the sum of the isotropic component and some arbitrary quantity $\tilde{\psi}$,

$$\psi = \frac{\varphi}{4\pi} + \tilde{\psi}. \quad (3.183)$$

Notice that substituting $\tilde{\psi} = 3(\mathbf{\Omega} \cdot \mathbf{J})/(4\pi)$ into Eq. (3.183) would result in Eq. (1.21), which is a spherical harmonics expansion truncated at the linear term. In Eq. (3.183), we truncated at the linear term, then introduced $\tilde{\psi}$. By writing Eq. (3.183), we have replaced the problem of computing ψ with the problem of computing $\tilde{\psi}$. Substituting Eq. (3.183) into Eq. (3.39a) gives,

$$\mathbf{\Omega} \cdot \nabla \left(\frac{\varphi}{4\pi} + \tilde{\psi} \right) + \sigma_t \left(\frac{\varphi}{4\pi} + \tilde{\psi} \right) = \frac{\sigma_s}{4\pi} \varphi + q. \quad (3.184)$$

Simplifying Eq. (3.184) gives,

$$\mathbf{\Omega} \cdot \nabla \tilde{\psi} + \sigma_t \tilde{\psi} = -\frac{1}{4\pi} (\sigma_a \varphi + \mathbf{\Omega} \cdot \nabla \varphi) + q. \quad (3.185a)$$

Note that the $\nabla \varphi$ term in Eq. (3.185a) is undefined if one replaces φ with the angle integrated intensity ϕ defined by Eq. (1.17). This is because ϕ may have points where the slope changes discontinuously, resulting in the function being non-differentiable at those points, for example at material interfaces. In the next section, Implementation Details of Deviatoric Estimator, I describe how to address this problem without resorting to material interface sources.

Substituting Eq. (3.183) into the boundary condition Eq. (3.39b) gives,

$$\tilde{\psi}(\mathbf{x}, \boldsymbol{\Omega}) = \psi_{\text{inc}}(\mathbf{x}, \boldsymbol{\Omega}) - \frac{\varphi(\mathbf{x})}{4\pi}, \quad \mathbf{x} \in \partial\mathcal{D} \text{ and } \boldsymbol{\Omega} \cdot \mathbf{n} < 0, \quad (3.185b)$$

where I have replaced $\bar{\psi}$ with ψ_{inc} in Eq. (3.185b) for notational clarity, as it is less likely to confuse ψ_{inc} with $\tilde{\psi}$ than to mistake $\bar{\psi}$ for $\tilde{\psi}$. Observe that we have replaced the linear transport problem defined by Eq. (3.39a), subject to the boundary condition Eq. (3.39b), with the linear transport problem defined by Eq. (3.185a), subject to the boundary condition Eq. (3.185b). We can relate the new problem to the old by using Eq. (3.183) to derive new estimators.

We start by considering the angle integrated intensity Eq. (1.17). Substituting Eq. (3.183) into Eq. (1.17) gives,

$$\phi = \int_{\mathbb{S}^2} \left(\frac{\varphi}{4\pi} + \tilde{\psi} \right) d\Omega. \quad (3.186)$$

Simplifying Eq. (3.186) gives,

$$\phi = \varphi + \int_{\mathbb{S}^2} \tilde{\psi} d\Omega. \quad (3.187)$$

Define $\tilde{\phi}$ to be,

$$\tilde{\phi} = \int_{\mathbb{S}^2} \tilde{\psi} d\Omega, \quad (3.188)$$

and substitute Eq. (3.188) into Eq. (3.187) to get,

$$\phi = \varphi + \tilde{\phi}. \quad (3.189)$$

For the pressure, substituting Eq. (3.183) into Eq. (1.19) and simplifying gives,

$$\mathbf{P} = \frac{\varphi}{3} \mathbf{I} + \tilde{\mathbf{P}}, \quad (3.190)$$

where we define $\tilde{\mathbf{P}}$ to be,

$$\tilde{\mathbf{P}} = \int_{\mathbb{S}^2} \boldsymbol{\Omega} \otimes \boldsymbol{\Omega} \tilde{\psi} d\Omega. \quad (3.191)$$

Finally, consider the boundary functional $B(\psi)$ in Eq. (1.25). Substituting Eq. (3.183) into Eq. (1.25) gives,

$$B = \int_{\mathbb{S}^2} |\boldsymbol{\Omega} \cdot \mathbf{n}| \left(\frac{\varphi}{4\pi} + \tilde{\psi} \right) d\Omega. \quad (3.192)$$

Simplifying Eq. (3.192) gives,

$$B = \frac{\varphi}{2} + \tilde{B}, \quad (3.193)$$

where we define \tilde{B} to be,

$$\tilde{B} = \int_{\mathbb{S}^2} |\boldsymbol{\Omega} \cdot \mathbf{n}| \tilde{\psi} d\Omega. \quad (3.194)$$

We can now define new, deviatoric estimators using the new, deviatoric forms of ϕ , \mathbf{P} , and $B(\psi)$ in Eq. (3.189), Eq. (3.190), Eq. (3.193), respectively. The estimators are,

$$\hat{\phi}_{\text{new}} = \varphi + \hat{\tilde{\phi}}, \quad (3.195a)$$

$$\hat{\mathbf{P}}_{\text{new}} = \frac{\varphi}{3}\mathbf{I} + \hat{\tilde{\mathbf{P}}}, \quad (3.195b)$$

$$\hat{\mathbf{T}}_{\text{new}} = \hat{\mathbf{P}}_{\text{new}} - \frac{1}{3}\mathbf{I}\hat{\phi}_{\text{new}}, \quad (3.195c)$$

$$\hat{B}_{\text{new}} = \frac{\varphi}{2} + \hat{\tilde{B}}, \quad (3.195d)$$

$$\hat{\phi}_{\text{new},s} = \varphi + \hat{\tilde{\phi}}_s, \quad (3.195e)$$

$$\hat{\beta}_{\text{new}} = \hat{B}_{\text{new}} - \frac{1}{2}\hat{\phi}_{\text{new},s}, \quad (3.195f)$$

where the symbols with hats above tildes denote the estimator of the symbol with the tilde. For example, $\hat{\tilde{\phi}}$ is an estimator of $\tilde{\phi}$. The hat-and-tilde symbols in Eqs. (3.195a) to (3.195f) are: $\hat{\tilde{\phi}}$, $\hat{\tilde{\mathbf{P}}}$, $\hat{\tilde{B}}$, and $\hat{\tilde{\phi}}_s$. We can compute these estimators using the $\tilde{\phi}$, $\tilde{\mathbf{P}}$, \tilde{B} and $\tilde{\phi}_s$ estimators that we previously derived in this chapter. The only difference is the source expressions, for which I provide more detail in the next section, Implementation Details of Deviatoric Estimator.

After we compute $\hat{\tilde{\phi}}$, $\hat{\tilde{\mathbf{P}}}$, $\hat{\tilde{B}}$, and $\hat{\tilde{\phi}}_s$, we form the new estimators in Eqs. (3.195a) to (3.195f) by simply adding linear terms which are functions of φ . The addends account for the isotropic component of the new estimators. The new angle integrated intensity estimator $\hat{\phi}_{\text{new}}$ is the deviatoric HSM method solution, and the associated SMM data are now $\hat{\mathbf{T}}_{\text{new}}$ and $\hat{\beta}_{\text{new}}$.

Finally, note that the isotropic components of the new estimators cancel when forming the new SMM data, $\hat{\mathbf{T}}_{\text{new}}$ and $\hat{\beta}_{\text{new}}$. The φ term in $\hat{\phi}_{\text{new}}$ cancels with the $(\varphi/3)\mathbf{I}$ term in $\hat{\mathbf{P}}_{\text{new}}$ when forming $\hat{\mathbf{T}}_{\text{new}}$. Substituting Eq. (3.195a) and Eq. (3.195b) into Eq. (3.195c) gives,

$$\begin{aligned} \hat{\mathbf{T}}_{\text{new}} &= \hat{\mathbf{P}}_{\text{new}} - \frac{1}{3}\mathbf{I}\hat{\phi}_{\text{new}} \\ &= \left(\frac{\varphi}{3}\mathbf{I} + \hat{\tilde{\mathbf{P}}}\right) - \frac{1}{3}\mathbf{I}\left(\varphi + \hat{\tilde{\phi}}\right) \\ &= \hat{\tilde{\mathbf{P}}} - \frac{\hat{\tilde{\phi}}}{3}\mathbf{I}. \end{aligned} \quad (3.196)$$

Eq. (3.196) implies an alternative notation for $\hat{\mathbf{T}}_{\text{new}}$,

$$\hat{\mathbf{T}} = \hat{\tilde{\mathbf{P}}} - \frac{\hat{\tilde{\phi}}}{3}\mathbf{I}, \quad (3.197)$$

where $\hat{\mathbf{T}} = \hat{\mathbf{T}}_{\text{new}}$, as established by Eq. (3.196). The $\varphi/2$ term in \hat{B}_{new} cancels with the φ term in $\hat{\phi}_{\text{new},s}$ when forming $\hat{\beta}_{\text{new}}$. Substituting Eq. (3.195d) and Eq. (3.195e) into Eq. (3.195f) gives,

$$\begin{aligned}\hat{\beta}_{\text{new}} &= \hat{B}_{\text{new}} - \frac{1}{2}\hat{\phi}_{\text{new},s} \\ &= \left(\frac{\varphi}{2} + \hat{B}\right) - \frac{1}{2}(\varphi + \hat{\phi}_s) \\ &= \hat{B} - \frac{1}{2}\hat{\phi}_s.\end{aligned}\tag{3.198}$$

Eq. (3.198) implies an alternative notation for $\hat{\beta}_{\text{new}}$,

$$\hat{\beta} = \hat{B} - \frac{1}{2}\hat{\phi}_s,\tag{3.199}$$

where $\hat{\beta} = \hat{\beta}_{\text{new}}$, as established by Eq. (3.198).

3.8.2 Implementation Details of Deviatoric Estimator

Comparing Eqs. (3.39a) and (3.39b) with Eqs. (3.185a) and (3.185b), we see that the sources in the new problem differ from the sources in the original problem. Table 3.4 shows the expressions for the volume and boundary sources.

	Fixed	Variable		
Volume	q	$\frac{\sigma_s}{4\pi}\varphi$	$O(\epsilon)$	$O(1/\epsilon)$
Boundary	ψ_{inc}	0	$O(1)$	N/A
	Fixed	Variable		
Volume	q	$-\frac{1}{4\pi}(\sigma_a\varphi + \mathbf{\Omega} \cdot \nabla\varphi)$	$O(\epsilon)$	$O(1)$
Boundary	ψ_{inc}	$-\frac{\varphi}{4\pi}$	$O(1)$	$O(1)$

Table 3.4: Source values and TDL orders for the original problem (top) and deviatoric problem (bottom).

Table 3.4 shows that, unlike the variable volume source in the original problem, the variable volume source in the deviatoric problem has a finite magnitude in the TDL. However, the source is more complicated because it involves derivatives of φ . This requires us to increase the order of the finite element space Y_p , defined in Eq. (2.19), from zero to unity, because we need φ to be at least linear in order for φ to have nonzero derivative. This

increases the number of degrees-of-freedom per element for φ from unity to four, hence raising the cost of the deterministic component described in Chapter 2. However, the increase in cost is minimal relative to the total cost, because the deterministic component accounts for only about 1% of the total HSM calculation runtime. Computing the gradient of the linear polynomial φ is also relatively inexpensive.

Table 3.4 also shows that, unlike the variable boundary source in the original problem, the variable boundary source in the deviatoric problem is nonzero. This means that we now need to source particles on the domain boundary (in addition to the domain volume) during the HSM iteration, and evaluate φ on the domain boundary at the locations of the boundary source particles. This is an expensive part of the deviatoric approach. However, it is orders of magnitude less expensive than the material interface sources required by $\hat{\mathbf{R}}$.

Consider, for example, the unit cube domain, $\mathcal{D} = [0, 1]^3$. Discretize \mathcal{D} with a mesh containing one billion elements, which is only 1000 elements along each axis. Evolve a turbulent, advective, multimaterial problem in time until the material interfaces are gone, and every element contains a unique mixture of the different materials. Computing $\hat{\mathbf{R}}$ on this mesh would require sampling $O(10^9)$ surface sources every cycle of the HSM iteration, which would be difficult, if not infeasible. Computing $\hat{\mathbf{T}}_{\text{new}}$ would require sampling just six surface sources every cycle of the HSM iteration, one on each face of the unit cube.

An issue with the deviatoric estimator that I mentioned in the previous section, Derivation of Deviatoric Estimator, is the undefined derivative when $\nabla\phi$ replaces $\nabla\varphi$ in Eq. (3.185a). This issue can be solved by replacing φ in Eq. (3.183) with some arbitrary function $\dot{\varphi}$ for which $\nabla\dot{\varphi}$ is well-defined. Substituting $\psi = \dot{\varphi}/(4\pi) + \tilde{\psi}$ into Eq. (3.39a) gives,

$$\mathbf{\Omega} \cdot \nabla \left(\frac{\varphi}{4\pi} + \tilde{\psi} \right) + \sigma_t \left(\frac{\varphi}{4\pi} + \tilde{\psi} \right) = \frac{(\sigma_t - \sigma_a)}{4\pi} \varphi + q, \quad (3.200)$$

where I replaced σ_s with $\sigma_t - \sigma_a$. Simplifying Eq. (3.200) gives,

$$\mathbf{\Omega} \cdot \nabla \tilde{\psi} + \sigma_t \tilde{\psi} = \frac{\sigma_t}{4\pi} (\varphi - \dot{\varphi}) - \frac{1}{4\pi} (\sigma_a \varphi + \mathbf{\Omega} \cdot \nabla \varphi) + q. \quad (3.201)$$

Eq. (3.201) highlights a crucial property that our choice of $\dot{\varphi}$ must satisfy for the deviatoric estimator to be efficient: $\varphi - \dot{\varphi}$ must be $O(1/\sigma_t)$. The reasoning behind this property will become clearer in the next section, Derivation of Deviatoric Estimator Variance in the Thick Diffusion Limit, where I show that $\text{Var}[\hat{\mathbf{T}}_{\text{new}}]$ is $O(\epsilon)$ in the TDL. The derivation of $\text{Var}[\hat{\mathbf{T}}_{\text{new}}]$ demonstrates that a necessary condition for the $O(\epsilon)$ variance is that the right-hand side of Eq. (3.185a) is $O(1)$ in the TDL. Observe, however, that the right-hand side of Eq. (3.201) is $O(1/\epsilon)$ in the TDL. This is because the term involving $\varphi - \dot{\varphi}$ is $O(1/\epsilon)$ in the TDL. Thus, preserving the $O(\epsilon)$ variance in the TDL requires choosing $\dot{\varphi}$ such that $\varphi - \dot{\varphi}$ is $O(1/\sigma_t)$. This choice causes $\sigma_t/(4\pi)$ and $\varphi - \dot{\varphi}$ to be the product of terms which are $O(1/\epsilon)$ and $O(\epsilon)$ in the TDL. The product of a term which is $O(1/\epsilon)$, and a term which is $O(\epsilon)$, is $O(1)$. Consequently, the necessary condition that the right-hand side of the linear transport equation is $O(1)$ is satisfied.

As an example, consider a problem with two materials, one which is optically-thick and one which is optically-thin. Denote the total opacities of the optically-thick and optically-thin materials as σ_t^{thick} and σ_t^{thin} , respectively. Let $\dot{\varphi}_1$ denote the choice of $\dot{\varphi}$ for this problem. The functional form of $\dot{\varphi}_1$ could be,

$$\dot{\varphi}_1 = \begin{cases} \varphi & \text{for } \{\mathbf{x} \mid \sigma_t(\mathbf{x}) = \sigma_t^{\text{thick}}\}, \\ \arg \min E[u] & \text{for } \{\mathbf{x} \mid \sigma_t(\mathbf{x}) = \sigma_t^{\text{thin}}\}, \end{cases} \quad (3.202)$$

where the minimization in Eq. (3.202) is performed with respect to the Dirichlet energy,

$$E[u] = \frac{1}{2} \int \|\nabla u(\mathbf{x})\|^2 d\mathbf{x}. \quad (3.203)$$

Minimizing the energy norm is equivalent to solving a Poisson problem, for which we would use the Dirichlet boundary condition,

$$u(\mathbf{x}) = \varphi(\mathbf{x}) \quad \text{for } \mathbf{x} \in \partial\mathcal{D}. \quad (3.204)$$

To illustrate the purpose of $\dot{\varphi}_1$, assume that the angle integrated intensity which solves the linear transport equation for the thick-thin problem is the absolute value function,

$$\phi(x) = |x - 3| + 1. \quad (3.205)$$

In Fig. 3.8, I plot Eq. (3.205) alongside the function $\dot{\varphi}_1$. In the optically-thick region, $\dot{\varphi}_1$ is equal to ϕ . In the optically-thin region, minimizing the Dirichlet energy offers an alternative to ϕ , ensuring that $\nabla\dot{\varphi}_1$ is well-defined everywhere.

3.8.3 Derivation of Deviatoric Estimator Variance in the Thick Diffusion Limit

We apply the same machinery that we developed in section 3.6, and used in section 3.7.3, to derive the variance of the deviatoric estimator for \mathbf{T} in the TDL. In section 3.6, we determined that $\text{Var}[\hat{\mathbf{T}}]$ is $O(1/\epsilon)$ in the TDL. In section 3.7.3, we determined that $\text{Var}[\hat{\mathbf{R}}]$ is $O(\epsilon)$ in the TDL, which represents a significant improvement, as $O(\epsilon) \ll O(1/\epsilon)$. Here, we derive $\text{Var}[\hat{\mathbf{T}}_{\text{new}}]$, which we find to be $O(\epsilon)$ in the TDL. This result indicates that $\hat{\mathbf{T}}_{\text{new}}$ is superior to both $\hat{\mathbf{R}}$ and $\hat{\mathbf{T}}$. Although $\hat{\mathbf{R}}$ and $\hat{\mathbf{T}}_{\text{new}}$ have the same $O(\epsilon)$ variance in the TDL, $\hat{\mathbf{T}}_{\text{new}}$ is a more efficient estimator because it is cheaper to compute than $\hat{\mathbf{R}}$ (due to the absence of material interface sources).

The derivation of $\text{Var}[\hat{\mathbf{T}}_{\text{new}}]$ follows the logic outlined in section 3.7.3, beginning with modifications to the quantities in the $\hat{\mathbf{R}}$ row of table 3.3. In the table, the boundary source for computing the $\hat{\mathbf{R}}$ estimator is,

$$\psi_{\text{inc}}^{(1)} = \frac{-1}{\sigma_t} Q, \quad (3.206)$$

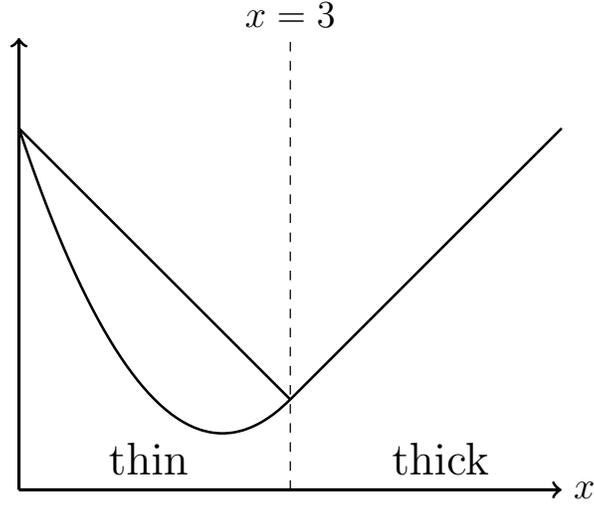


Figure 3.8: The function $\phi(x) = |x - 3| + 1$, which has undefined derivative at $x = 3$, is adjusted in the thin region ($x < 3$) such that the derivative becomes well-defined for all x values.

and the volume source for computing $\hat{\mathbf{R}}$ is,

$$Q^{(1)} = \frac{-1}{\sigma_t} \mathbf{\Omega} \cdot \nabla Q. \quad (3.207)$$

Replace the boundary source with,

$$\tilde{\psi}_{\text{inc}} = \psi_{\text{inc}} - \frac{\varphi}{4\pi}, \quad (3.208)$$

and the volume source with,

$$\tilde{Q} = \frac{-1}{4\pi} (\sigma_a \varphi + \mathbf{\Omega} \cdot \nabla \varphi) + q, \quad (3.209)$$

which are simply the boundary source and the volume source from the deviatoric linear transport Eq. (3.185a) and boundary condition Eq. (3.185b). In the TDL, Eq. (3.209) is,

$$\tilde{Q} = \frac{-1}{4\pi} (\epsilon \sigma_a \varphi + \mathbf{\Omega} \cdot \nabla \varphi) + \epsilon q, \quad (3.210)$$

and the terms in Eq. (3.210) scale as,

$$\underbrace{\frac{-1}{4\pi} \left(\underbrace{\epsilon \sigma_a \varphi}_{O(\epsilon)} + \underbrace{\mathbf{\Omega} \cdot \nabla \varphi}_{O(1)} \right) + \underbrace{\epsilon q}_{O(\epsilon)}}_{O(1)}. \quad (3.211)$$

Thus, \tilde{Q} is $O(1)$. This is the same order as $Q^{(1)}$. Per section 3.7.3, an order unity volume source leads to an estimator which is $O(\epsilon)$ in the TDL⁹. This means that $\hat{\phi}$ and $\hat{\mathbf{P}}$, which are estimators that appear in Eqs. (3.195a) and (3.195b) for $\hat{\phi}_{\text{new}}$ and $\hat{\mathbf{P}}_{\text{new}}$, respectively, are both $O(\epsilon)$ in the TDL. The last step before we obtain $\text{Var}[\hat{\mathbf{T}}_{\text{new}}]$ is to obtain $\text{Var}[\hat{\phi}_{\text{new}}]$ and $\text{Var}[\hat{\mathbf{P}}_{\text{new}}]$.

The variance of Eq. (3.195a) is,

$$\text{Var}[\hat{\phi}_{\text{new}}] = \text{Var}[\varphi + \hat{\phi}]. \quad (3.212)$$

If we treat φ as a constant¹⁰, then Eq. (3.212) simplifies to,

$$\text{Var}[\hat{\phi}_{\text{new}}] = \text{Var}[\hat{\phi}], \quad (3.213)$$

which is $O(\epsilon)$. Similarly, the variance of Eq. (3.195b) is,

$$\text{Var}[\hat{\mathbf{P}}_{\text{new}}] = \text{Var} \left[\frac{\varphi}{3} \mathbf{I} + \hat{\mathbf{P}} \right], \quad (3.214)$$

Again, if φ is considered a constant, then Eq. (3.214) simplifies to,

$$\text{Var}[\hat{\mathbf{P}}_{\text{new}}] = \text{Var}[\hat{\mathbf{P}}], \quad (3.215)$$

which is also $O(\epsilon)$. From Eq. (3.143), we have that,

$$\text{Var}[\hat{\mathbf{T}}_{\text{new}}] = \text{Var}[\hat{\mathbf{P}}_{\text{new}}] + \frac{1}{9} \mathbf{I} \text{Var}[\hat{\phi}_{\text{new}}] - 2 \text{Cov} \left[\hat{\mathbf{P}}_{\text{new}}, \frac{1}{3} \mathbf{I} \hat{\phi}_{\text{new}} \right], \quad (3.216)$$

After writing Eq. (3.143), I argued that $\text{Var}[\hat{\mathbf{T}}]$ is $O(1/\epsilon)$ in the TDL by demonstrating that all three terms in Eq. (3.143) are of the same order, $O(1/\epsilon)$. The sum of three terms with the same order remains of the same order, $O(1/\epsilon)$. Using the same reasoning (outlined in the enumerated paragraphs following Eq. (3.143)), I claim that the three terms in Eq. (3.216) are all $O(\epsilon)$. Therefore, the sum of these terms is also $O(\epsilon)$.

Thus, $\text{Var}[\hat{\mathbf{T}}_{\text{new}}]$ is $O(\epsilon)$.

⁹Read section 3.7.3 beginning at Eq. (3.172) and observe that the result is the same if one substitutes \tilde{Q} for $Q^{(1)}$ in the equations that follow.

¹⁰Although φ varies during the HSM iteration, it is fixed during each cycle of the HSM iteration, including during the Monte Carlo transport solve.

Chapter 4

Combining the Two Components of Hybrid Second Moment

In this chapter, I collect the results of Chapters 2 and 3 which are necessary for my HSM implementation and concisely present them with details on how I combine the deterministic and Monte Carlo components to create the hybrid method. The hybrid quality of the method is immediately apparent in section 4.1, where I discuss the convergence numerics, which have a Monte Carlo term and a deterministic term. In section 4.2, I describe how I separate the fixed source and the scattering source, which is made possible by the hybrid method, because a fully deterministic SMM cannot treat them separately. I describe my HSM algorithm in section 4.3. Finally, I describe choices that I made in my HSM implementation in section 4.4, such as my choice of how many spatial dimensions to consider.

4.1 Numerical Properties

We hypothesize that the error of the HSM solution is $O(h) + O(N^{-1/2})$. The first term is due to the h^{p+1} convergence of the mixed finite element discretization of the SMM system, where $p = 0$ in our case because we use lowest-order finite elements. Thus, decreasing the numerical error due to the spatial discretization by a factor of 2 requires decreasing the element width h by the same factor. The second term is due to the $\text{Var}[\cdot]/\sqrt{N}$ uncertainty of the estimators computed in the Monte Carlo component of HSM. Thus, decreasing the uncertainty of the MC estimators by a factor of 2 requires increasing the number of MC particles N by a factor of 4.

4.2 Separating the Sources

There are two sources of photons in Eq. (3.39a), the fixed source q and the scattering source, which is the product of $\sigma_s/(4\pi)$ and φ . As the name implies, the fixed source does not change, whereas the scattering source changes with each iteration of the HSM method.

This is not only true of HSM. The scattering source changes with each iteration of SMM. Deterministic SMM calculations must add the fixed source into the scattering source iterate every cycle in order to compute the right-hand side vector for the linear system which is inverted using the matrix-free transport sweep. Our hybrid SMM method incorporates the effect of the scattering source by having the MC particles carry the weight of the scattering source iterate every cycle.

Using Monte Carlo instead of a transport sweep allows for the sources to be separated. That is, we may use MC to sample q and ψ_{inc} before iterating, then sample the scattering source in the iteration. This allows us to re-use all of the particle memory devoted to sampling q and ψ_{inc} (before the iteration) for sampling the scattering source (inside the iteration). It also allows us to compute the first cycle moment solution iterate $\varphi^{(1)}$ using nonzero correction estimates $\hat{\mathbf{T}}$ and $\hat{\beta}$, which will be better approximations than using zero for problems where \mathbf{T} and β are nonzero. Finally, separation provides sampling flexibility, in that the position and direction of simulation particles can be sampled differently for q , ψ_{inc} , and the scattering source. The cost of separating the sources is the extra memory usage required to store the pre-iteration estimators $\varphi^{(0)}$, $\hat{\mathbf{T}}^{(0)}$, and $\hat{\beta}^{(0)}$. The storage cost of the estimators grows linearly with the number of mesh elements.

4.3 Hybrid Second Moment Algorithm

Algorithm 3 shows how to use a Monte Carlo solver to implement HSM while also preserving the functionality of an unaccelerated MC solver. Lines 3-5 run the Monte Carlo solver defined by Algorithm 2 *without* HSM. Lines 7-17 implement HSM. Line 8 runs the same MC solver to sample the fixed source q and compute tallies while disregarding the scattering source, thus computing the unscattered contribution to the solution, which I have decomposed as $\phi = \phi_{\text{unscattered}} + \phi_{\text{scattered}}$. This allows me to reuse all the MC particle memory that I allocate for sampling the fixed source to sample the scattering source. Line 9 initializes the iteration index i to 1. The loop, which I describe in the next paragraph, iterates until convergence of the scattering source, which is slightly different in every cycle.

Line 11 runs the Second Moment solver defined by Algorithm 1, which uses the fixed source angular moments Q_0 and Q_1 as well as the SMM data $\hat{\mathbf{T}}^{(i-1)}$ and $\hat{\beta}^{(i-1)}$. The i in the superscript is the iteration index, so a quantity with superscript $(i-1)$ is an iterate from the previous cycle. The coefficient matrix and right-hand-side integrals involving Q_0 and Q_1 must be computed on the first cycle of the iteration, but can then be stored and re-used for subsequent cycles. Only the integrals involving \mathbf{T} and β , which are linear forms that appear on the right-hand side of Eq. (2.22b), must be recomputed every cycle. Line 13 runs the MC solver to sample the scattering source, which is the product of $\sigma_s/(4\pi)$ and $\varphi^{(i)}$. I store the tallies computed during the scattering source sampling in temporary variables denoted by the “temp” subscript. Lines 13-15 update the iterates by accumulating the scattered tallies into the unscattered tallies, and line 16 updates the iteration index. Line 18 returns the solution, which is the MC tally that was computed upon convergence of the scattering

source, $\hat{\phi}^{(i)}$. The iteration converges when the relative difference of successive iterates falls below a user-provided threshold η ,

$$\max_j \left(\frac{|\hat{\phi}_j^{(i-1)} - \hat{\phi}_j^{(i)}|}{\hat{\phi}_j^{(i-1)}} \right) < \eta, \quad j = 1, \dots, |\mathcal{T}|, \quad (4.1)$$

where $|\mathcal{T}|$ is the number of elements in the mesh.

Algorithm 3 Hybrid Second Moment

```

1: Input: user-provided boolean value HSM
2: if not HSM then
3:   scattering_events  $\leftarrow$  true
4:    $\hat{\phi} \leftarrow$  mc( $q$ , scattering_events,  $\bar{\psi}$ )
5:   return  $\hat{\phi}$ 
6: end if
7: scattering_events  $\leftarrow$  false
8:  $\hat{\phi}^{(0)}, \hat{\mathbf{T}}^{(0)}, \hat{\beta}^{(0)} \leftarrow$  mc( $q$ , scattering_events,  $\bar{\psi}$ )
9:  $i \leftarrow 1$ 
10: while not converged( $\hat{\phi}^{(i-1)}, \hat{\phi}^{(i)}$ ) do
11:    $\varphi^{(i)} \leftarrow$  sm( $Q_0, \mathbf{Q}_1, \hat{\mathbf{T}}^{(i-1)}, \hat{\beta}^{(i-1)}$ )
12:    $\hat{\phi}_{\text{temp}}, \hat{\mathbf{T}}_{\text{temp}}, \hat{\beta}_{\text{temp}} \leftarrow$  mc( $\varphi^{(i)}$ , scattering_events)
13:    $\hat{\phi}^{(i)} \leftarrow \hat{\phi}^{(0)} + \hat{\phi}_{\text{temp}}$ 
14:    $\hat{\mathbf{T}}^{(i)} \leftarrow \hat{\mathbf{T}}^{(0)} + \hat{\mathbf{T}}_{\text{temp}}$ 
15:    $\hat{\beta}^{(i)} \leftarrow \hat{\beta}^{(0)} + \hat{\beta}_{\text{temp}}$ 
16:    $i \leftarrow i + 1$ 
17: end while
18: return  $\hat{\phi}^{(i)}$ 

```

4.4 Implementation Specifics

In this dissertation, my description of the HSM method takes into account all three spatial dimensions, but an effective demonstration of the method might use less. In section 4.4.1, I present a few choices of different dimensionalities, and explain how I selected two spatial dimensions for my implementation. An implementation ultimately requires a programming language, compiler, platform, and other details. I describe my choices for these in section 4.4.2. The choices that I made are simply one way to implement my HSM method. They are not necessarily better than alternative implementation choices.

4.4.1 Dimensionality

The radiation intensity is seven dimensional because $I = I(\mathbf{x}, \boldsymbol{\Omega}, \nu, t)$ has $3+2+1+1=7$ independent variables. While $\mathbf{x} = (x, y, z)^T$ and $\boldsymbol{\Omega} = (\Omega_x, \Omega_y, \Omega_z)^T$ are both three-vectors, the latter is normalized to unity, $|\boldsymbol{\Omega}| = 1$, which means that it may be specified using two dimensions instead of three. Those dimensions are the azimuthal angle ϕ and polar angle θ , as shown in Fig. 4.1. The three elements of $\boldsymbol{\Omega}$ are the projections of $\boldsymbol{\Omega}$ onto the coordinate axes,

$$\boldsymbol{\Omega} = \begin{pmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{pmatrix} = \begin{pmatrix} \boldsymbol{\Omega} \cdot \mathbf{e}_x \\ \boldsymbol{\Omega} \cdot \mathbf{e}_y \\ \boldsymbol{\Omega} \cdot \mathbf{e}_z \end{pmatrix} = \begin{pmatrix} \boldsymbol{\Omega} \cdot (1, 0, 0)^T \\ \boldsymbol{\Omega} \cdot (0, 1, 0)^T \\ \boldsymbol{\Omega} \cdot (0, 0, 1)^T \end{pmatrix} = \begin{pmatrix} \sin \theta \cos \phi \\ \sin \theta \sin \phi \\ \cos \theta \end{pmatrix}. \quad (4.2)$$

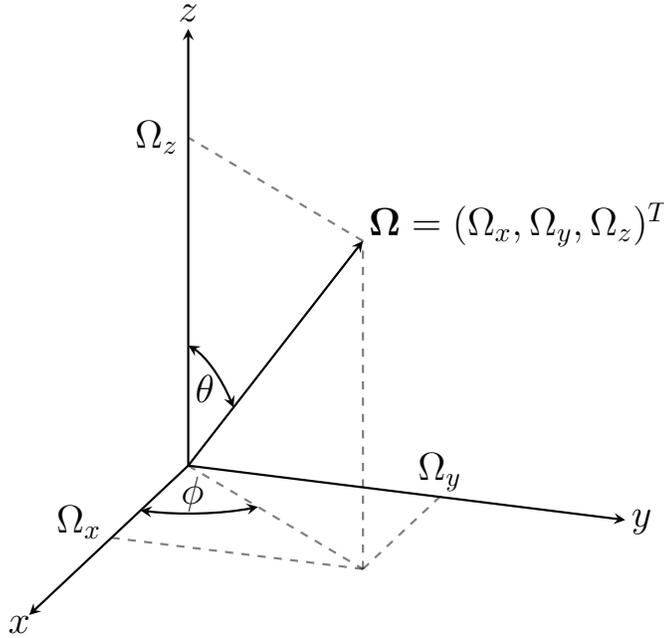


Figure 4.1: The direction vector $\boldsymbol{\Omega}$.

In section 1.3.2, I said that if one disregards time- and frequency-dependence, then one may consider a reduced-dimensional intensity $\psi = \psi(\mathbf{x}, \boldsymbol{\Omega})$ that has only $3+2=5$ coordinates. If one additionally assumes azimuthal symmetry of the solution, then $\partial\psi/\partial x = \partial\psi/\partial y = 0$, and the solution is only two dimensional, $\psi = \psi(z, \mu)$, where $\mu = \cos \theta$ is the projection of $\boldsymbol{\Omega}$ onto the z -axis. This is captured by Column V of table 4.1.

One spatial dimension would simplify my HSM implementation, but it may not be enough to convincingly demonstrate HSM because many aspects of the deterministic component of HSM are greatly simplified. In one spatial dimension,

- the current \mathbf{J} is a scalar,

	Column I Full	Column II Steady-State	Column III Gray	Column IV 2D Cartesian	Column V 1D Slab
Transport	7 coordinates $x, y, z, \theta, \phi, \nu, t$	6 coordinates $x, y, z, \theta, \phi, \nu$	5 coordinates x, y, z, θ, ϕ	4 coordinates x, y, θ, ϕ	2 coordinates $z, \mu \doteq \cos \theta$
Moment	5 coordinates x, y, z, ν, t	4 coordinates x, y, z, ν	3 coordinates x, y, z	2 coordinates x, y	1 coordinate z

Table 4.1: Transport equation and moment system dimensionality reductions.

- the gradient and divergence are just ordinary derivatives, and
- the RT space is a discrete subspace of $H^1(\mathcal{D})$ because there is no distinction between $H(\text{div}; \mathcal{D})$ and $H^1(\mathcal{D})$.

The simplified, azimuthally symmetric geometry is called “1D Slab” geometry because it has just a single spatial dimension in Cartesian space, as opposed to having a single dimension in curvilinear space, which is called “1D spherical” geometry. Assume instead that the solution does not change along the z -axis. That is, $\partial\psi/\partial z = 0$. The solution is then four dimensional, $\psi = \psi(\mathbf{x}, \mathbf{\Omega})$, where $\mathbf{x} = (x, y)^T$ and $\mathbf{\Omega}$ has the usual dependence on the azimuthal and polar angle coordinates. Two spatial dimensions preserves the vector current, preserves the distinction of the gradient and divergence, and preserves the distinction of $H(\text{div}; \mathcal{D})$ and $H^1(\mathcal{D})$. I use the term “2D Cartesian” (or “2D XY”) for this geometry because it has two spatial dimensions in Cartesian space, as opposed to having two dimensions in curvilinear space, which I call “2D Curvilinear” (or “2D RZ”) geometry. It is captured by Column IV in table 4.1 and it is the geometry type of my HSM implementation.

An example of the oversimplification that 1D geometries provide manifests in angular integration, which is the same for table 4.1 Columns I-IV, but differs for Column V,

$$\begin{aligned} \text{Column I-IV : } \int_{S^2} (\cdot) d\Omega &= \int_{\theta=0}^{\pi} \int_{\phi=0}^{2\pi} (\cdot) \sin \theta d\phi d\theta, \\ \text{Column V : } \int_{S^2} (\cdot) d\Omega &= \int_{\phi=0}^{2\pi} d\phi \int_{\mu=0}^1 (\cdot) d\mu = 2\pi \int_{\mu=-1}^1 (\cdot) d\mu. \end{aligned}$$

Even methods for problems without angular dependence encounter oversimplification in 1D. One example occurs when solving the Navier-Stokes equations for fluid dynamics, where Lagrangian is strictly superior to Eulerian because it is impossible to have mesh tangling in 1D. Thus, neither 1D Lagrangian nor 1D Eulerian tangle, yet the latter has additional numerical error due to mass advection, so 1D Lagrangian is superior. For the same reasons, 1D Arbitrary Lagrangian-Eulerian (ALE) also makes no sense, yet both Eulerian and ALE are incredibly important methods for hydrodynamics. One who never leaves 1D would disagree.

4.4.2 Programming Language, Compiler, Platform, and Dependencies

I implemented HSM in less than 5000 lines of code that I wrote using the C and C++ programming languages [77, 78]. I used C everywhere except for the SMM solver which is C++ because it uses the MFEM finite element library which is written in C++ [79]. My SMM solver is a lightly modified copy of the mixed FEM SMM solver implemented by Olivier in [7]. My C code adheres to the C99 standard with GNU extensions (`-std=gnu99`) and my C++ code adheres to the C++11 standard (`-std=c++11`). I compile with gcc and g++ versions 10.3.1 and link with the mvapich 2.3.7 implementation [80] of the MPI standard. I run on the Lawrence Livermore National Laboratory (LLNL) RZWhippet cluster, which has two sockets per node. Each socket has one Intel Xeon Platinum 8479 “Sapphire Rapids” processor, providing 56 CPU cores per socket, for a total of 112 cores per RZWhippet node [81]. I use one MPI rank per core, so I run 112 ranks on 1 node, 224 ranks on 2 nodes, and so on. The operating system is a Linux distribution based off RHEL8 called TOSS [82]. I link my executable with the following libraries,

- Lua 5.4.6 [83]
- IREP 1.0.0 [84]
- MFEM 4.6 [79]
- SuperLU 6.0.1 [61]
- HDF5 1.14.3 [85]
- GPerfTools 2.15 [86]

I use Lua (with IREP) for parsing user input, MFEM and SuperLU for the SMM solver, HDF5 to write the solution, and GPerfTools to generate program counter sampling and heap memory allocation profiles. I used address sanitizer (`-fsanitize=address`) to find memory errors and gdb for general debugging.

I wrote tens of thousands of lines of code prototyping 1D and 2D implementations of my HSM method in Python, using a finite difference method instead of the mixed finite element method. I also wrote thousands of lines of Python code while experimenting with MFEM using PyMFEM, which is MFEM’s Python interface [87]. My results in [88] are from my 1D and 2D Python prototypes. My results in [89] and here in this dissertation are from my C and C++ implementation, which is 2D.

Chapter 5

Non-deviatoric Numerical Results

This chapter describes numerical results that I collected by running calculations using my HSM implementation. The three categories of results which I present are i) the verification of my HSM implementation using the Method of Manufactured Solutions, ii) the efficiency of my HSM implementation in the thick diffusion limit, and iii) the effectiveness of my HSM implementation on two-material problems composed of one optically-thick material and one optically-thin material. The aforementioned categories i) and ii) appear in sections 5.1 and 5.2, respectively. Category iii) is constituted by both sections 5.3 and 5.4, each of which include a thick-and-thin two-material problem, but with different geometries and sources.

Implementers of numerical methods distinguish between method “verification” and “validation”. Verification answers the question,

Are we solving the equations right?

Validation answers the question,

Are we solving the right equations?

An example of verification is the comparison of the numerical solution with an analytic solution, which is a solution for which the functional form is known. An example of validation is the comparison of the numerical solution with an experiment, such as the experiment described in Appendix A. I run verification calculations. Validation calculations, which are often more complicated than verification calculations, exceed the scope of this dissertation. I present verification calculations in a single material in section 5.1 and section 5.2, and in two materials in sections 5.3 and 5.4. My intention is to provide an interesting demonstration of my HSM implementation that could be used to inform one’s decision whether to use my novel HSM method in production computer simulations of hot matter.

5.1 Method of Manufactured Solutions Problem

The Method of Manufactured Solutions (MMS) is a code verification technique in which one chooses the analytic solution, which is called the “MMS solution”. One then substitutes the MMS solution into the PDE to derive expressions for the problem data, which includes: sources, boundary conditions, and the initial condition [90–92]. I use the MMS to verify that my numerical solution converges to the MMS solution under mesh refinement and MC sample augmentation, and that the rate of convergence with respect to the element width and number of MC particles matches my hypothesis of $O(h) + O(N^{-1/2})$. My hypothesized convergence rate $O(h) + O(N^{-1/2})$ defines an error surface that decreases in height as one traverses simultaneously upward and rightward in Fig. 5.1.

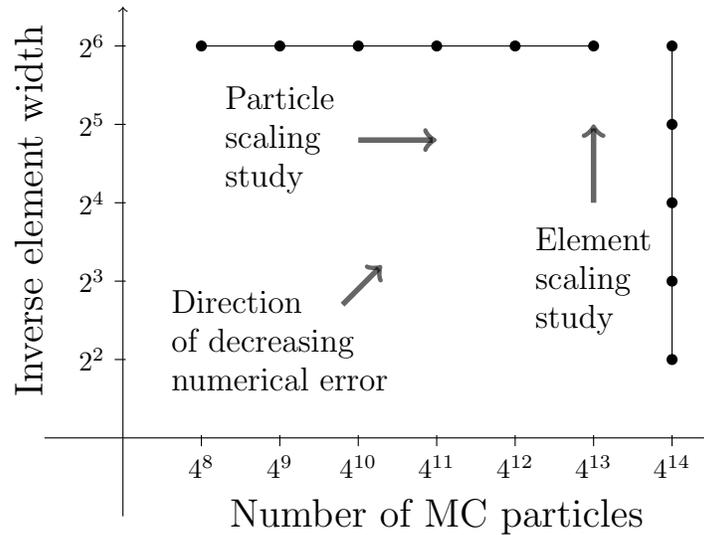


Figure 5.1: Calculation points on the HSM error surface.

By making h small and running multiple calculations with increasing N in the “particle scaling study”, I observe the $O(N^{-1/2})$ term in the hypothesized convergence rate, and by making N large and running multiple calculations with decreasing h in the “element scaling study”, I observe the $O(h)$ term. I let $\mathcal{D} = [0, 1]^2$, $\sigma_t = 2$, $\sigma_s = 1$, and I solve the problem specified by Eqs. (1.13a) and (1.13b) for the MMS solution in Equation (89) in [39]. The MMS solution is,

$$\begin{aligned} \psi_{\text{MMS}}(\mathbf{x}, \boldsymbol{\Omega}) = & \frac{1}{4\pi} \left(\sin(\pi x) \sin(\pi y) + \boldsymbol{\Omega}_x \boldsymbol{\Omega}_y \sin(2\pi x) \sin(2\pi y) \right. \\ & \left. + \boldsymbol{\Omega}_x^2 \sin\left(\frac{5\pi}{2}x + \frac{\pi}{4}\right) \sin\left(\frac{5\pi}{2}y + \frac{\pi}{4}\right) + 0.5 \right). \end{aligned} \quad (5.1)$$

I substitute the MMS solution into the equations on the left of Fig. 1.3 and solve for q and $\bar{\psi}$ and use them in the MC component of my HSM solver, and I substitute the MMS

solution into the equations on the right of Fig. 1.3 and solve for Q_0 , \mathbf{Q}_1 , and J_{in} and use them in the deterministic component of my HSM solver. Eq. (5.1) cannot be computed with the radiation diffusion approximation because it is quadratically-anisotropic. Eq. (5.1) exercises the coupling of the MC component of HSM with the deterministic component of HSM because it has nonzero SMM data $\mathbf{T} \neq 0$ and $\beta \neq 0$. Finally, Eq. (5.1) exercises my implementation of the boundary conditions because it has nonzero inflow $\bar{\psi} \neq 0$, and so $J_{\text{in}} \neq 0$.

The solution that I calculate with HSM is the angle integrated intensity, which I can express analytically by integrating Eq. (5.1) over the unit sphere to get,

$$\phi_{\text{MMS}}(\mathbf{x}) = \sin(\pi x) \sin(\pi y) + \frac{1}{3} \sin\left(\frac{5\pi}{2}x + \frac{\pi}{4}\right) \sin\left(\frac{5\pi}{2}y + \frac{\pi}{4}\right) + 0.5. \quad (5.2)$$

The HSM solution is an estimator of the angle integrated intensity averaged over each element. Averaging Eq. (5.2) over an arbitrary rectangle gives the analytic expression to which I compare my HSM solution,

$$\bar{\phi}_{\text{MMS}}(x_1, x_2, y_1, y_2) = \frac{1}{((x_2 - x_1)(y_2 - y_1))} \int_{x_1}^{x_2} \int_{y_1}^{y_2} \phi_{\text{MMS}}(\mathbf{x}) \, dy \, dx. \quad (5.3)$$

Evaluating the integrals in Eq. (5.3) gives the final analytic expression,

$$\begin{aligned} \bar{\phi}_{\text{MMS}}(x_1, x_2, y_1, y_2) = & \frac{1}{((x_2 - x_1)(y_2 - y_1))} \left(\frac{1}{\pi^2} \left\{ \left[\cos(\pi x_2) - \cos(\pi x_1) \right] \left[\cos(\pi y_2) - \cos(\pi y_1) \right] \right\} \right. \\ & \left. + \frac{4}{75\pi^2} \left\{ \left[\cos\left(\frac{5\pi}{2}x_2 + \frac{\pi}{4}\right) - \cos\left(\frac{5\pi}{2}x_1 + \frac{\pi}{4}\right) \right] \left[\cos\left(\frac{5\pi}{2}y_2 + \frac{\pi}{4}\right) - \cos\left(\frac{5\pi}{2}y_1 + \frac{\pi}{4}\right) \right] \right\} \right). \end{aligned} \quad (5.4)$$

I use a computational mesh of equally-sized squares, so I simply evaluate Eq. (5.4) using the x_1, x_2, y_1 , and y_2 values which define the left, right, bottom, and top edges of each square in my mesh. Fig. 5.2 shows plots of ϕ_{MMS} and $\bar{\phi}_{\text{MMS}}$ on the calculation domain $\mathcal{D} = [0, 1]^2$. I determine the error of my numerical solution with respect to Eq. (5.4) by computing the norm of the difference in the following manner.

First, define the width of the rectangles in a mesh of equal-sized rectangles to be,

$$\Delta x = 1/N_x, \quad (5.5a)$$

where N_x is the number of elements along the x -axis, or equivalently the number of columns in the mesh. Define the height of the equal-sized rectangles to be,

$$\Delta y = 1/N_y, \quad (5.5b)$$

where N_y is the number of elements along the y -axis, or equivalently the number of rows in the mesh. Let $\hat{\phi}_{i,j}$ correspond to the value of the estimator in row i and column j of the mesh, $i = 1, \dots, N_y$ and $j = 1, \dots, N_x$, as in Fig. 5.3.

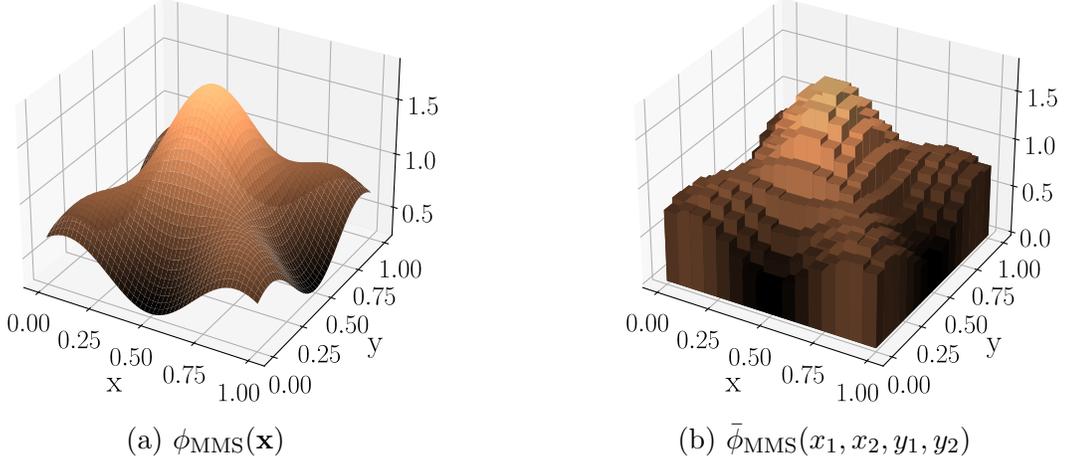


Figure 5.2: The MMS solution Eq. (5.2) and its average Eq. (5.4) on $\mathcal{D} = [0, 1]^2$.

Let $\hat{\phi}_q(x, y) : [0, 1] \times [0, 1] \rightarrow \{\hat{\phi}_{i,j} \mid i, j \in \mathbb{Z}, 1 \leq i \leq N_y, 1 \leq j \leq N_x\}$ be a quantization function defined by,

$$\hat{\phi}_q(x, y) = \begin{cases} \hat{\phi}_{1,1} & x < \Delta x \text{ and } y < \Delta y \\ \hat{\phi}_{1,2} & \Delta x < x < 2\Delta x \text{ and } y < \Delta y \\ \vdots & \vdots \\ \hat{\phi}_{1,N_x} & (N_x - 1)\Delta x < x < 1 \text{ and } y < \Delta y \\ \hat{\phi}_{2,1} & x < \Delta x \text{ and } \Delta y < y < 2\Delta y \\ \hat{\phi}_{2,2} & \Delta x < x < 2\Delta x \text{ and } \Delta y < y < 2\Delta y \\ \vdots & \vdots \\ \hat{\phi}_{2,N_x} & (N_x - 1)\Delta x < x < 1 \text{ and } \Delta y < y < 2\Delta y \\ \vdots & \vdots \\ \vdots & \vdots \\ \hat{\phi}_{N_y,N_x} & (N_x - 1)\Delta x < x < 1 \text{ and } (N_y - 1)\Delta y < y < 1. \end{cases} \quad (5.6)$$

Let $\bar{\phi}$ be a MMS solution tensor with $\dim(\bar{\phi}) = \dim(\hat{\phi})$ defined by,

$$\bar{\phi}_{i,j} = \bar{\phi}_{\text{MMS}}\left((i-1)\Delta x, i\Delta x, (j-1)\Delta y, j\Delta y\right), \quad (5.7)$$

where $\bar{\phi}_{\text{MMS}}$ is Eq. (5.4) and $i, j \in \mathbb{Z}, 1 \leq i \leq N_y, 1 \leq j \leq N_x$. Let $\bar{\phi}_{\text{MMS}}^q(x, y)$ be another quantization function, defined exactly like $\hat{\phi}_q(x, y)$, but with $\hat{\phi}$ in Eq. (5.6) replaced by $\bar{\phi}$ as defined in Eq. (5.7). Define the numerical error of the HSM method with respect to the MMS solution as $e(x, y) : [0, 1] \times [0, 1] \rightarrow \mathbb{R}$,

$$e(x, y) = \hat{\phi}_q(x, y) - \bar{\phi}_{\text{MMS}}^q(x, y). \quad (5.8)$$

The continuous $L^2(\mathcal{D})$ norm of $e(x, y)$ is,

$$\|e(x, y)\|_2 = \sqrt{\int_0^1 \int_0^1 |e(x, y)|^2 dy dx}. \quad (5.9)$$

Approximate Eq. (5.9) using a Riemann sum,

$$\begin{aligned} \|e(x, y)\|_2 &\approx \sqrt{\sum_{i=1}^{N_y} \sum_{j=1}^{N_x} |e(x_i, y_i)|^2 \Delta x \Delta y} \\ &= \frac{\left(\sum_{i=1}^{N_y} \sum_{j=1}^{N_x} |e(x_i, y_i)|^2\right)^{1/2}}{(N_x N_y)^{1/2}}. \end{aligned} \quad (5.10)$$

Eq. (5.10) can be written using the vector 2-norm of the difference of $\hat{\phi}$ and $\bar{\phi}$,

$$\frac{\left(\sum_{i=1}^{N_y} \sum_{j=1}^{N_x} |e(x_i, y_i)|^2\right)^{1/2}}{(N_x N_y)^{1/2}} = \frac{\|\hat{\phi} - \bar{\phi}\|_2}{(N_x N_y)^{1/2}}, \quad (5.11)$$

where the vector 2-norm $\|\cdot\|_2$ is defined as,

$$\|\mathbf{a}\|_2 = \sqrt{\sum_i (a_i)^2}, \quad (5.12)$$

for some vector \mathbf{a} with elements a_i . The vector 2-norm in Eq. (5.11) is for the flattened representation where $\hat{\phi}, \bar{\phi} \in \mathbb{R}^{N_y N_x}$. In the tensor representation, $\hat{\phi}, \bar{\phi} \in \mathbb{R}^{N_y \times N_x}$, replace the vector 2-norm $\|\cdot\|_2$ in Eq. (5.11) with the Frobenius norm $\|\cdot\|_F$.

Finally, $\Delta x = \Delta y$ because the equal-sized rectangular elements in my mesh are actually equal-sized squares, so $N_x = N_y$, and Eq. (5.11) becomes,

$$\frac{\|\hat{\phi} - \bar{\phi}\|_2}{(N_x N_y)^{1/2}} = \frac{\|\hat{\phi} - \bar{\phi}\|_2}{|\mathcal{T}|^{1/2}}, \quad (5.13)$$

where $|\mathcal{T}|$ is the number of elements in the mesh. The result in Eq. (5.13) is the way that I compute the error in the element scaling study, which corresponds to the vertical line in Fig. 5.1,

$$\text{Element scaling study error} = \frac{\|\hat{\phi} - \bar{\phi}\|_2}{|\mathcal{T}|^{1/2}}, \quad (5.14)$$

For the particle scaling study, which corresponds to the horizontal line in Fig. 5.1, the mesh is fixed at 64×64 elements for all 6 calculations, so I replace the normalization factor in the denominator of Eq. (5.14) by unity,

$$\text{Particle scaling study error} = \|\hat{\phi} - \bar{\phi}\|_2. \quad (5.15)$$

The results in Fig. 5.4 demonstrate that HSM converges under mesh refinement and MC sample augmentation, and that the rate of convergence with respect to the element width and number of MC particles matches our hypothesis of $O(h) + O(N^{-1/2})$. The slight degradation in convergence at the highest mesh resolution, appearing as liftoff above the dashed curve at the bottom-left of Fig. 5.4b, is expected behavior because I distribute a fixed number of MC particles across more and more elements under mesh refinement. I confirmed that this was the case by running fewer MC particles and observing earlier liftoff, as well as running more MC particles and observing delayed liftoff.

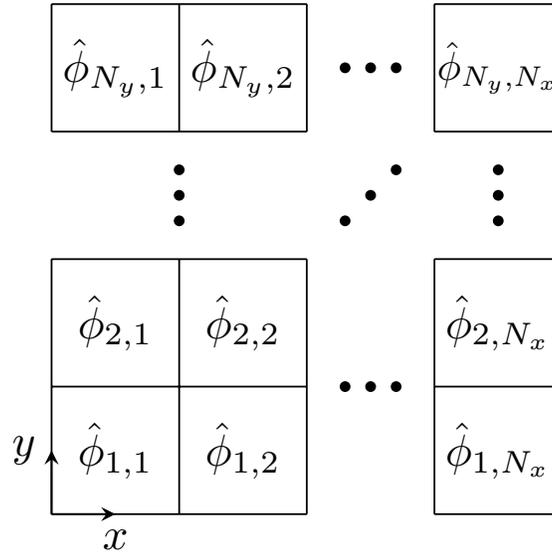


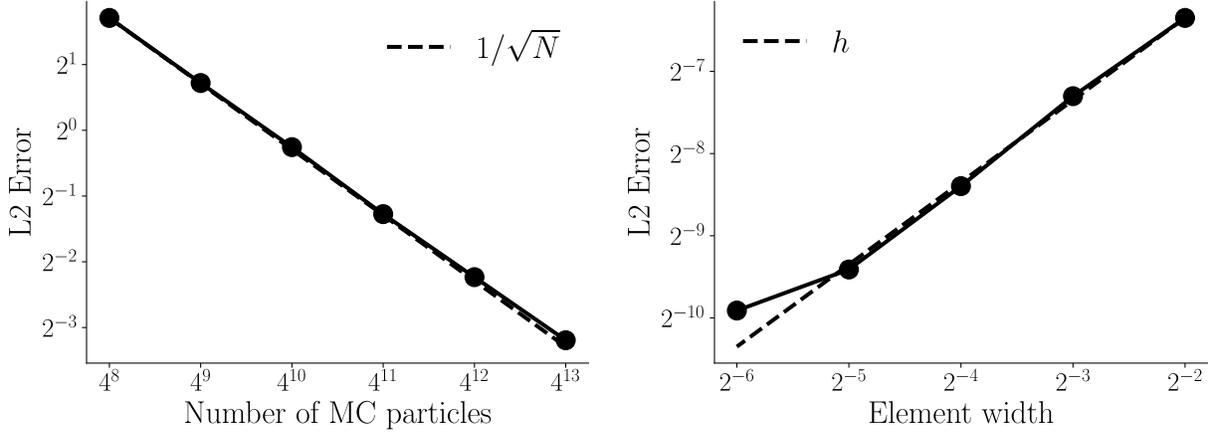
Figure 5.3: Indexing of $\hat{\phi}$ estimator on a mesh of equal-size rectangles (in this case, squares).

5.1.1 Source Expressions

After writing the MMS solution Eq. (5.1), I mentioned that its substitution into Eq. (1.13a) and Eq. (1.13b) results in the MMS sources for q and $\bar{\psi}$, which we need for the MC solve, and that its substitution into the SMM system of Eqs. (1.20a), (1.43) and (1.46) results in Q_0 , Q_1 , and J_{in} , which we need for the moment solve. These quantities are also sometimes called the PDE “data”. For completeness of my presentation of the MMS verification, I present the result of the substitutions here.

The q source used in the MC solve is,

$$\begin{aligned} q_{\text{MMS}}(\mathbf{x}, \boldsymbol{\Omega}) &= \boldsymbol{\Omega} \cdot \nabla \psi_{\text{MMS}} + \sigma_t \psi_{\text{MMS}} - \frac{\sigma_s}{4\pi} \phi_{\text{MMS}} \\ &= \Omega_x \frac{\partial \psi_{\text{MMS}}}{\partial x} + \Omega_y \frac{\partial \psi_{\text{MMS}}}{\partial y} + \sigma_t \psi_{\text{MMS}} - \frac{\sigma_s}{4\pi} \phi_{\text{MMS}}, \end{aligned} \quad (5.16a)$$



(a) Error as number of MC particles increases. (b) Error as element width decreases.

Figure 5.4: Error of HSM iterate $\hat{\phi}^{(i)}$ upon convergence.

where ψ_{MMS} is Eq. (5.1), ϕ_{MMS} is Eq. (5.2), and

$$\begin{aligned} \frac{\partial \psi_{\text{MMS}}}{\partial x}(\mathbf{x}, \boldsymbol{\Omega}) &= \frac{1}{4\pi} \left(\pi \cos(\pi x) \sin(\pi y) + 2\pi \boldsymbol{\Omega}_x \boldsymbol{\Omega}_y \cos(2\pi x) \sin(2\pi y) \right. \\ &\quad \left. + \frac{5\pi}{2} \boldsymbol{\Omega}_x^2 \cos\left(\frac{5\pi}{2}x + \frac{\pi}{4}\right) \sin\left(\frac{5\pi}{2}y + \frac{\pi}{4}\right) \right), \end{aligned} \quad (5.16b)$$

$$\begin{aligned} \frac{\partial \psi_{\text{MMS}}}{\partial y}(\mathbf{x}, \boldsymbol{\Omega}) &= \frac{1}{4\pi} \left(\pi \sin(\pi x) \cos(\pi y) + 2\pi \boldsymbol{\Omega}_x \boldsymbol{\Omega}_y \sin(2\pi x) \cos(2\pi y) \right. \\ &\quad \left. + \frac{5\pi}{2} \boldsymbol{\Omega}_x^2 \sin\left(\frac{5\pi}{2}x + \frac{\pi}{4}\right) \cos\left(\frac{5\pi}{2}y + \frac{\pi}{4}\right) \right). \end{aligned} \quad (5.16c)$$

The $\bar{\psi}$ source used in the MC solve is,

$$\bar{\psi}_{\text{MMS}}(\mathbf{x}, \boldsymbol{\Omega}) = \frac{1}{4\pi} + \boldsymbol{\Omega}_x^2 \sin\left(\frac{5\pi}{2}x + \frac{\pi}{4}\right) \cos\left(\frac{5\pi}{2}y + \frac{\pi}{4}\right) + 0.5, \quad (5.17)$$

where the domain of definition is $\mathbf{x} \in \partial\mathcal{D}$ and $\boldsymbol{\Omega} \cdot \mathbf{n} < 0$. The Q_0 source used in the moment solve is,

$$Q_{0,\text{MMS}}(\mathbf{x}) = \sigma_a \phi_{\text{MMS}}, \quad (5.18)$$

where ϕ_{MMS} is Eq. (5.2). The Q_1 source used in the moment solve is,

$$\begin{aligned} Q_{1,\text{MMS}}(\mathbf{x}) &= \sigma_t \mathbf{J}_{\text{MMS}} + \frac{1}{3} \nabla \phi_{\text{MMS}} + \nabla \cdot \mathbf{T} \\ &= \frac{1}{3} \begin{pmatrix} \frac{\partial \phi_{\text{MMS}}}{\partial x} \\ \frac{\partial \phi_{\text{MMS}}}{\partial y} \end{pmatrix} + \begin{pmatrix} \frac{\partial T_{\text{MMS}}^{11}}{\partial x} + \frac{\partial T_{\text{MMS}}^{12}}{\partial y} \\ \frac{\partial T_{\text{MMS}}^{21}}{\partial x} + \frac{\partial T_{\text{MMS}}^{22}}{\partial y} \end{pmatrix}, \end{aligned} \quad (5.19a)$$

where the partial derivatives are,

$$\frac{\partial \phi_{\text{MMS}}}{\partial x}(\mathbf{x}) = \pi \cos(\pi x) \sin(\pi y) + \frac{5\pi}{18} \cos\left(\frac{5\pi}{2}x + \frac{\pi}{4}\right) \sin\left(\frac{5\pi}{2}y + \frac{\pi}{4}\right), \quad (5.19b)$$

$$\frac{\partial \phi_{\text{MMS}}}{\partial y}(\mathbf{x}) = \pi \sin(\pi x) \cos(\pi y) + \frac{5\pi}{18} \sin\left(\frac{5\pi}{2}x + \frac{\pi}{4}\right) \cos\left(\frac{5\pi}{2}y + \frac{\pi}{4}\right), \quad (5.19c)$$

$$\frac{\partial T_{\text{MMS}}^{11}}{\partial x}(\mathbf{x}) = \frac{4}{45} \frac{5\pi}{2} \cos\left(\frac{5\pi}{2}x + \frac{\pi}{4}\right) \sin\left(\frac{5\pi}{2}y + \frac{\pi}{4}\right), \quad (5.19d)$$

$$\frac{\partial T_{\text{MMS}}^{12}}{\partial y}(\mathbf{x}) = \frac{2\pi}{15} \sin(2\pi x) \cos(2\pi y), \quad (5.19e)$$

$$\frac{\partial T_{\text{MMS}}^{21}}{\partial x}(\mathbf{x}) = \frac{2\pi}{15} \cos(2\pi x) \sin(2\pi y), \quad (5.19f)$$

$$\frac{\partial T_{\text{MMS}}^{22}}{\partial y}(\mathbf{x}) = -\frac{2}{45} \frac{5\pi}{2} \sin\left(\frac{5\pi}{2}x + \frac{\pi}{4}\right) \cos\left(\frac{5\pi}{2}y + \frac{\pi}{4}\right), \quad (5.19g)$$

For $\mathcal{D} = [0, 1]^2$, the J_{in} source used in the moment solve has piecewise definition on the four edges of the unit square,

$$\begin{aligned} J_{\text{in, MMS}}(\mathbf{x}) &= \int_{\Omega \cdot \mathbf{n} < 0} \boldsymbol{\Omega} \cdot \mathbf{n} \bar{\psi}_{\text{MMS}} \, d\Omega \\ &= \begin{cases} J_{\text{MMS}}^{\text{right}} & x = 1, \\ J_{\text{MMS}}^{\text{left}} & x = 0, \\ J_{\text{MMS}}^{\text{top}} & y = 1, \\ J_{\text{MMS}}^{\text{bottom}} & y = 0. \end{cases} \end{aligned} \quad (5.20a)$$

We can analytically compute these four functions as follows: the functions on the vertical boundaries are,

$$\begin{aligned} J_{\text{MMS}}^{\text{right}}(y) &= \int_{\Omega \cdot \mathbf{n}_{\text{right}}} \boldsymbol{\Omega} \cdot \mathbf{n}_{\text{right}} \bar{\psi}_{\text{MMS}} \, d\Omega \\ &= \int_{\theta=0}^{\pi} \int_{\phi=\pi/2}^{3\pi/2} \boldsymbol{\Omega} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} \bar{\psi}_{\text{MMS}}(1, y) \sin \theta \, d\phi \, d\theta \\ &= -\frac{1}{8} - \frac{1}{8} \sin\left(\frac{11\pi}{4}\right) \sin\left(\frac{5\pi}{2}y + \frac{\pi}{4}\right), \end{aligned} \quad (5.20b)$$

and

$$\begin{aligned}
J_{\text{MMS}}^{\text{left}}(y) &= \int_{\boldsymbol{\Omega} \cdot \mathbf{n}_{\text{left}}} \boldsymbol{\Omega} \cdot \mathbf{n}_{\text{right}} \bar{\psi}_{\text{MMS}} \, d\Omega \\
&= \int_{\theta=0}^{\pi} \int_{\phi=-\pi/2}^{\pi/2} \boldsymbol{\Omega} \cdot \begin{pmatrix} -1 \\ 0 \end{pmatrix} \bar{\psi}_{\text{MMS}}(0, y) \sin \theta \, d\phi \, d\theta \\
&= -\frac{1}{8} - \frac{1}{8} \sin\left(\frac{\pi}{4}\right) \sin\left(\frac{5\pi}{2}y + \frac{\pi}{4}\right). \tag{5.20c}
\end{aligned}$$

The functions on the horizontal boundaries are,

$$\begin{aligned}
J_{\text{MMS}}^{\text{top}}(x) &= \int_{\boldsymbol{\Omega} \cdot \mathbf{n}_{\text{top}}} \boldsymbol{\Omega} \cdot \mathbf{n}_{\text{top}} \bar{\psi}_{\text{MMS}} \, d\Omega \\
&= \int_{\theta=0}^{\pi} \int_{\phi=\pi}^{2\pi} \boldsymbol{\Omega} \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} \bar{\psi}_{\text{MMS}}(x, 1) \sin \theta \, d\phi \, d\theta \\
&= -\frac{1}{8} - \frac{1}{16} \sin\left(\frac{5\pi}{2}x + \frac{\pi}{4}\right) \sin\left(\frac{11\pi}{4}\right), \tag{5.20d}
\end{aligned}$$

and

$$\begin{aligned}
J_{\text{MMS}}^{\text{bottom}}(x) &= \int_{\boldsymbol{\Omega} \cdot \mathbf{n}_{\text{bottom}}} \boldsymbol{\Omega} \cdot \mathbf{n}_{\text{bottom}} \bar{\psi}_{\text{MMS}} \, d\Omega \\
&= \int_{\theta=0}^{\pi} \int_{\phi=0}^{\pi} \boldsymbol{\Omega} \cdot \begin{pmatrix} 0 \\ -1 \end{pmatrix} \bar{\psi}_{\text{MMS}}(x, 0) \sin \theta \, d\phi \, d\theta \\
&= -\frac{1}{8} - \frac{1}{16} \sin\left(\frac{5\pi}{2}x + \frac{\pi}{4}\right) \sin\left(\frac{\pi}{4}\right). \tag{5.20e}
\end{aligned}$$

In the preceding Eqs. (5.20b) to (5.20e),

- I used the notation $\bar{\psi}_{\text{MMS}}(a, b)$ as a concise expression for $\bar{\psi}_{\text{MMS}}\left((a, b)^T, \boldsymbol{\Omega}\right)$, which is just $\bar{\psi}_{\text{MMS}}(\mathbf{x}, \boldsymbol{\Omega})$ in Eq. (5.17) with $\mathbf{x} = (a, b)^T$, and
- Fig. 5.5 shows the boundary surface unit normal vectors \mathbf{n}_{dir} for $\text{dir} \in \{\text{right, left, top, bottom}\}$ as well as the azimuthal bounds of integration ϕ_1 and ϕ_2 in the integrals $\int_{\phi=\phi_1}^{\phi_2}$.

This concludes the presentation of the MMS data.

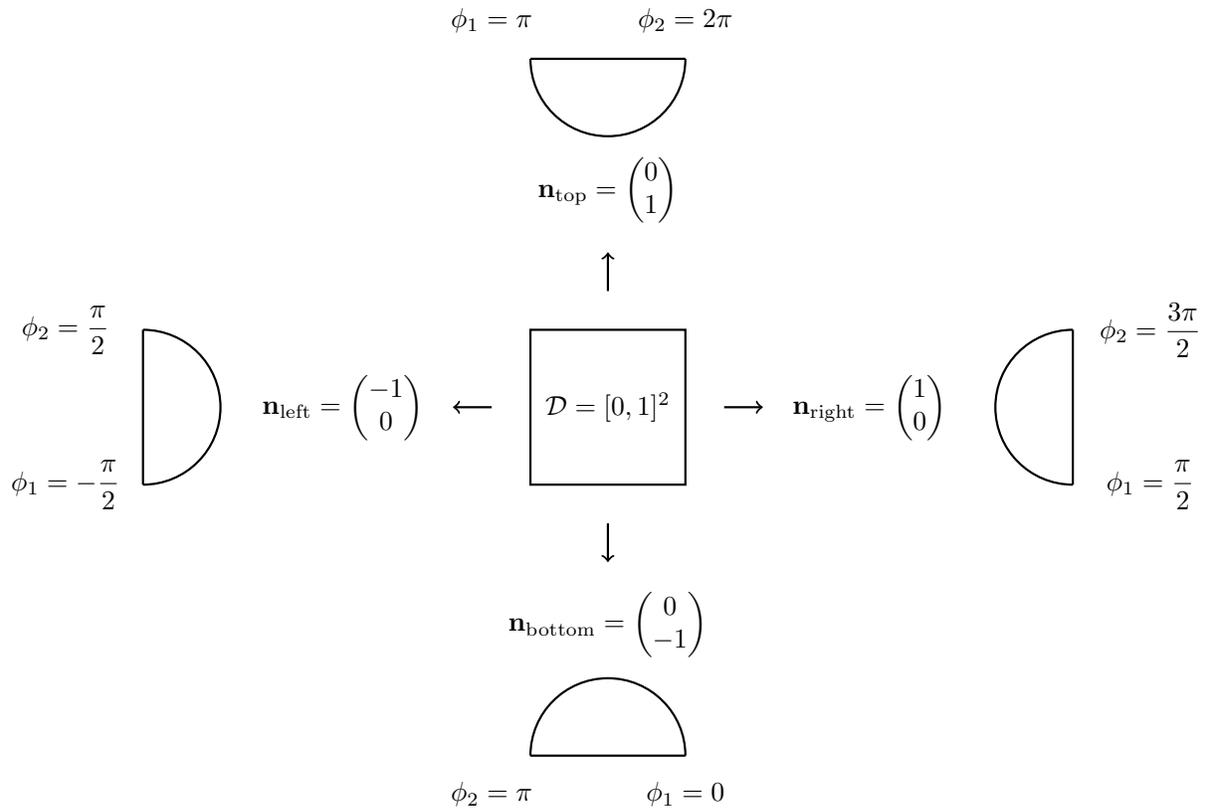


Figure 5.5: The domain $\mathcal{D} = [0, 1]^2$ with boundary surface unit normal vectors and azimuthal bounds of integration used for integrating Eqs. (5.20b) to (5.20e). The normals are the four arrows emanating from the edges of the square and the azimuthal bounds are the half circles facing the arrows. The half circles are projections of inflow hemispheres onto the xy -plane.

5.2 Thick Diffusion Limit Problem

The thick diffusion limit, which I described in section 1.3.6, is a challenging regime for transport methods. The TDL is characterized by the asymptotic scaling parameter, also known as the optical thickness parameter, $\epsilon \in (0, 1]$. The matter becomes arbitrarily optically thick in the limit $\epsilon \rightarrow 0$, as evidenced by the TDL scaling Eqs. (1.48a) to (1.48d).

The TDL is challenging because popular existing methods for solving the linear transport equation arising from TRT linearizations can be prohibitively expensive. Source iteration, which I described in section 1.3.3, converges arbitrarily slowly in the TDL. The runtime required to compute all Monte Carlo particle histories, which I described in section 1.3.4, is arbitrarily large in the TDL. My novel HSM method is a new way to address this challenge. The problems in this section are intended to show that I can calculate the diffusion limit solution accurately and efficiently using my HSM implementation, though we will see that this goal has yet to be achieved because my HSM implementation computes $\hat{\mathbf{T}}$, and $\text{Var}[\hat{\mathbf{T}}]$

is $O(1/\epsilon)$.

Let $\mathcal{D} = [0, 1]^2$ as in section 5.1, but now define the problem data in Eqs. (1.13a) and (1.13b) using the TDL scaling Eqs. (1.48a) to (1.48d). Thus, unlike in section 5.1 where σ_t and σ_s were constants, here they are functions of the TDL scaling parameter ϵ . Also, q in section 5.1 was Eq. (5.16a), a complicated expression that comes from substituting the MMS solution into Eq. (1.13a). Here, $q = \epsilon$, which is just the TDL scaling of q defined in Eq. (1.48d).

Consider four different optical thickness parameter values, 10^{-1} , 10^{-2} , 10^{-3} , and 10^{-4} . I run calculations for each value using 10 million HSM volume source particles, no boundary source particles, and a vacuum boundary condition. I compute the angle integrated intensity averaged over each element in an 8-by-8 mesh of squares of equal size. My solution is a surface on the xy -plane. I present HSM solution lineouts. A lineout is the line of intersection between the solution surface and a plane. A lineout provides a 1D picture of a 2D solution in the same way that a cross section provides a 2D picture of a 3D object. Fig. 5.6 shows that my HSM solution appears to converge to the diffusion limit solution as $\epsilon \rightarrow 0$ because the lineouts appear to approach an equilibrium solution in the sense that successive lineouts differ less for smaller ϵ .

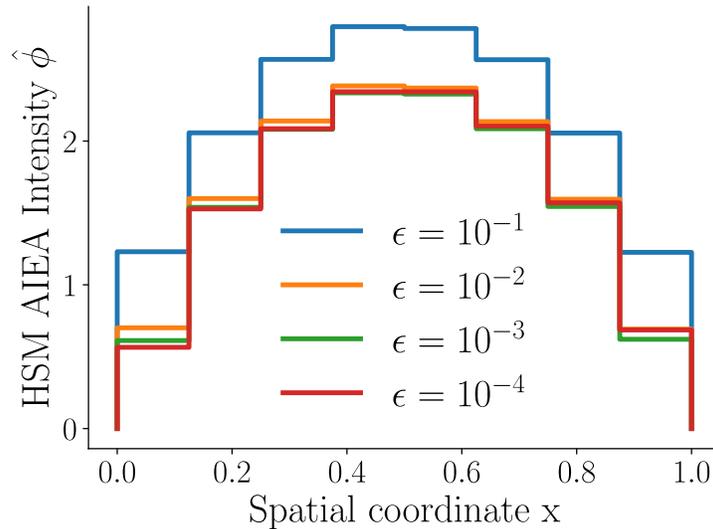


Figure 5.6: Lineouts at $y = 0.5$ of HSM angle integrated element averaged (AIEA) intensity for calculations using four different values of the TDL scaling parameter ϵ .

What about runtime? The convergence depicted by Fig. 5.6, which shows solution lineouts for four calculations using the HSM method, would be the same picture for unaccelerated Monte Carlo (UMC)¹. However, the runtimes of HSM and UMC in the TDL are very dif-

¹UMC is just the MC method for linear transport, which I described in section 1.3.4. I use the acronym UMC here to emphasize the fact that I have not employed any of the acceleration techniques described in section 1.3.4, such as RW, IMD, or DDMC.

ferent. Fig. 5.7 shows that the HSM runtime is constant with respect to the TDL scaling parameter, or $O(1)$, whereas the UMC runtime increases quadratically with the inverse of ϵ , or $O(1/\epsilon^2)$. Clearly, HSM outperforms UMC in terms of runtime as $\epsilon \rightarrow 0$.

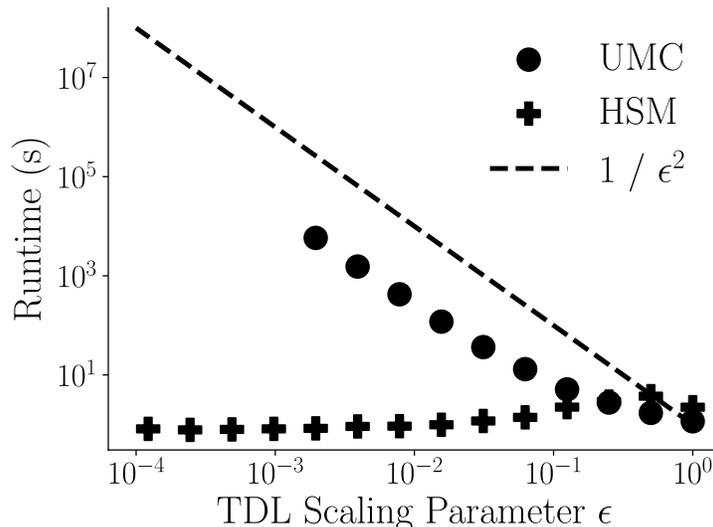


Figure 5.7: Runtime of unaccelerated Monte Carlo (UMC) and hybrid second moment (HSM) methods for calculations with different values of the TDL scaling parameter ϵ . The ϵ values are $2^0, 2^{-1}, 2^{-2}, \dots, 2^{-13}$.

What about variance? A figure of merit (FOM) is a quantity by which two numerical methods may be compared. A common FOM used with MC methods is the inverse of the product of the variance and the runtime,

$$\text{Figure of merit} = \frac{1}{\text{Var}[\cdot] T}, \quad (5.21)$$

where $\text{Var}[\cdot]$ is the variance of some estimator and T is the runtime of the calculation. Decreasing either $\text{Var}[\cdot]$ or T increases the FOM. Fig. 5.7 shows that $T_{\text{HSM}} < T_{\text{UMC}}$ after the optical thickness parameter value ϵ falls below a value somewhere between $\epsilon = 1/4$ and $1/8$, and that $T_{\text{HSM}} \ll T_{\text{UMC}}$ beginning at about $\epsilon = 1/32$, which is a result that holds for all smaller ϵ values. This factor of ϵ^2 decrease in the runtime would be useless if the variance of the HSM estimator increased by a factor of $1/\epsilon^2$ relative to the UMC variance, because Eq. (5.21) shows that the FOM would be left unchanged.

I calculate the variance of the HSM solution in the TDL by running 600 realizations of the same calculation with different PRNG seeds for each calculation. Each calculation computes the angle integrated intensity averaged over a single mesh element. The domain is $\mathcal{D} = [0, 1]^2$. I use 100 million HSM particles for each calculation. I consider five different optical thickness parameter values, $10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}$, and 10^{-5} . Thus, the total number of calculations is $600 \times 5 = 3000$. I estimate the variance as the sum of squares of the

difference of the HSM angle integrated element averaged intensity estimator realizations with the mean estimate,

$$\text{Var}[\hat{\phi}] \approx \frac{1}{600} \sum_{i=1}^{600} (\hat{\phi}_i - \Phi)^2, \quad (5.22)$$

where Φ is the mean estimate defined by,

$$\Phi = \frac{1}{600} \sum_{i=1}^{600} \hat{\phi}_i. \quad (5.23)$$

Fig. 5.8 shows that the empirical variance of the HSM estimator $\hat{\phi}$ in the TDL appears to be $O(1/\epsilon)$, which matches our expectation based on the fact that $\text{Var}[\hat{\phi}]$ should match $\text{Var}[\hat{\mathbf{T}}]$, which we derived to be $O(1/\epsilon)$ in section 3.6. The empirical support of our analytically derived result gives us confidence that the HSM FOM is $O(\epsilon)$ in the TDL.

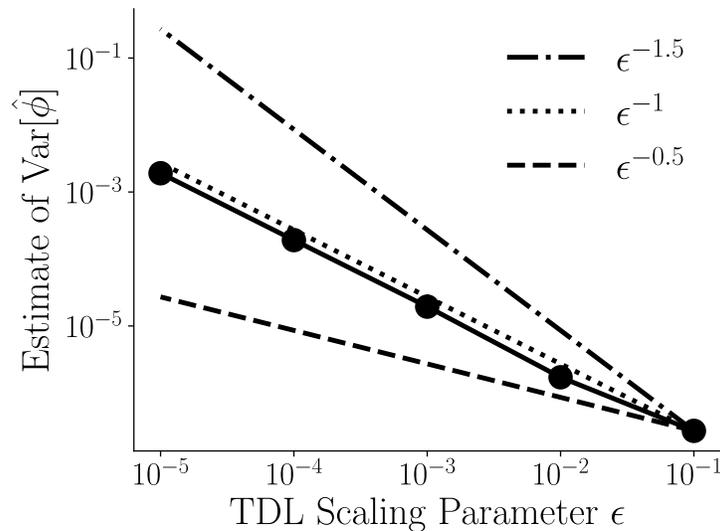


Figure 5.8: Estimate of the variance of the HSM angle integrated element averaged intensity.

5.3 Linearized Crooked Pipe Problem

Modeling fluid flow through a cylindrical pipe, airflow over an airfoil, and stress in a loaded beam are all classic problems in engineering. The field of radiative transfer has its own classic problems, such as radiative heating or cooling of a hydrogen cloud in the interstellar medium, and ionization front propagation in the same medium. An important example of a classic radiative transfer problem arising from the study of applied physics, which comes from the laboratory astrophysics community at LLNL, first appeared in a technical report

by Graziani and LeBlanc [93]. The authors describe radiation flow down a pipe with a bend. In the quarter century since the release of the report describing it, Crooked Pipe (CP) has become a classic problem in thermal radiative transfer. Developers of radiative transfer methods working in applied physics frequently use CP when they present their work.

In the CP problem description, a boundary source of radiation shines on one end of an optically-thin pipe surrounded by an optically-thick medium. Radiation flows preferentially through the pipe, around a bend in the pipe, and then leaves the problem through a vacuum boundary on the other end of the pipe. The radiative heating process in the CP problem occurs due to the physical processes which I described in Chapter 1. Fig. 1.1 lists some example processes by which the radiation gets absorbed and re-emitted.

Crooked Pipe is a time-dependent problem. The solution described in the problem statement is the temperature time-series at five fiducial points inside the pipe. I solve a steady-state simplification sometimes called the Linearized Crooked Pipe (LCP). The linearization is equivalent to calculating one very large timestep of a backward Euler time integrator. The timestep is such that $c\Delta t = 10^3$. The pipe is 1000 times optically-thinner than the surrounding medium. Fig. 5.9 depicts the geometry, material data, and boundary conditions for the problem. The boundary source is isotropic on the inflow hemisphere with magnitude,

$$\bar{\psi} = \begin{cases} \frac{1}{2\pi} & x = 0 \text{ and } -0.5 < y < 0.5, \\ 0 & \text{otherwise.} \end{cases} \quad (5.24)$$

Finally, there is an isotropic volume source with constant value $q = 10^{-7}$ everywhere in space.

Consider a mesh composed of squares of equal sizes. The minimal number of uniform squares required to tessellate the LCP geometry such that all material interfaces are on element boundaries is $14 \times 8 = 112$. Fig. 5.10 shows the aforementioned mesh elements, all of which contain material with single-valued properties, because each element contains either optically-thin pipe material, or optically-thick surrounding material, but not both. Fig. 5.11 shows the mesh after splitting each element into four squares of equal size to create a new mesh with 28×16 elements. Repeating this refinement operation three more times results in a 224×128 mesh, with 28,672 square elements, which is the mesh that I used in my calculations. I set the convergence threshold η in the convergence criterion Eq. (4.1) to 10^{-3} . I computed a piecewise constant angle integrated intensity using my $\hat{\phi}$ estimator of ϕ , which I derived in section 3.3.2. My estimator approximates the average of ϕ on each element, meaning that it is single-valued within a mesh element.

I ran three calculations. I computed the HSM solution using 8 billion simulation particles, I computed the UMC solution using 16 million simulation particles, and I computed the radiation diffusion approximation solution. Fig. 5.12 shows pseudocolor plots of the solutions. A “pseudocolor” plot uses different colors to represent various values of a specific quantity in an image or diagram. For example, it can illustrate flow velocity in a pipe, pressure on an airfoil, or, in this case, angle-integrated radiation intensity in a bent pipe. In this representation, higher intensities are depicted with bright orange, with the highest intensities

appearing as a very bright yellow, while lower intensities are shown in dark purple, and the lowest are rendered in black.

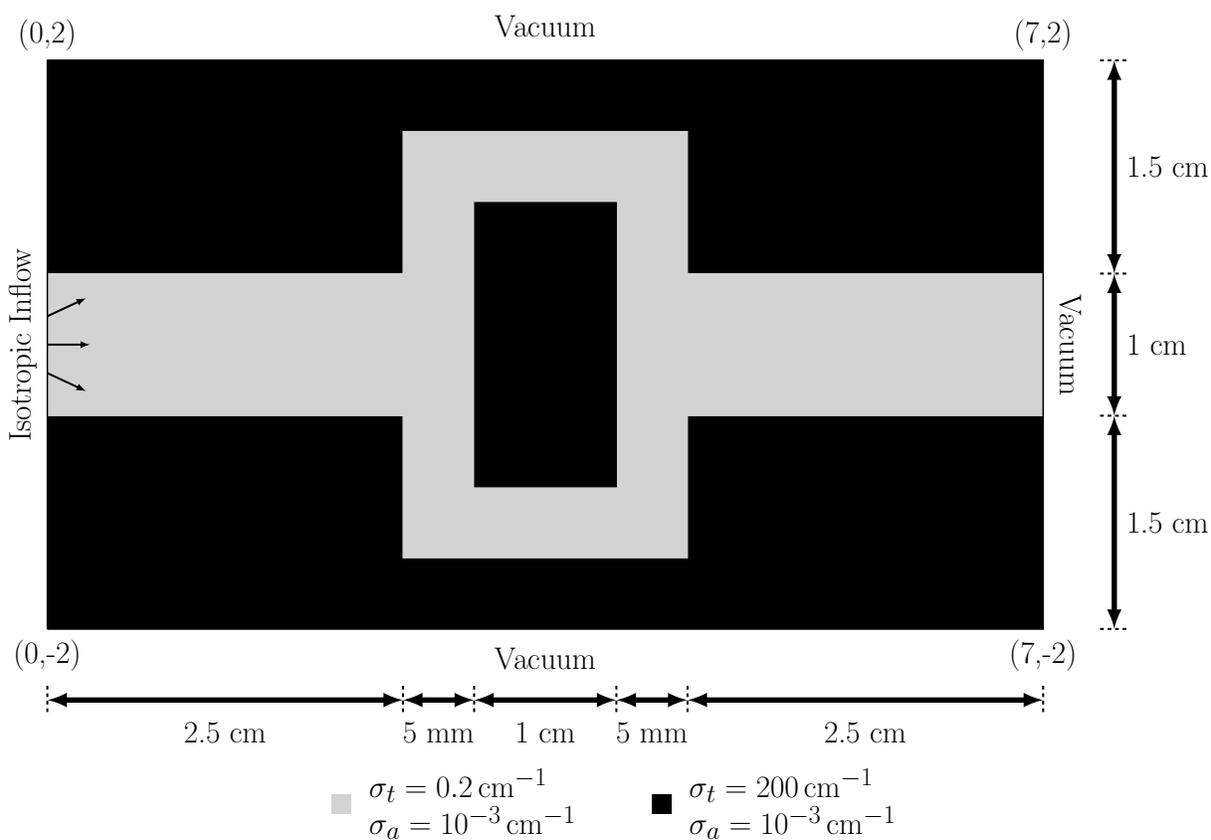


Figure 5.9: Linearized Crooked Pipe geometry, material data, and boundary conditions [39].

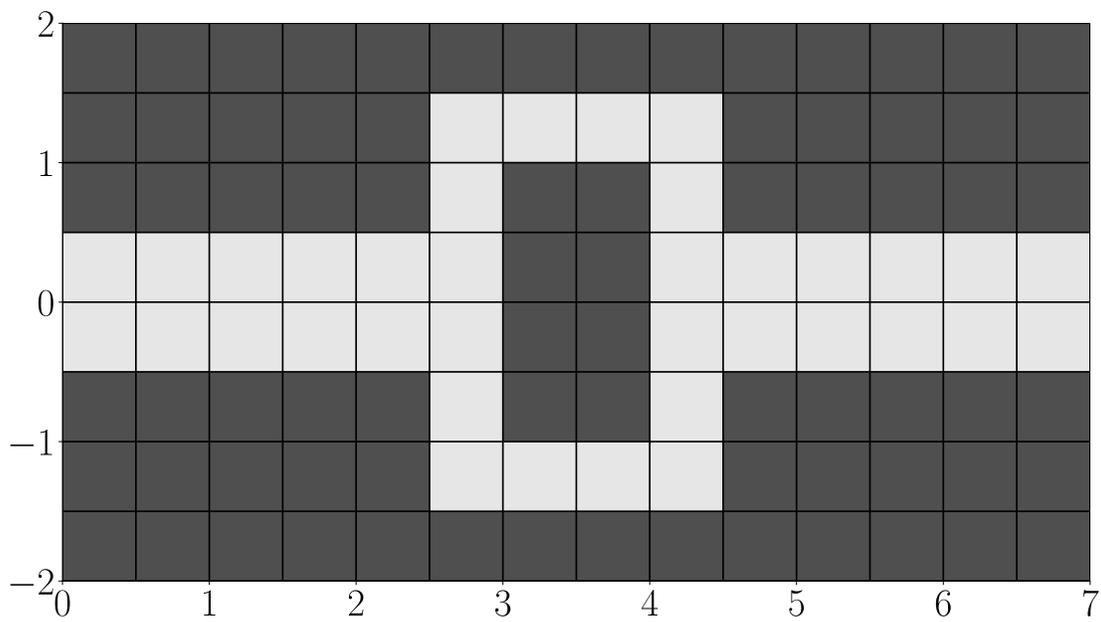


Figure 5.10: Uniform mesh of the Linearized Crooked Pipe geometry, $|\mathcal{T}| = 14 \times 8 = 112$.

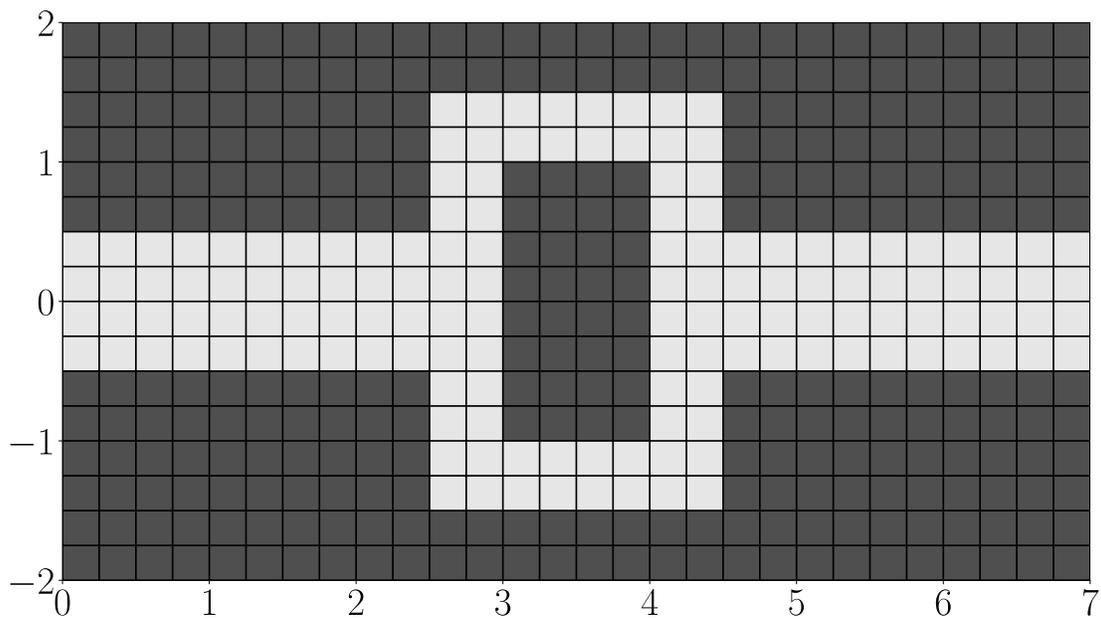
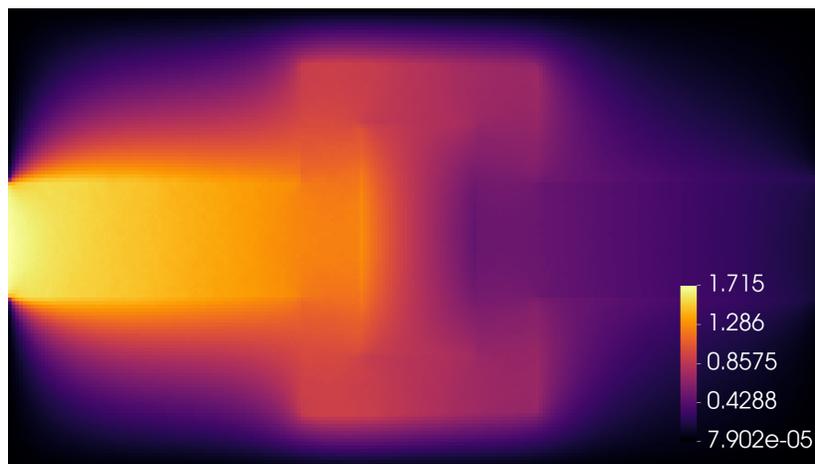
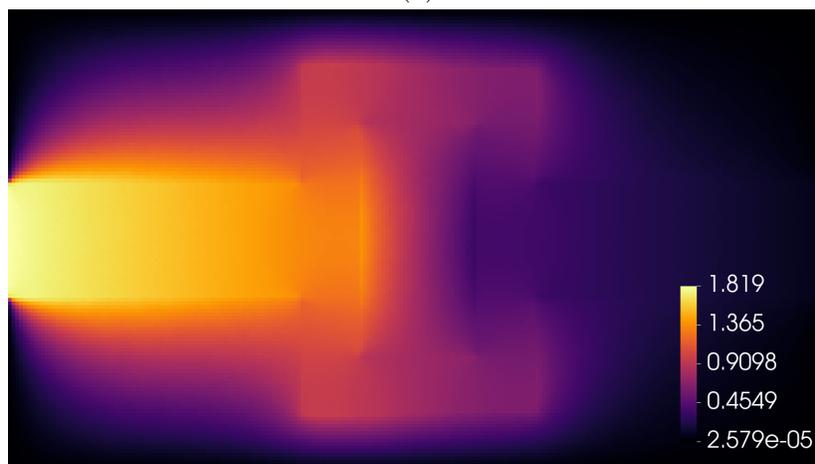


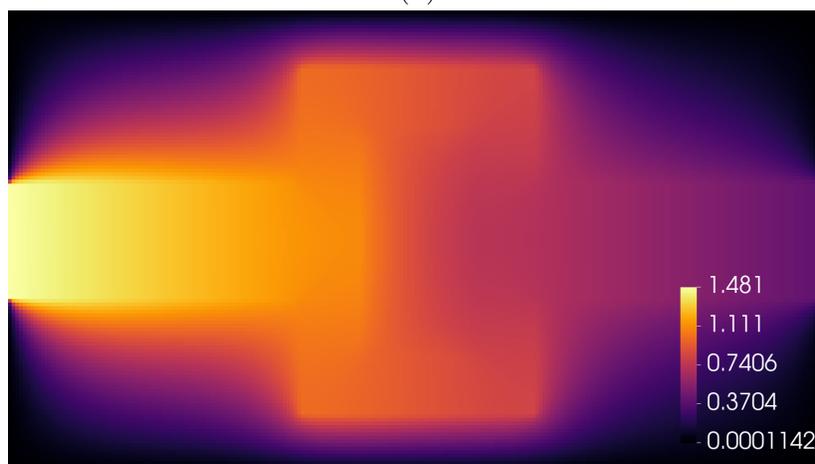
Figure 5.11: Refined mesh, $|\mathcal{T}| = 28 \times 16 = 448$.



(a)



(b)



(c)

Figure 5.12: HSM (a), UMC (b), and diffusion (c) solutions for the linearized crooked pipe.

Four comments regarding the solutions in Fig. 5.12 are:

1. My UMC solution closely matches the S_N solution in Fig 9 (b) in Olivier *et al.* [39]. Both have bright inner wall illumination and dark outer wall shadows characteristic of transport solutions. Additionally, the maximum colorbar values agree to within 1.06%.
2. My diffusion solution closely matches the diffusion solution in Fig 9 (c) in Olivier *et al.* [39]. The radiation diffusion approximation gives a solution with a substantial approximation error, in this case 20%, as judged by the ratio of the colorbar maximum values (assuming that the UMC solution is the correct solution). The diffusion approximation radiation wavefront speed also appears to be fast, as expected, relative to transport, as evidenced by the substantially enhanced brightness of the outer wall and the region in between the outer wall and the far end of the pipe.
3. My radiation diffusion approximation calculation is simply my SMM solver from my HSM implementation with the transport correction set to zero. This is because SMM reduces to radiation diffusion when $\mathbf{T} = \mathbf{0}$ and $\beta = 0$.
4. Running fewer than 8 billion particles in the HSM calculation, or fewer than 16 million particles in the UMC calculation, causes visible noise to appear in the solution. The factor of 500 more particles required for the HSM calculation is an undesirable consequence of the $\text{Var}[\hat{\phi}] = O(1/\epsilon)$ result that I derived in section 3.6, and for which I provided supporting empirical evidence in Fig. 5.8.

5.3.1 Particle-Equivalent Comparisons with Varied Optical Thicknesses

This section shows plots of the HSM and UMC solutions for the same number of particles. I run calculations for the LCP problem, as well as three additional problems, which are just the LCP problem but with different optical thickness ratios. The optical thickness ratio of the materials from the LCP problem definition in [39] are,

$$\frac{\sigma_t^{\text{thick}}}{\sigma_t^{\text{thin}}} = \frac{200}{0.2} = 1000, \quad (5.25)$$

where “thin” denotes the pipe material and “thick” denotes the material surrounding the pipe. In addition to running with the original optical thickness ratio, I run with new values of σ_t^{thick} and σ_t^{thin} that provide optical thickness ratios of 2, 10, and 100. The opacities for the four problems are listed in table 5.1. All of the entries in table 5.1 have dimensions of inverse centimeters.

I ran HSM and UMC calculations for the 2, 10, 100, and 1000 optical thickness ratio LCP problems using 1 million, 4 million, and 16 million particles. Thus, I ran the following

24 calculations,

$$\underbrace{\{\text{HSM, UMC}\}}_{2 \text{ methods}} \times \underbrace{\{1\text{e}6, 4\text{e}6, 16\text{e}6\}}_{3 \text{ particle counts}} \times \underbrace{\{2, 10, 100, 1000\}}_{4 \text{ problem definitions}},$$

where “1e6” is 1 million, “4e6” is 4 million, and “16e6” is 16 million.

Observe from Fig. 5.9 that everything in the LCP problem definition—geometry, materials, boundary conditions—is symmetric about the x -axis. The HSM and UMC solutions preserve the symmetry up to the Monte Carlo noise level, which allows us to compare the two solutions for each problem by cutting the pseudocolor plots in half along the line $y = 0$. In Figs. 5.13 to 5.16, I show HSM on top of UMC.

<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;"></th> <th style="width: 20%;">Thick</th> <th style="width: 20%;">Thin</th> <th style="width: 20%;">Ratio</th> </tr> </thead> <tbody> <tr> <td>σ_a</td> <td>1</td> <td>0.5</td> <td>2</td> </tr> <tr> <td>σ_s</td> <td>1</td> <td>0.5</td> <td>2</td> </tr> <tr> <td>σ_t</td> <td>2</td> <td>1</td> <td>2</td> </tr> </tbody> </table> <p style="text-align: center;">(a) $\sigma_t^{\text{thick}}/\sigma_t^{\text{thin}} = 2$</p>		Thick	Thin	Ratio	σ_a	1	0.5	2	σ_s	1	0.5	2	σ_t	2	1	2	<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;"></th> <th style="width: 20%;">Thick</th> <th style="width: 20%;">Thin</th> <th style="width: 20%;">Ratio</th> </tr> </thead> <tbody> <tr> <td>σ_a</td> <td>0.00001</td> <td>0.001</td> <td>0.01</td> </tr> <tr> <td>σ_s</td> <td>1.99999</td> <td>0.199</td> <td>10.05</td> </tr> <tr> <td>σ_t</td> <td>2</td> <td>0.2</td> <td>10</td> </tr> </tbody> </table> <p style="text-align: center;">(b) $\sigma_t^{\text{thick}}/\sigma_t^{\text{thin}} = 10$</p>		Thick	Thin	Ratio	σ_a	0.00001	0.001	0.01	σ_s	1.99999	0.199	10.05	σ_t	2	0.2	10
	Thick	Thin	Ratio																														
σ_a	1	0.5	2																														
σ_s	1	0.5	2																														
σ_t	2	1	2																														
	Thick	Thin	Ratio																														
σ_a	0.00001	0.001	0.01																														
σ_s	1.99999	0.199	10.05																														
σ_t	2	0.2	10																														
<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;"></th> <th style="width: 20%;">Thick</th> <th style="width: 20%;">Thin</th> <th style="width: 20%;">Ratio</th> </tr> </thead> <tbody> <tr> <td>σ_a</td> <td>0.0001</td> <td>0.001</td> <td>0.1</td> </tr> <tr> <td>σ_s</td> <td>19.9999</td> <td>0.199</td> <td>100.5</td> </tr> <tr> <td>σ_t</td> <td>20</td> <td>0.2</td> <td>100</td> </tr> </tbody> </table> <p style="text-align: center;">(c) $\sigma_t^{\text{thick}}/\sigma_t^{\text{thin}} = 100$</p>		Thick	Thin	Ratio	σ_a	0.0001	0.001	0.1	σ_s	19.9999	0.199	100.5	σ_t	20	0.2	100	<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;"></th> <th style="width: 20%;">Thick</th> <th style="width: 20%;">Thin</th> <th style="width: 20%;">Ratio</th> </tr> </thead> <tbody> <tr> <td>σ_a</td> <td>0.001</td> <td>0.001</td> <td>1</td> </tr> <tr> <td>σ_s</td> <td>199.999</td> <td>0.199</td> <td>1005</td> </tr> <tr> <td>σ_t</td> <td>200</td> <td>0.2</td> <td>1000</td> </tr> </tbody> </table> <p style="text-align: center;">(d) $\sigma_t^{\text{thick}}/\sigma_t^{\text{thin}} = 1000$</p>		Thick	Thin	Ratio	σ_a	0.001	0.001	1	σ_s	199.999	0.199	1005	σ_t	200	0.2	1000
	Thick	Thin	Ratio																														
σ_a	0.0001	0.001	0.1																														
σ_s	19.9999	0.199	100.5																														
σ_t	20	0.2	100																														
	Thick	Thin	Ratio																														
σ_a	0.001	0.001	1																														
σ_s	199.999	0.199	1005																														
σ_t	200	0.2	1000																														

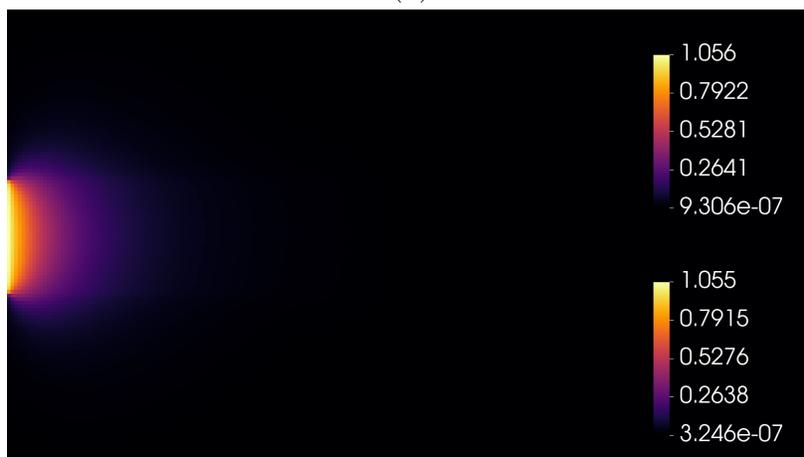
Table 5.1: Material data for four distinct problems, each with a unique optical thickness ratio, $\sigma_t^{\text{thick}}/\sigma_t^{\text{thin}}$.



(a)

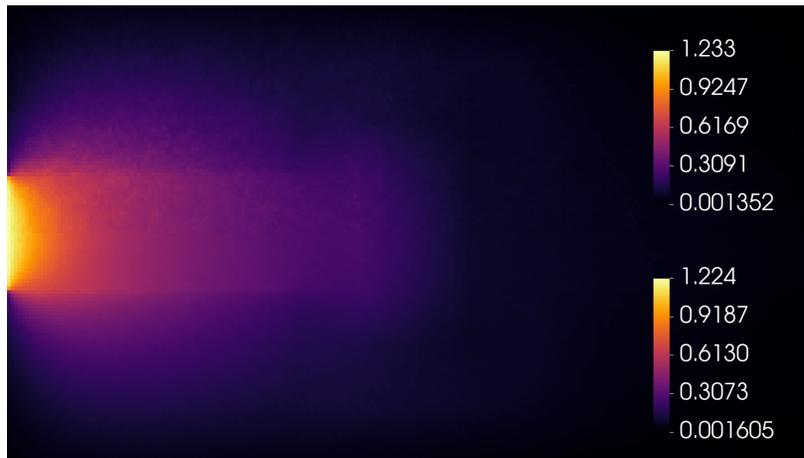


(b)

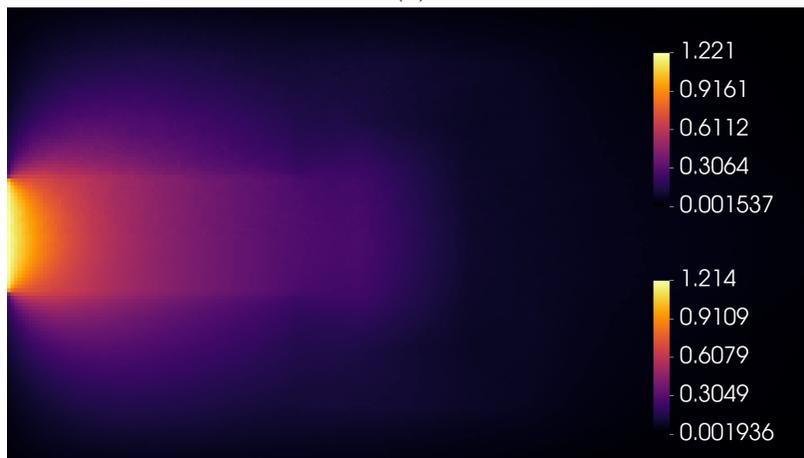


(c)

Figure 5.13: Solutions using 1e6 (a), 4e6 (b), and 16e6 (c) particles with HSM (top of a,b,c) and UMC (bottom of a,b,c) for the LCP problem with $\sigma_t^{\text{thick}}/\sigma_t^{\text{thin}} = 2$.



(a)

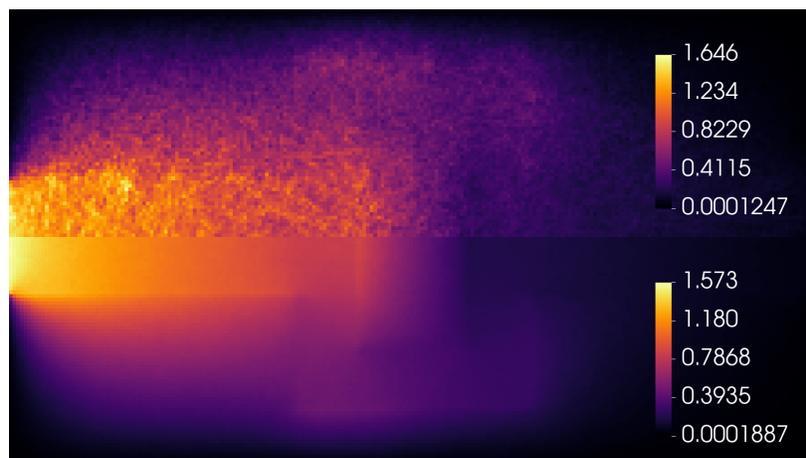


(b)

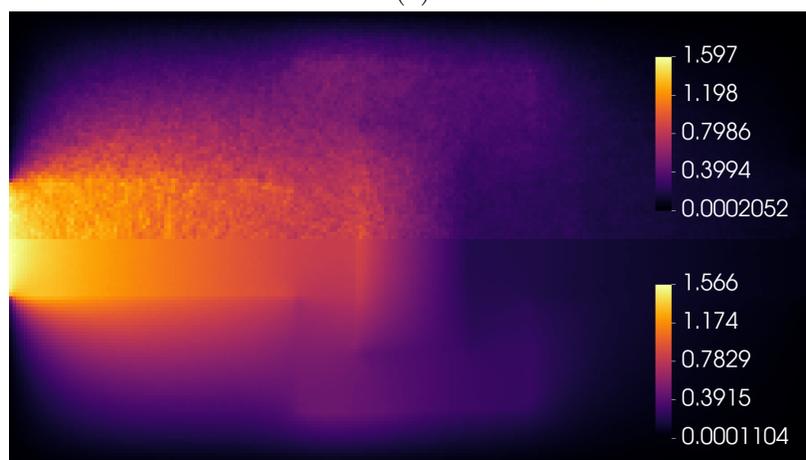


(c)

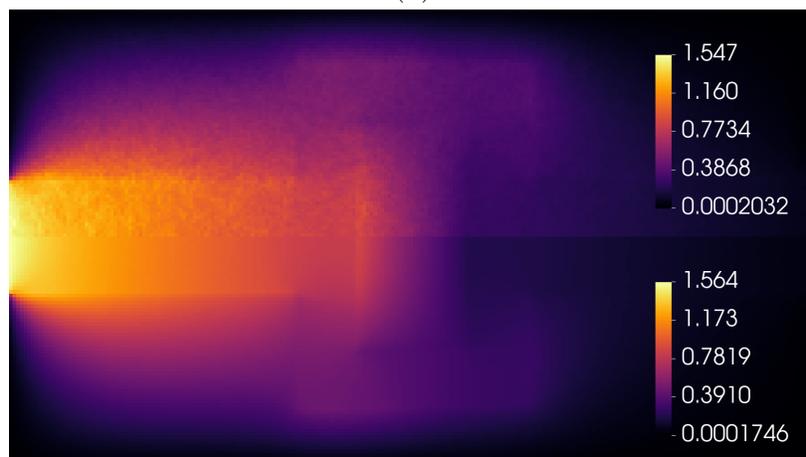
Figure 5.14: Solutions using 1e6 (a), 4e6 (b), and 16e6 (c) particles with HSM (top of a,b,c) and UMC (bottom of a,b,c) for the LCP problem with $\sigma_t^{\text{thick}}/\sigma_t^{\text{thin}} = 10$.



(a)

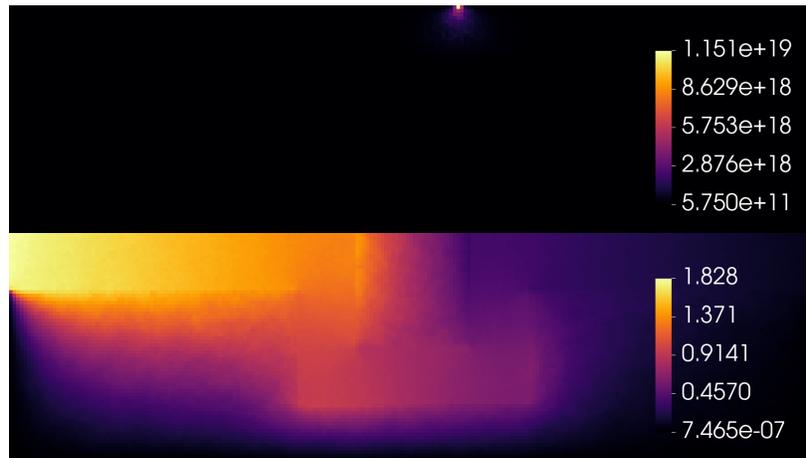


(b)

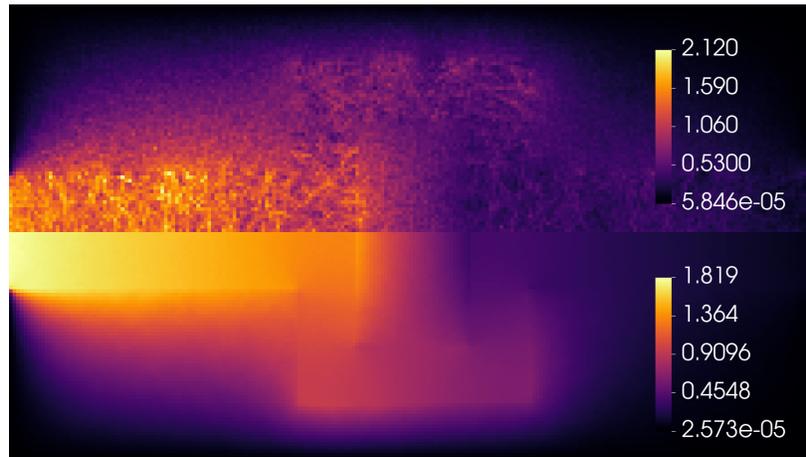


(c)

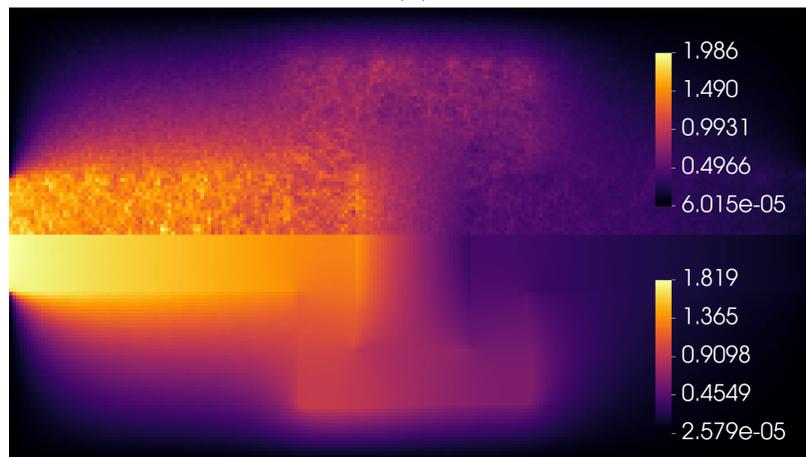
Figure 5.15: Solutions using $1e6$ (a), $4e6$ (b), and $16e6$ (c) particles with HSM (top of a,b,c) and UMC (bottom of a,b,c) for the LCP problem with $\sigma_t^{\text{thick}}/\sigma_t^{\text{thin}} = 100$.



(a)



(b)



(c)

Figure 5.16: Solutions using $1e6$ (a), $4e6$ (b), and $16e6$ (c) particles with HSM (top of a,b,c) and UMC (bottom of a,b,c) for the LCP problem with $\sigma_t^{\text{thick}}/\sigma_t^{\text{thin}} = 1000$.

Four comments regarding the results in Figs. 5.13 to 5.16 are:

1. The $\sigma_t^{\text{thick}}/\sigma_t^{\text{thin}} = 2$ results in Fig. 5.13 show excellent agreement between HSM and UMC for all 3 particle counts with no visible noise in any of the solutions. The radiation exhibits minimal propagation.
2. The $\sigma_t^{\text{thick}}/\sigma_t^{\text{thin}} = 10$ results in Fig. 5.14 show very good agreement between HSM and UMC for all 3 particle counts. The HSM solution has visible noise at 1e6 particles that becomes less visible at 4e6 and is indiscernible at 16e6 particles. The UMC solutions have no visible noise. The radiation propagates to the front wall.
3. The $\sigma_t^{\text{thick}}/\sigma_t^{\text{thin}} = 100$ results in Fig. 5.15 show some agreement between HSM and UMC which increases with particle count. The HSM solution has significant noise at 1e6 particles. The noise is less severe at 4e6 and 16e6 particles, but remains visible. The UMC solutions have no visible noise. The radiation propagates to the back wall.
4. The $\sigma_t^{\text{thick}}/\sigma_t^{\text{thin}} = 1000$ results in Fig. 5.16 show significant differences between HSM and UMC, though the disagreement decreases with particle count. The HSM solution at 1e6 is dark except for a single boundary element with a non-physical solution of approximately 10^{19} , indicating that the iteration diverged. The HSM solutions at 4e6 and 16e6 particles look reasonable, but very noisy. The UMC solution has significant noise at 1e6 particles, less at 4e6, and indiscernible noise at 16e6 particles. The radiation propagates beyond the back wall.

The HSM solution has much more noise than UMC for an equivalent number of simulation particles when the material is optically thick. The issue causing the noise is my choice of estimators for the SMM data: $\hat{\mathbf{T}}$ is noisy in optically-thicker materials because $\text{Var}[\hat{\mathbf{T}}]$ is $O(1/\epsilon)$, as I derived in section 3.6 and showed in Fig. 5.8.

5.3.2 Confirming Avoidance of Noise Amplification

This section is dedicated to a brief examination of the noise issue in the HSM solution to the LCP problem, in order to confirm that the mixed finite element solve of the SMM system does not amplify the Monte Carlo noise in the SMM data estimators $\hat{\mathbf{T}}$ and $\hat{\beta}$. The calculations in this section use HSM to solve the LCP problem as defined in [39], for which $\sigma_t^{\text{thick}}/\sigma_t^{\text{thin}} = 1000$. I use 500 million particles, which is a factor of 16 fewer particles than the 8 billion particles that I used to produce the visually-noiseless HSM solution in Fig. 5.12 (a). I plot intermediate solutions during the HSM calculation before convergence in order to show when the noise appears and how it changes during the HSM iteration.

Fig. 5.17 shows the logic in Algorithm 3 for computing the HSM solution alongside pseudocolor plots of the unscattered estimator $\hat{\phi}^{(0)}$, the estimator first iterate $\hat{\phi}^{(1)}$, and the estimator at convergence $\hat{\phi}^{(12)}$. Fig. 5.18 shows the unscattered estimator $\hat{\phi}^{(0)}$ and the estimator first iterate $\hat{\phi}^{(1)}$ again, but with the second moment system solution iterate $\varphi^{(1)}$ interposed between them.

Algorithm Hybrid Second Moment

```

1: Input: user-provided boolean value HSM
2: if not HSM then
3:   scattering_events  $\leftarrow$  true
4:    $\hat{\phi} \leftarrow$  mc(q, scattering_events,  $\bar{\psi}$ )
5:   return  $\hat{\phi}$ 
6: end if
7: scattering_events  $\leftarrow$  false
8:  $\hat{\phi}^{(0)}, \hat{\mathbf{T}}^{(0)}, \hat{\beta}^{(0)} \leftarrow$  mc(q, scattering_events,  $\bar{\psi}$ )
9: i  $\leftarrow$  1
10: while not converged( $\hat{\phi}^{(i-1)}, \hat{\phi}^{(i)}$ ) do
11:    $\varphi^{(i)} \leftarrow$  sm(Q0, Q1,  $\hat{\mathbf{T}}^{(i-1)}, \hat{\beta}^{(i-1)}$ )
12:    $\hat{\phi}_{\text{temp}}, \hat{\mathbf{T}}_{\text{temp}}, \hat{\beta}_{\text{temp}} \leftarrow$  mc( $\varphi^{(i)}$ , scattering_events)
13:    $\hat{\phi}^{(i)} \leftarrow \hat{\phi}^{(0)} + \hat{\phi}_{\text{temp}}$ 
14:    $\hat{\mathbf{T}}^{(i)} \leftarrow \hat{\mathbf{T}}^{(0)} + \hat{\mathbf{T}}_{\text{temp}}$ 
15:    $\hat{\beta}^{(i)} \leftarrow \hat{\beta}^{(0)} + \hat{\beta}_{\text{temp}}$ 
16:   i  $\leftarrow$  i + 1
17: end while
18: return  $\hat{\phi}^{(i)}$ 

```



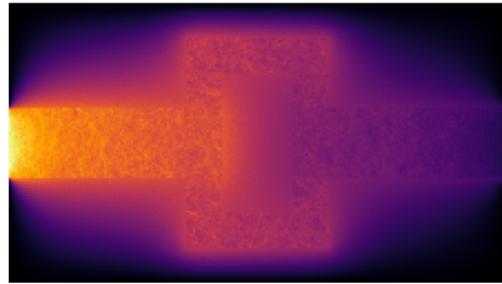
(a)

Algorithm Hybrid Second Moment

```

1: Input: user-provided boolean value HSM
2: if not HSM then
3:   scattering_events  $\leftarrow$  true
4:    $\hat{\phi} \leftarrow$  mc(q, scattering_events,  $\bar{\psi}$ )
5:   return  $\hat{\phi}$ 
6: end if
7: scattering_events  $\leftarrow$  false
8:  $\hat{\phi}^{(0)}, \hat{\mathbf{T}}^{(0)}, \hat{\beta}^{(0)} \leftarrow$  mc(q, scattering_events,  $\bar{\psi}$ )
9: i  $\leftarrow$  1
10: while not converged( $\hat{\phi}^{(i-1)}, \hat{\phi}^{(i)}$ ) do
11:    $\varphi^{(i)} \leftarrow$  sm(Q0, Q1,  $\hat{\mathbf{T}}^{(i-1)}, \hat{\beta}^{(i-1)}$ )
12:    $\hat{\phi}_{\text{temp}}, \hat{\mathbf{T}}_{\text{temp}}, \hat{\beta}_{\text{temp}} \leftarrow$  mc( $\varphi^{(i)}$ , scattering_events)
13:    $\hat{\phi}^{(i)} \leftarrow \hat{\phi}^{(0)} + \hat{\phi}_{\text{temp}}$ 
14:    $\hat{\mathbf{T}}^{(i)} \leftarrow \hat{\mathbf{T}}^{(0)} + \hat{\mathbf{T}}_{\text{temp}}$ 
15:    $\hat{\beta}^{(i)} \leftarrow \hat{\beta}^{(0)} + \hat{\beta}_{\text{temp}}$ 
16:   i  $\leftarrow$  i + 1
17: end while
18: return  $\hat{\phi}^{(i)}$ 

```



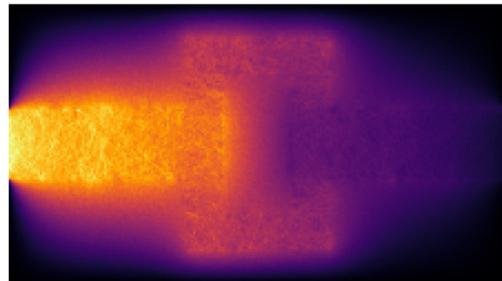
(b)

Algorithm Hybrid Second Moment

```

1: Input: user-provided boolean value HSM
2: if not HSM then
3:   scattering_events  $\leftarrow$  true
4:    $\hat{\phi} \leftarrow$  mc(q, scattering_events,  $\bar{\psi}$ )
5:   return  $\hat{\phi}$ 
6: end if
7: scattering_events  $\leftarrow$  false
8:  $\hat{\phi}^{(0)}, \hat{\mathbf{T}}^{(0)}, \hat{\beta}^{(0)} \leftarrow$  mc(q, scattering_events,  $\bar{\psi}$ )
9: i  $\leftarrow$  1
10: while not converged( $\hat{\phi}^{(i-1)}, \hat{\phi}^{(i)}$ ) do
11:    $\varphi^{(i)} \leftarrow$  sm(Q0, Q1,  $\hat{\mathbf{T}}^{(i-1)}, \hat{\beta}^{(i-1)}$ )
12:    $\hat{\phi}_{\text{temp}}, \hat{\mathbf{T}}_{\text{temp}}, \hat{\beta}_{\text{temp}} \leftarrow$  mc( $\varphi^{(i)}$ , scattering_events)
13:    $\hat{\phi}^{(i)} \leftarrow \hat{\phi}^{(0)} + \hat{\phi}_{\text{temp}}$ 
14:    $\hat{\mathbf{T}}^{(i)} \leftarrow \hat{\mathbf{T}}^{(0)} + \hat{\mathbf{T}}_{\text{temp}}$ 
15:    $\hat{\beta}^{(i)} \leftarrow \hat{\beta}^{(0)} + \hat{\beta}_{\text{temp}}$ 
16:   i  $\leftarrow$  i + 1
17: end while
18: return  $\hat{\phi}^{(i)}$ 

```



(c)

Figure 5.17: The HSM estimator $\hat{\phi}$ for the LCP problem before iterating (a), after the first cycle (b), and at convergence (c), where 12 cycles were required to converge HSM using 500 million particles.

Algorithm Hybrid Second Moment

```

1: Input: user-provided boolean value HSM
2: if not HSM then
3:   scattering_events  $\leftarrow$  true
4:    $\hat{\phi} \leftarrow$  mc(q, scattering_events,  $\bar{\psi}$ )
5:   return  $\hat{\phi}$ 
6: end if
7: scattering_events  $\leftarrow$  false
8:  $\hat{\phi}^{(0)}, \hat{\mathbf{T}}^{(0)}, \hat{\beta}^{(0)} \leftarrow$  mc(q, scattering_events,  $\bar{\psi}$ )
9: i  $\leftarrow$  1
10: while not converged( $\hat{\phi}^{(i-1)}, \hat{\phi}^{(i)}$ ) do
11:    $\varphi^{(i)} \leftarrow$  sm(Q0, Q1,  $\hat{\mathbf{T}}^{(i-1)}, \hat{\beta}^{(i-1)}$ )
12:    $\hat{\phi}_{\text{temp}}, \hat{\mathbf{T}}_{\text{temp}}, \hat{\beta}_{\text{temp}} \leftarrow$  mc( $\varphi^{(i)}$ , scattering_events)
13:    $\hat{\phi}^{(i)} \leftarrow \hat{\phi}^{(0)} + \hat{\phi}_{\text{temp}}$ 
14:    $\hat{\mathbf{T}}^{(i)} \leftarrow \hat{\mathbf{T}}^{(0)} + \hat{\mathbf{T}}_{\text{temp}}$ 
15:    $\hat{\beta}^{(i)} \leftarrow \hat{\beta}^{(0)} + \hat{\beta}_{\text{temp}}$ 
16:   i  $\leftarrow$  i + 1
17: end while
18: return  $\hat{\phi}^{(i)}$ 

```



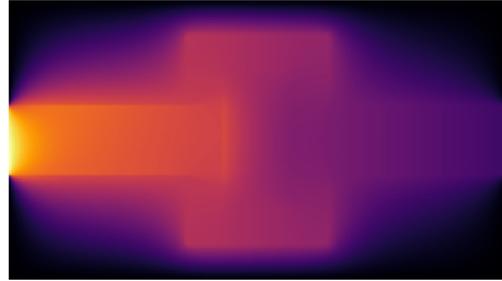
(a)

Algorithm Hybrid Second Moment

```

1: Input: user-provided boolean value HSM
2: if not HSM then
3:   scattering_events  $\leftarrow$  true
4:    $\hat{\phi} \leftarrow$  mc(q, scattering_events,  $\bar{\psi}$ )
5:   return  $\hat{\phi}$ 
6: end if
7: scattering_events  $\leftarrow$  false
8:  $\hat{\phi}^{(0)}, \hat{\mathbf{T}}^{(0)}, \hat{\beta}^{(0)} \leftarrow$  mc(q, scattering_events,  $\bar{\psi}$ )
9: i  $\leftarrow$  1
10: while not converged( $\hat{\phi}^{(i-1)}, \hat{\phi}^{(i)}$ ) do
11:    $\varphi^{(i)} \leftarrow$  sm(Q0, Q1,  $\hat{\mathbf{T}}^{(i-1)}, \hat{\beta}^{(i-1)}$ )
12:    $\hat{\phi}_{\text{temp}}, \hat{\mathbf{T}}_{\text{temp}}, \hat{\beta}_{\text{temp}} \leftarrow$  mc( $\varphi^{(i)}$ , scattering_events)
13:    $\hat{\phi}^{(i)} \leftarrow \hat{\phi}^{(0)} + \hat{\phi}_{\text{temp}}$ 
14:    $\hat{\mathbf{T}}^{(i)} \leftarrow \hat{\mathbf{T}}^{(0)} + \hat{\mathbf{T}}_{\text{temp}}$ 
15:    $\hat{\beta}^{(i)} \leftarrow \hat{\beta}^{(0)} + \hat{\beta}_{\text{temp}}$ 
16:   i  $\leftarrow$  i + 1
17: end while
18: return  $\hat{\phi}^{(i)}$ 

```



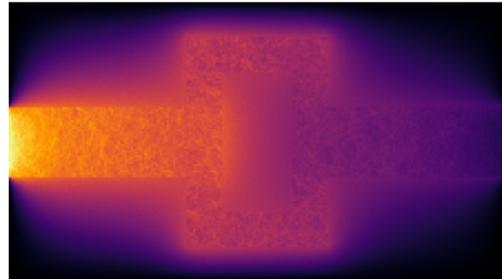
(b)

Algorithm Hybrid Second Moment

```

1: Input: user-provided boolean value HSM
2: if not HSM then
3:   scattering_events  $\leftarrow$  true
4:    $\hat{\phi} \leftarrow$  mc(q, scattering_events,  $\bar{\psi}$ )
5:   return  $\hat{\phi}$ 
6: end if
7: scattering_events  $\leftarrow$  false
8:  $\hat{\phi}^{(0)}, \hat{\mathbf{T}}^{(0)}, \hat{\beta}^{(0)} \leftarrow$  mc(q, scattering_events,  $\bar{\psi}$ )
9: i  $\leftarrow$  1
10: while not converged( $\hat{\phi}^{(i-1)}, \hat{\phi}^{(i)}$ ) do
11:    $\varphi^{(i)} \leftarrow$  sm(Q0, Q1,  $\hat{\mathbf{T}}^{(i-1)}, \hat{\beta}^{(i-1)}$ )
12:    $\hat{\phi}_{\text{temp}}, \hat{\mathbf{T}}_{\text{temp}}, \hat{\beta}_{\text{temp}} \leftarrow$  mc( $\varphi^{(i)}$ , scattering_events)
13:    $\hat{\phi}^{(i)} \leftarrow \hat{\phi}^{(0)} + \hat{\phi}_{\text{temp}}$ 
14:    $\hat{\mathbf{T}}^{(i)} \leftarrow \hat{\mathbf{T}}^{(0)} + \hat{\mathbf{T}}_{\text{temp}}$ 
15:    $\hat{\beta}^{(i)} \leftarrow \hat{\beta}^{(0)} + \hat{\beta}_{\text{temp}}$ 
16:   i  $\leftarrow$  i + 1
17: end while
18: return  $\hat{\phi}^{(i)}$ 

```



(c)

Figure 5.18: The HSM estimator $\hat{\phi}$ for the LCP problem before iterating (a), φ in the first cycle (b), and $\hat{\phi}$ after cycle 1 (c). The pseudocolor plots in (a) and (c) are identical to the pseudocolor plots in Fig. 5.17 (a) and (b), respectively.

Three observations about Figs. 5.17 and 5.18 are:

- i. The unscattered estimator $\hat{\phi}^{(0)}$, plotted in Fig. 5.17 (a) and Fig. 5.18 (a), illuminates the pipe so weakly that the front wall is not even visible. Consequently, a substantial amount of work in the HSM algorithm is left to the iteration of the scattering source. Additionally, the unscattered estimator $\hat{\phi}^{(0)}$ does not show the noise issue.
- ii. The noise issue is apparent in both $\hat{\phi}^{(1)}$ and $\hat{\phi}^{(12)}$, which are plotted in Fig. 5.17 (b) and (c), respectively. However, the noise issue does not appear to worsen between cycle 1 and 12, because $\hat{\phi}^{(12)}$ appears no more noisy than $\hat{\phi}^{(1)}$.
- iii. The second moment solution iterate $\varphi^{(1)}$, plotted in Fig. 5.18 (b), does not show the noise issue. Thus, the noise problem results from the transport of the scattering source, not the moment system solve. Avoiding differentiation of the correction tensor by solving the first order moment system using a FEM to offload the derivative appears to be effective for the prevention of noise amplification.

I conclude that the noise issue is a result of the calculation and transport of the scattering source. One final observation is that the total weight of the simulation particles before iterating is only 0.5, whereas it is about 1,400 during iteration, plus or minus a few percent because the weight varies by a few percent over the iteration.

Recall from section 3.6 that the variance of my $\hat{\mathbf{T}}$ estimator of the SMM transport correction tensor is $O(1/\epsilon)$ in the TDL, as supported by the empirical result in Fig. 5.8. I derived $\hat{\mathbf{R}}$ in section 3.7, which is an alternative estimator for \mathbf{T} that is $O(\epsilon)$ in the TDL. I derived another alternative estimator for \mathbf{T} , called the deviatoric estimator, which is also $O(\epsilon)$ in the TDL (see section 3.8). In Chapter 6, I show that computing the deviatoric estimator $\hat{\mathbf{T}}_{\text{new}}$ instead of $\hat{\mathbf{T}}$ fixes the noise issue.

5.4 Linearized Lattice Problem

In section 5.3, I presented calculation results for a classic problem in applied physics, one which has been used many times in the past quarter century to demonstrate the efficacy of different numerical methods for solving the equations of thermal radiative transfer. Here, I do the same for a much newer problem, proposed less than two years ago, which attempts to combine the good features of several classic problems into one. The new problem first appeared in a LLNL technical report by Brunner [94]. The author describes radiation flow through a lattice.

In Brunner's problem description, an intense boundary source of radiation shines on one face of a lattice composed of optically-thick iron and optically-thin foam with a central mass of hot iron which is also radiating. Radiation flows preferentially through the foam, and then leaves the problem through vacuum boundaries. The radiative heating process in Brunner's lattice occurs due to the physical processes which I described in Chapter 1, and Fig. 1.1 lists

some example processes by which we assume that the radiation in the lattice problem gets absorbed and re-emitted.

Brunner's lattice is a time-dependent problem. I solve a steady-state simplification which I call the Linearized Lattice (LL). I make the foam 1000 times optically-thinner than the iron. Fig. 5.19 depicts the geometry, material data, and boundary conditions for the LL problem. The boundary source is isotropic on the inflow hemisphere with magnitude,

$$\bar{\psi} = \begin{cases} 1 & x = 0, \\ 0 & \text{otherwise.} \end{cases} \quad (5.26)$$

Finally, there is an isotropic volume source in the central iron cube with magnitude,

$$q = \begin{cases} 1/2 & 3 \leq x \leq 4 = 0 \text{ and } 3 \leq y \leq 4, \\ 0 & \text{otherwise.} \end{cases} \quad (5.27)$$

I use a mesh composed of squares of equal sizes. The minimal number of uniform squares required to tessellate the LL geometry such that all material interfaces are on element boundaries is 7×7 . Fig. 5.20 shows the aforementioned 49-element mesh and how each element has material with single-valued properties, because each element contains either optically-thin foam or optically-thick iron, but not both. Fig. 5.21 shows the mesh after splitting each element into four squares of equal size to create a new mesh with 14×14 elements. Repeating this refinement operation four more times results in a 224×224 mesh, with 50,176 square elements, which is the mesh that I use in my calculations. I compute the angle integrated intensity. I set the convergence threshold η in the convergence criterion Eq. (4.1) to 10^{-3} .

Here, just like in Fig. 5.12 for LCP, I carried out three calculations. I computed the HSM solution using 8 billion simulation particles, I computed the UMC using 16 million simulation particles, and I computed the radiation diffusion approximation solution. Fig. 5.22 shows pseudocolor plots of the solutions.

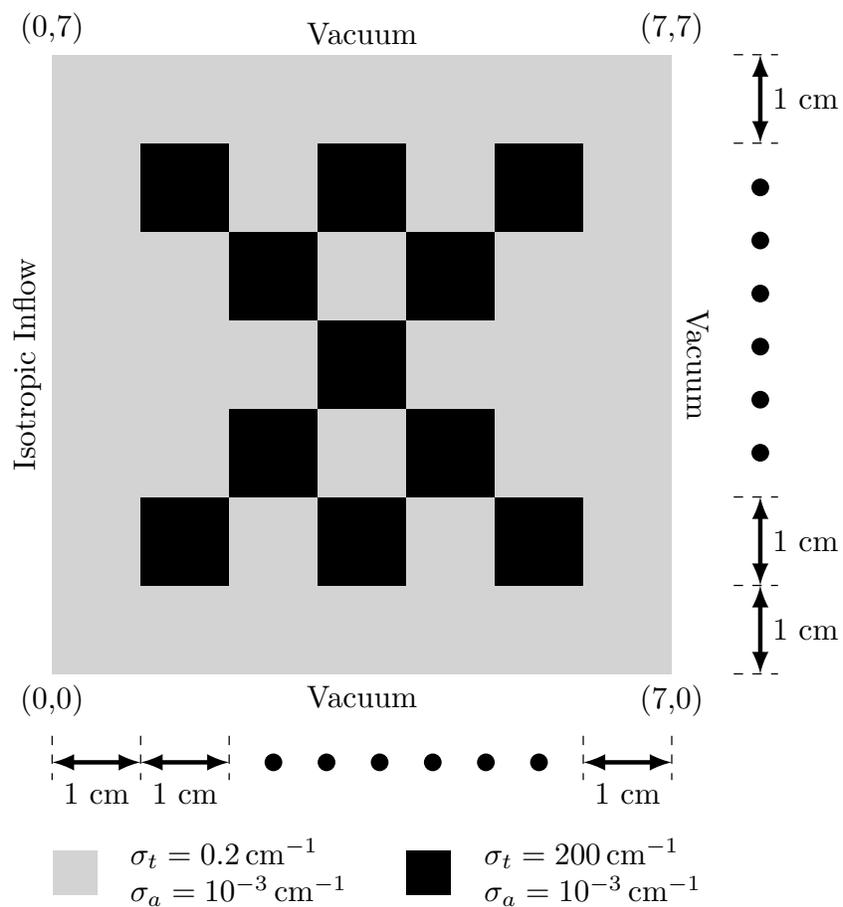


Figure 5.19: Linearized Lattice geometry, material data, and boundary conditions.

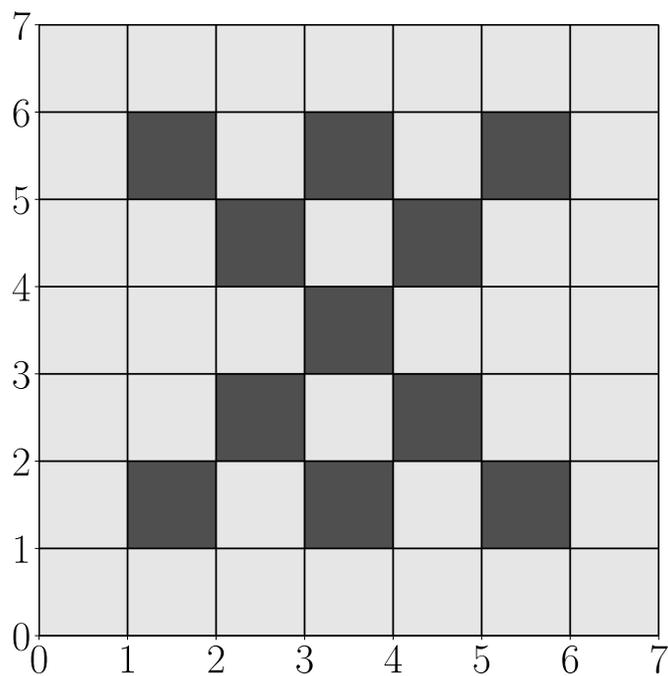


Figure 5.20: Uniform mesh of the Linearized Lattice geometry, $|\mathcal{T}| = 7 \times 7 = 49$.

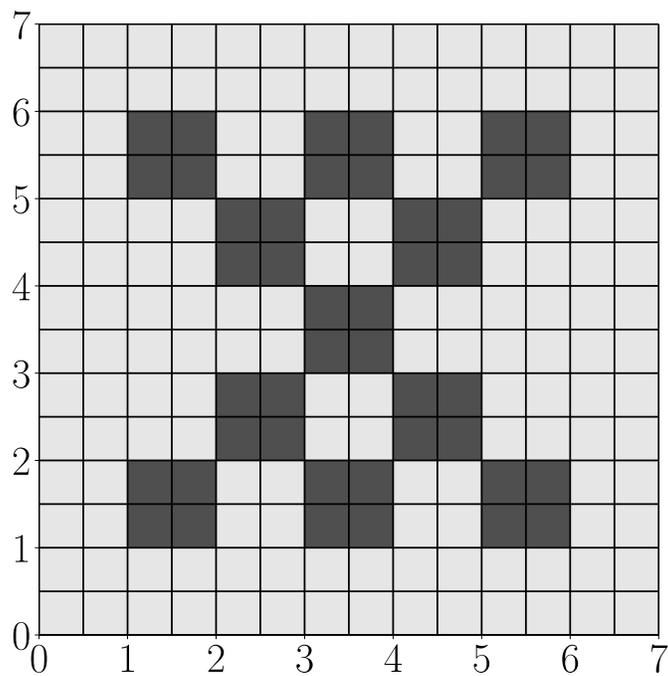


Figure 5.21: Refined mesh, $|\mathcal{T}| = 14 \times 14 = 196$.

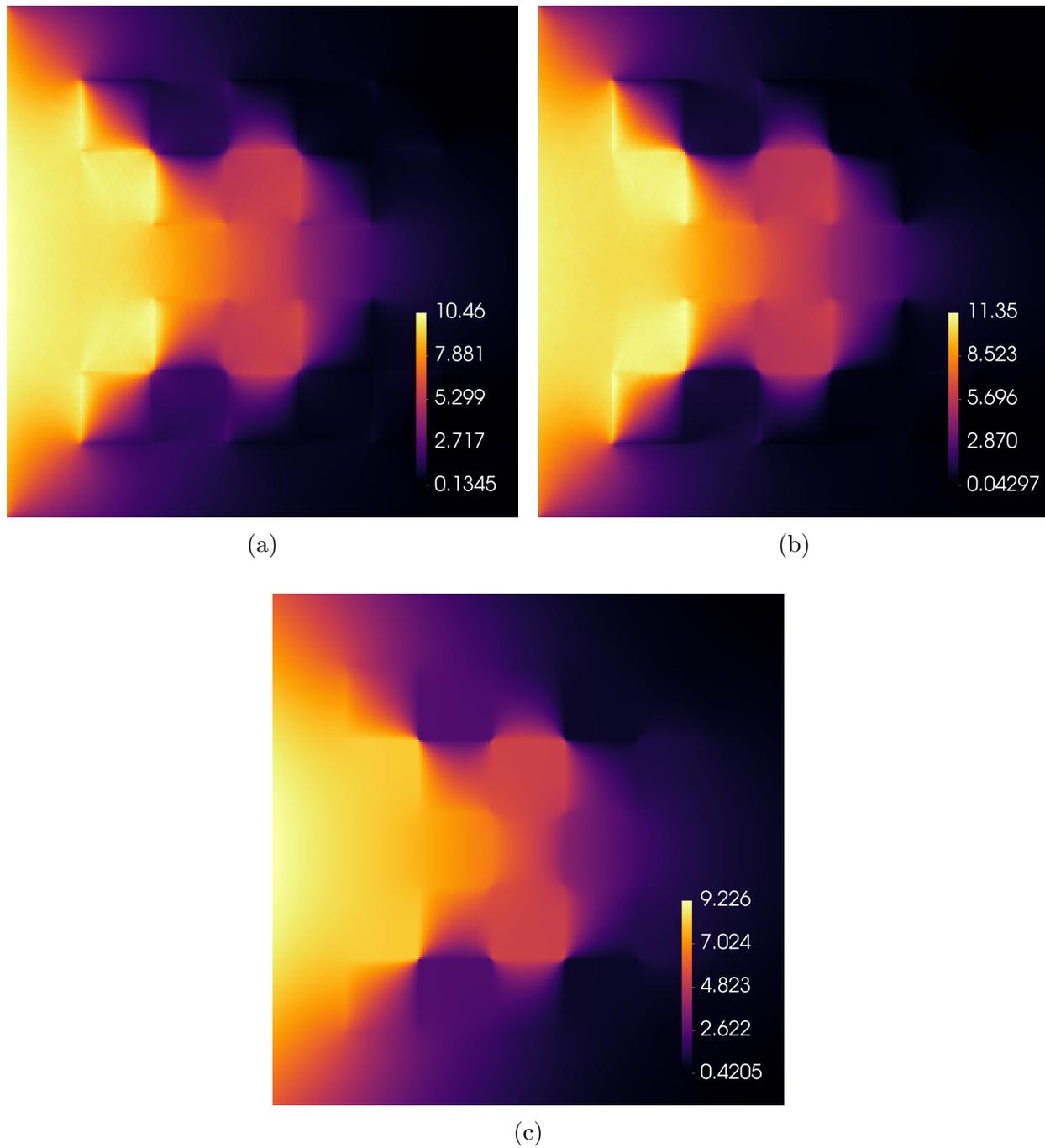


Figure 5.22: HSM (a), UMC (b), and diffusion (c) solutions for the linearized lattice.

Four comments regarding the solutions in Fig. 5.22 are:

1. My HSM and UMC solutions appear to agree within 8.5%, as judged by the ratio of the colorbar maximum values. Both show bright illumination on the edges of the iron squares that are facing the boundary and volume sources, and dark shadows on the edges facing away from the sources. However, the areas of highest intensity and lowest intensity are noticeably brighter and darker, respectively, in the UMC solution compared to the HSM solution.
2. The radiation diffusion approximation gives a solution with a substantial approximation error, in this case 23%, as judged by the ratio of the colorbar maximum values (assuming that the UMC solution is the correct solution). The radiation diffusion approximation tends to over-illuminate regions of low intensity (dark regions in the UMC solution) and under-illuminate regions of high intensity (bright regions in the UMC solution). This behavior gives the impression that the radiation is “diffusing” through the iron rather than “transporting” around the iron via the foam.
3. Running fewer than 8 billion particles in the HSM calculation, or fewer than 16 million particles in the UMC calculation, causes visible noise to appear in the solution. The factor of 500 more particles required for the HSM calculation is an undesirable consequence of the $\text{Var}[\hat{\phi}] = O(1/\epsilon)$ result that I derived in section 3.6, and for which I provided supporting empirical evidence in Fig. 5.8.
4. The similarity between the LL results in Fig. 5.22 and the LCP results in Fig. 5.12 suggests that the features of the HSM, UMC, and radiation diffusion approximation solutions are independent of the geometry, because these features are present in both a pipe and a lattice. Consequently, these features appear to be intrinsic to the methods themselves.

5.4.1 Particle-Equivalent Comparisons with Varied Optical Thicknesses

The purpose of this section is to use the LL problem to compare the quality of the HSM and UMC solutions using the same number of particles, just as we did in section 5.3.1 using the LCP problem. Figs. 5.23 to 5.26 show plots of the HSM and UMC solutions for $\sigma_t^{\text{thick}}/\sigma_t^{\text{thin}} = 2, 10, 100, 1000$ (see table 5.1). I ran HSM and UMC calculations for the 2, 10, 100, and 1000 optical thickness ratio LL problems using 1 million, 4 million, and 16 million particles. Thus, I ran the following 24 calculations,

$$\underbrace{\{\text{HSM, UMC}\}}_{2 \text{ methods}} \times \underbrace{\{1\text{e}6, 4\text{e}6, 16\text{e}6\}}_{3 \text{ particle counts}} \times \underbrace{\{2, 10, 100, 1000\}}_{4 \text{ problem definitions}},$$

where “1e6” is 1 million, “4e6” is 4 million, and “16e6” is 16 million. As in section 5.3.1, I have divided the pseudocolor plots in half, this time along the line $y = 3.5$.

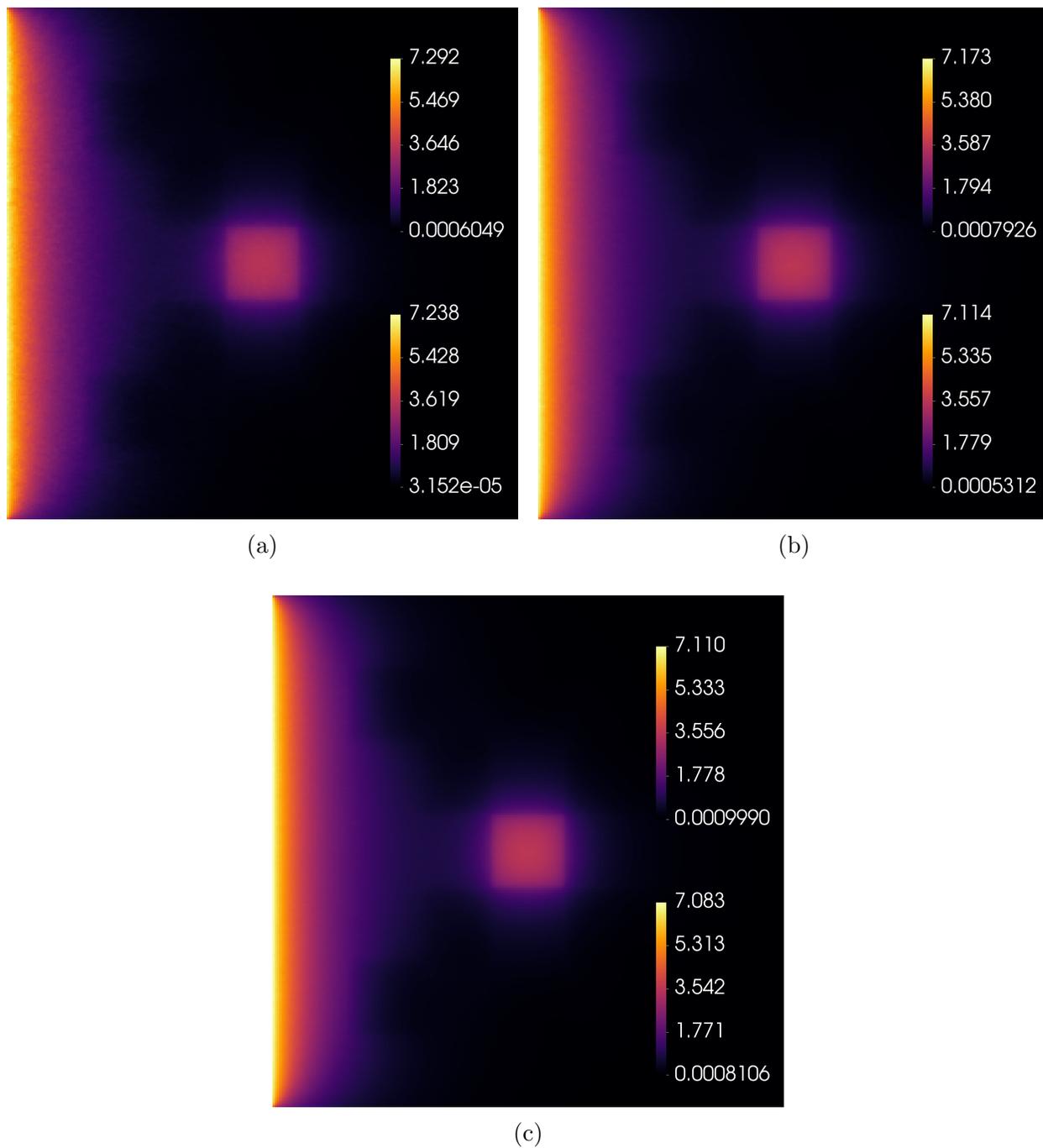


Figure 5.23: Solutions using 1e6 (a), 4e6 (b), and 16e6 (c) particles with HSM (top of a,b,c) and UMC (bottom of a,b,c) for the LL problem with $\sigma_t^{\text{thick}}/\sigma_t^{\text{thin}} = 2$.

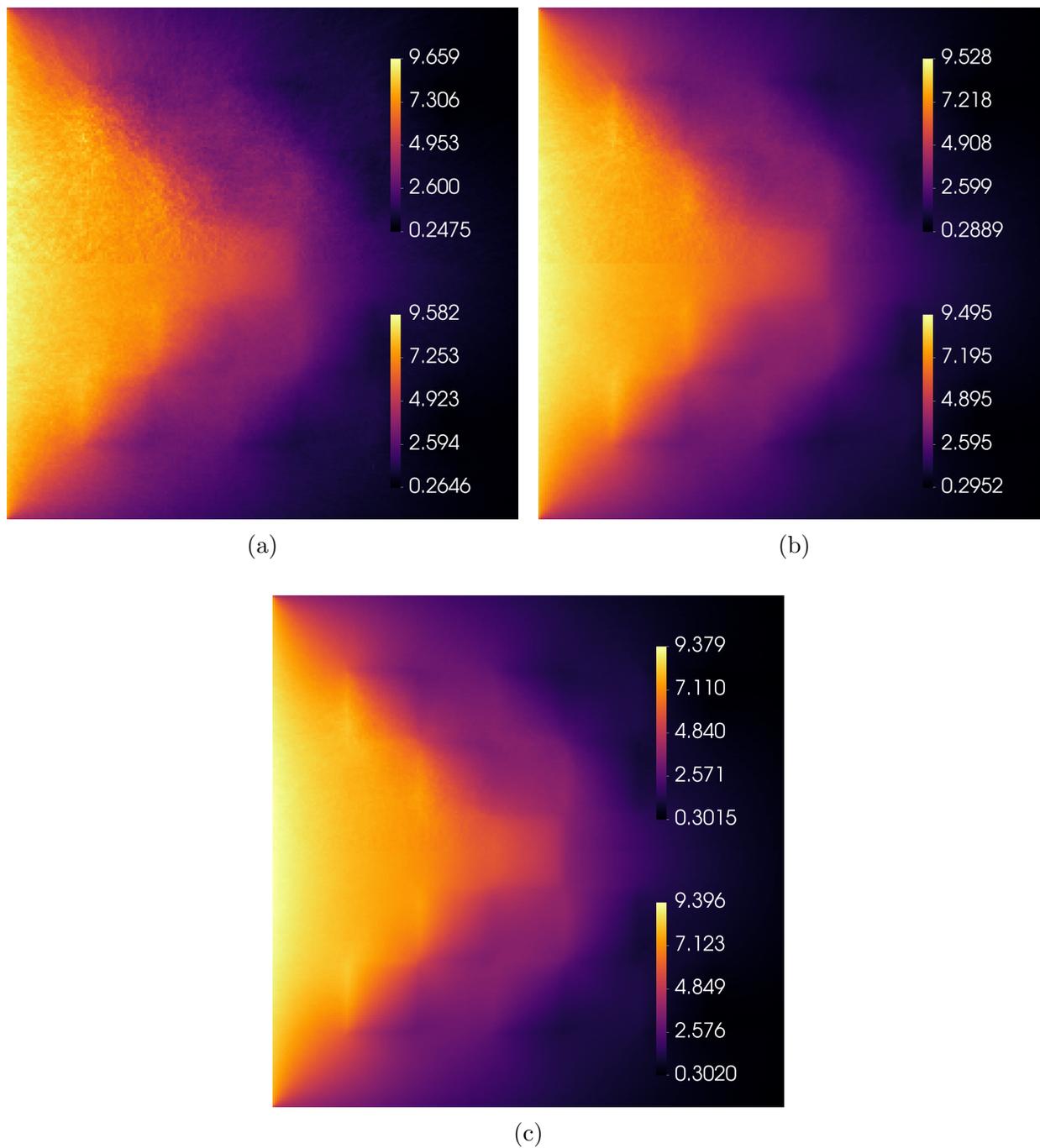


Figure 5.24: Solutions using 1e6 (a), 4e6 (b), and 16e6 (c) particles with HSM (top of a,b,c) and UMC (bottom of a,b,c) for the LL problem with $\sigma_t^{\text{thick}}/\sigma_t^{\text{thin}} = 10$.

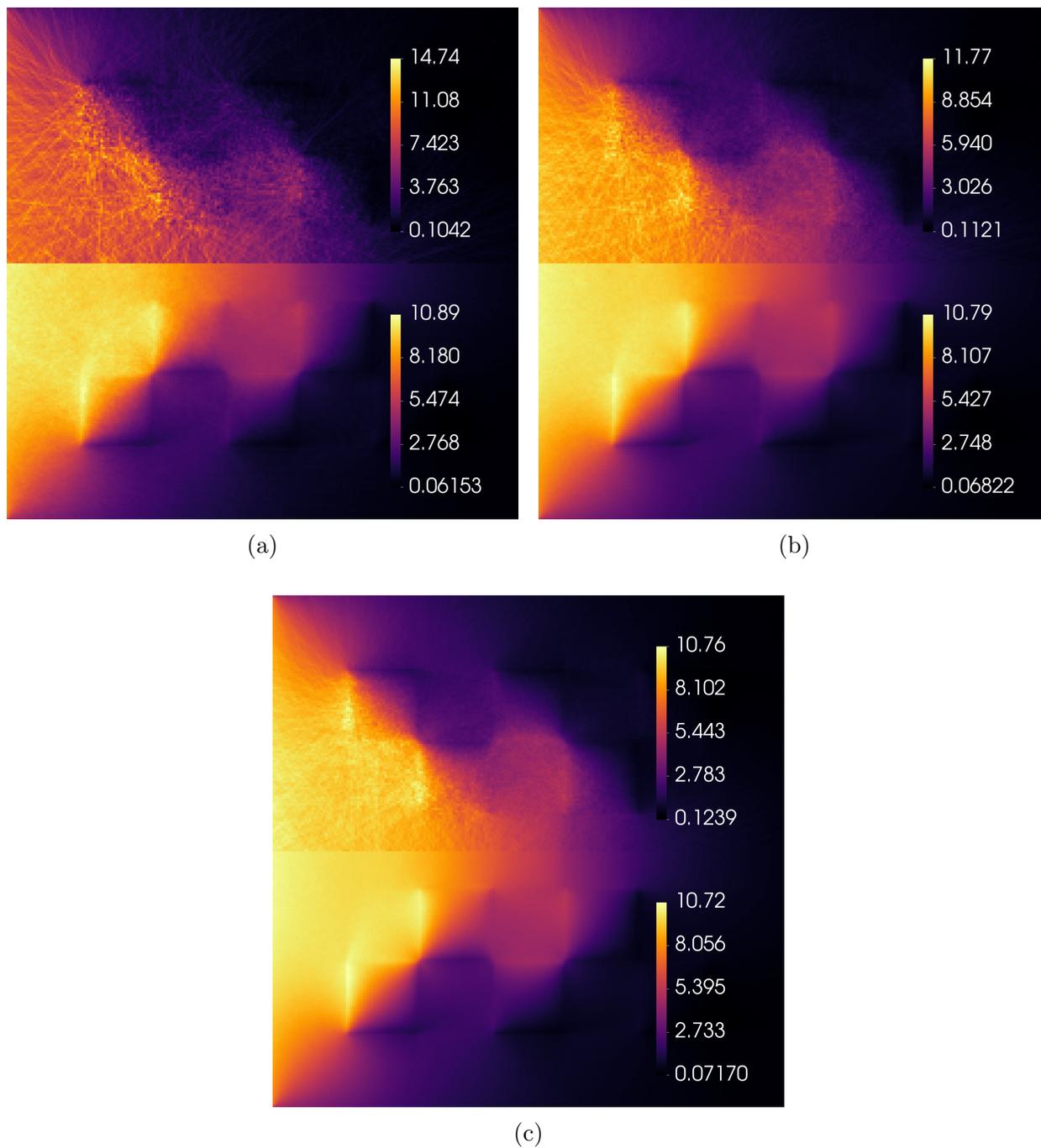


Figure 5.25: Solutions using $1e6$ (a), $4e6$ (b), and $16e6$ (c) particles with HSM (top of a,b,c) and UMC (bottom of a,b,c) for the LL problem with $\sigma_t^{\text{thick}}/\sigma_t^{\text{thin}} = 100$.

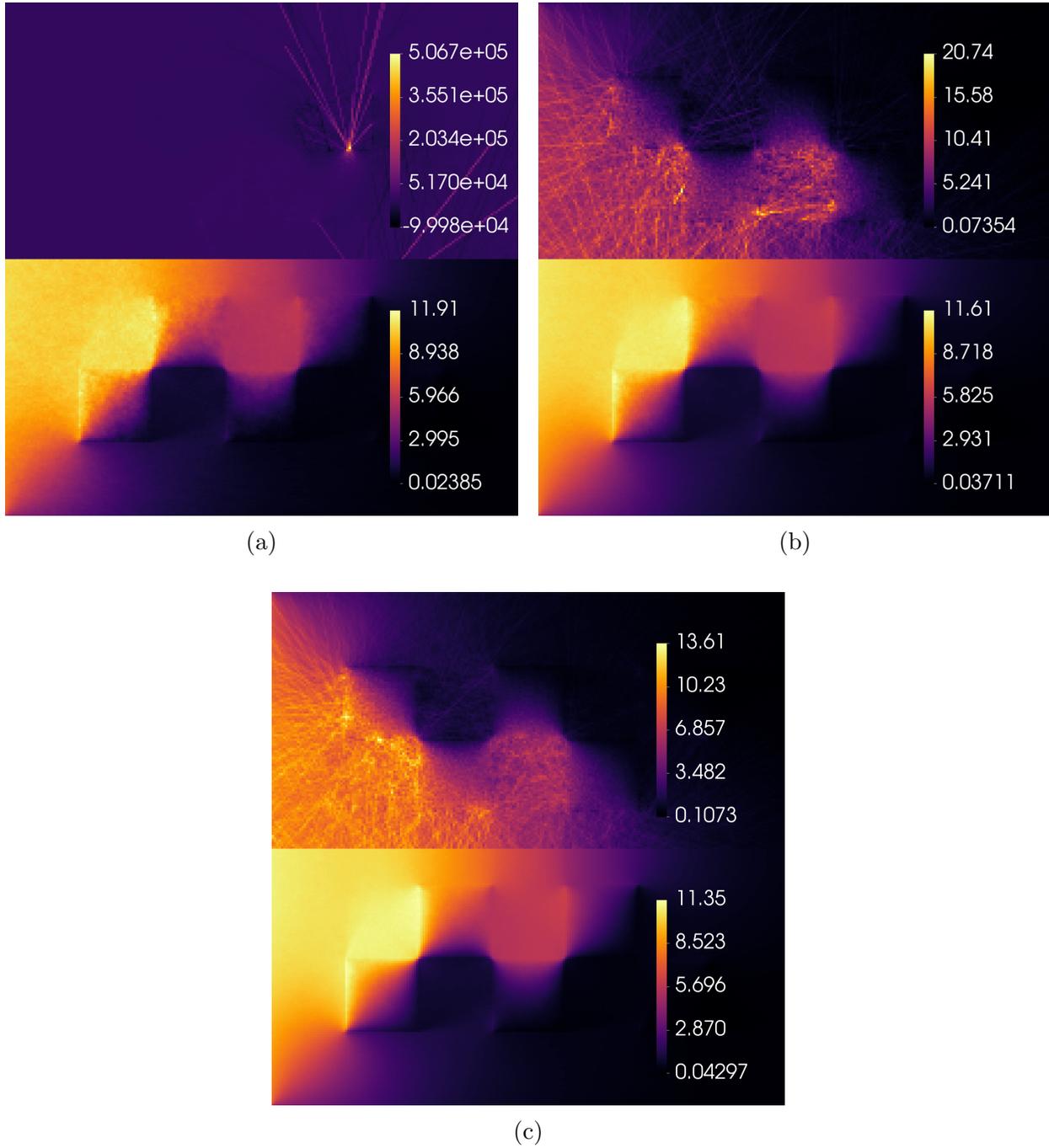


Figure 5.26: Solutions using 1e6 (a), 4e6 (b), and 16e6 (c) particles with HSM (top of a,b,c) and UMC (bottom of a,b,c) for the LL problem with $\sigma_t^{\text{thick}}/\sigma_t^{\text{thin}} = 1000$.

Four comments regarding the results in Figs. 5.23 to 5.26 are:

1. The $\sigma_t^{\text{thick}}/\sigma_t^{\text{thin}} = 2$ results in Fig. 5.23 show excellent agreement between HSM and UMC for all 3 particle counts. Both HSM and UMC have nearly identical amounts of visible noise at 1e6 particles that becomes less visible at 4e6 and is indiscernible at 16e6 particles. The radiation exhibits minimal propagation.
2. The $\sigma_t^{\text{thick}}/\sigma_t^{\text{thin}} = 10$ results in Fig. 5.24 show good agreement between HSM and UMC for all 3 particle counts. At 1e6 and 4e6 particles, the HSM and UMC solutions both exhibit visible noise, though the HSM noise is much more noticeable. The noise remains visible at 16e6 particles in the HSM solution, but not in the UMC solution. The radiation propagates further.
3. The $\sigma_t^{\text{thick}}/\sigma_t^{\text{thin}} = 100$ results in Fig. 5.25 show some agreement between HSM and UMC which increases with particle count. The HSM solution has significant noise at 1e6 particles. The noise is less severe at 4e6 and 16e6 particles, but remains visible. The UMC solution has visible noise at 1e6 particles, barely noticeable noise at 4e6 particles, and indiscernible at 16e6 particles. The radiation propagation in Fig. 5.25 appears similar to Fig. 5.24, but with intensities that show significantly higher contrast (brighter brights and darker darks).
4. The $\sigma_t^{\text{thick}}/\sigma_t^{\text{thin}} = 1000$ results in Fig. 5.26 show significant differences between HSM and UMC, though the disagreement decreases with particle count. The HSM solution at 1e6 is dark except for a small collection of neighboring elements as well as elements along streak lines and with a colorbar maximum value that is orders of magnitude too large, all of which is unphysical and indicates that the iteration diverged. The HSM solutions at 4e6 and 16e6 particles look reasonable, but very noisy. The UMC solution has noticeable noise at 1e6 particles, less at 4e6, and indiscernible noise at 16e6 particles. The radiation propagation in Fig. 5.26 appears similar to Fig. 5.25, but with intensities that show noticeably higher contrast (brighter brights and darker darks).

As we saw in section 5.3.1 Particle-Equivalent Comparisons with Varied Optical Thicknesses for the LCP problem, the HSM solution has much more noise than UMC for an equivalent number of simulation particles when the material is optically thick. This issue is caused by my choice of estimators for the SMM data: $\hat{\mathbf{T}}$ is noisy in optically-thicker materials because $\text{Var}[\hat{\mathbf{T}}]$ is $O(1/\epsilon)$, as I derived in section 3.6 and showed in Fig. 5.8.

5.4.2 Confirming Avoidance of Noise Amplification

The purpose of this section is to show that the results observed with the LCP problem in section 5.3.2 Confirming Avoidance of Noise Amplification also hold for the LL problem. Specifically, the mixed finite element solve of the SMM system does not amplify the Monte Carlo noise in the SMM data estimators $\hat{\mathbf{T}}$ and $\hat{\beta}$. The calculations in this section use HSM

to solve the LL problem with $\sigma_t^{\text{thick}}/\sigma_t^{\text{thin}} = 1000$. I use 500 million particles, which is a factor of 16 fewer particles than the 8 billion particles that I used to produce the visually-noiseless HSM solution in Fig. 5.22 (a). I plot intermediate solutions during the HSM calculation before convergence in order to show when the noise appears and how it changes during the HSM iteration.

Fig. 5.27 shows the logic in Algorithm 3 for computing the HSM solution alongside pseudocolor plots of the unscattered estimator $\hat{\phi}^{(0)}$, the estimator first iterate $\hat{\phi}^{(1)}$, and the estimator at convergence $\hat{\phi}^{(13)}$. Fig. 5.28 shows the unscattered estimator $\hat{\phi}^{(0)}$ and the estimator first iterate $\hat{\phi}^{(1)}$ again, but with the second moment system solution iterate $\varphi^{(1)}$ interposed between them.

Three observations about Figs. 5.27 and 5.28 are:

- i. The unscattered estimator $\hat{\phi}^{(0)}$, plotted in Fig. 5.27 (a) and Fig. 5.28 (a), illuminates the lattice so weakly that the iron squares look black, meaning that the intensity in the iron squares is zero or near-zero. Consequently, the work required to compute the illumination of the iron squares in the HSM solve of the LL problem is left to the iteration of the scattering source. Additionally, the unscattered estimator $\hat{\phi}^{(0)}$ does not show the noise issue.
- ii. The noise issue is apparent in both $\hat{\phi}^{(1)}$ and $\hat{\phi}^{(13)}$, which are plotted in Fig. 5.27 (b) and (c), respectively. However, the noise issue does not appear to worsen between cycle 1 and 13, because $\hat{\phi}^{(13)}$ appears unsubstantially noisier than $\hat{\phi}^{(1)}$.
- iii. The second moment solution iterate $\varphi^{(1)}$, plotted in Fig. 5.28 (b), does not show the noise issue. Thus, the noise problem results from the transport of the scattering source, not the moment system solve. Avoiding differentiation of the correction tensor by solving the first order moment system using a FEM to offload the derivative appears to be effective for the prevention of noise amplification.

I conclude that the noise issue is a result of the calculation and transport of the scattering source. One final observation is that the total weight of the simulation particles before iterating is about 28, whereas it is about 6,800 during iteration, plus or minus a few percent because the weight varies by a few percent over the iteration.

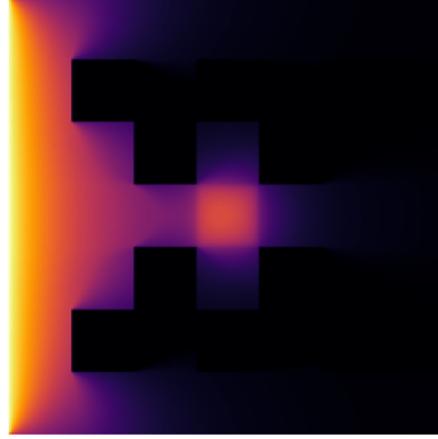
Recall from section 3.6 that the variance of my $\hat{\mathbf{T}}$ estimator of the SMM transport correction tensor is $O(1/\epsilon)$ in the TDL, as supported by the empirical result in Fig. 5.8. I derived $\hat{\mathbf{R}}$ in section 3.7, which is an alternative estimator for \mathbf{T} that is $O(\epsilon)$ in the TDL. I derived another alternative estimator for \mathbf{T} , called the deviatoric estimator, which is also $O(\epsilon)$ in the TDL (see section 3.8). In Chapter 6, I show that computing the deviatoric estimator $\hat{\mathbf{T}}_{\text{new}}$ instead of $\hat{\mathbf{T}}$ fixes the noise issue.

Algorithm Hybrid Second Moment

```

1: Input: user-provided boolean value HSM
2: if not HSM then
3:   scattering_events  $\leftarrow$  true
4:    $\hat{\phi} \leftarrow$  mc(q, scattering_events,  $\bar{\psi}$ )
5:   return  $\hat{\phi}$ 
6: end if
7: scattering_events  $\leftarrow$  false
8:  $\hat{\phi}^{(0)}, \hat{\mathbf{T}}^{(0)}, \hat{\beta}^{(0)} \leftarrow$  mc(q, scattering_events,  $\bar{\psi}$ )
9: i  $\leftarrow$  1
10: while not converged( $\hat{\phi}^{(i-1)}, \hat{\phi}^{(i)}$ ) do
11:    $\varphi^{(i)} \leftarrow$  sm(Q0, Q1,  $\hat{\mathbf{T}}^{(i-1)}, \hat{\beta}^{(i-1)}$ )
12:    $\hat{\phi}_{\text{temp}}, \hat{\mathbf{T}}_{\text{temp}}, \hat{\beta}_{\text{temp}} \leftarrow$  mc( $\varphi^{(i)}$ , scattering_events)
13:    $\hat{\phi}^{(i)} \leftarrow \hat{\phi}^{(0)} + \hat{\phi}_{\text{temp}}$ 
14:    $\hat{\mathbf{T}}^{(i)} \leftarrow \hat{\mathbf{T}}^{(0)} + \hat{\mathbf{T}}_{\text{temp}}$ 
15:    $\hat{\beta}^{(i)} \leftarrow \hat{\beta}^{(0)} + \hat{\beta}_{\text{temp}}$ 
16:   i  $\leftarrow$  i + 1
17: end while
18: return  $\hat{\phi}^{(i)}$ 

```



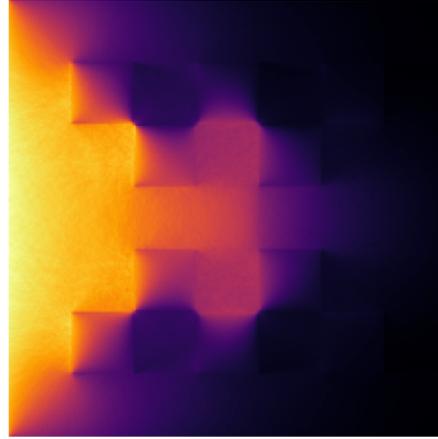
(a)

Algorithm Hybrid Second Moment

```

1: Input: user-provided boolean value HSM
2: if not HSM then
3:   scattering_events  $\leftarrow$  true
4:    $\hat{\phi} \leftarrow$  mc(q, scattering_events,  $\bar{\psi}$ )
5:   return  $\hat{\phi}$ 
6: end if
7: scattering_events  $\leftarrow$  false
8:  $\hat{\phi}^{(0)}, \hat{\mathbf{T}}^{(0)}, \hat{\beta}^{(0)} \leftarrow$  mc(q, scattering_events,  $\bar{\psi}$ )
9: i  $\leftarrow$  1
10: while not converged( $\hat{\phi}^{(i-1)}, \hat{\phi}^{(i)}$ ) do
11:    $\varphi^{(i)} \leftarrow$  sm(Q0, Q1,  $\hat{\mathbf{T}}^{(i-1)}, \hat{\beta}^{(i-1)}$ )
12:    $\hat{\phi}_{\text{temp}}, \hat{\mathbf{T}}_{\text{temp}}, \hat{\beta}_{\text{temp}} \leftarrow$  mc( $\varphi^{(i)}$ , scattering_events)
13:    $\hat{\phi}^{(i)} \leftarrow \hat{\phi}^{(0)} + \hat{\phi}_{\text{temp}}$ 
14:    $\hat{\mathbf{T}}^{(i)} \leftarrow \hat{\mathbf{T}}^{(0)} + \hat{\mathbf{T}}_{\text{temp}}$ 
15:    $\hat{\beta}^{(i)} \leftarrow \hat{\beta}^{(0)} + \hat{\beta}_{\text{temp}}$ 
16:   i  $\leftarrow$  i + 1
17: end while
18: return  $\hat{\phi}^{(i)}$ 

```



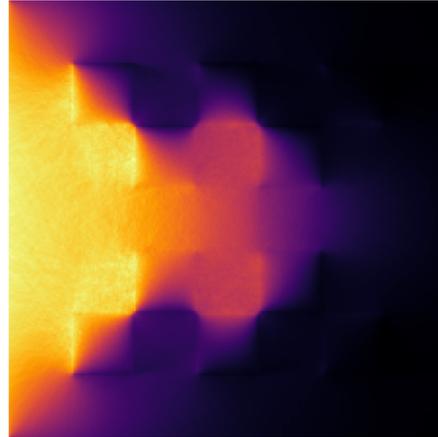
(b)

Algorithm Hybrid Second Moment

```

1: Input: user-provided boolean value HSM
2: if not HSM then
3:   scattering_events  $\leftarrow$  true
4:    $\hat{\phi} \leftarrow$  mc(q, scattering_events,  $\bar{\psi}$ )
5:   return  $\hat{\phi}$ 
6: end if
7: scattering_events  $\leftarrow$  false
8:  $\hat{\phi}^{(0)}, \hat{\mathbf{T}}^{(0)}, \hat{\beta}^{(0)} \leftarrow$  mc(q, scattering_events,  $\bar{\psi}$ )
9: i  $\leftarrow$  1
10: while not converged( $\hat{\phi}^{(i-1)}, \hat{\phi}^{(i)}$ ) do
11:    $\varphi^{(i)} \leftarrow$  sm(Q0, Q1,  $\hat{\mathbf{T}}^{(i-1)}, \hat{\beta}^{(i-1)}$ )
12:    $\hat{\phi}_{\text{temp}}, \hat{\mathbf{T}}_{\text{temp}}, \hat{\beta}_{\text{temp}} \leftarrow$  mc( $\varphi^{(i)}$ , scattering_events)
13:    $\hat{\phi}^{(i)} \leftarrow \hat{\phi}^{(0)} + \hat{\phi}_{\text{temp}}$ 
14:    $\hat{\mathbf{T}}^{(i)} \leftarrow \hat{\mathbf{T}}^{(0)} + \hat{\mathbf{T}}_{\text{temp}}$ 
15:    $\hat{\beta}^{(i)} \leftarrow \hat{\beta}^{(0)} + \hat{\beta}_{\text{temp}}$ 
16:   i  $\leftarrow$  i + 1
17: end while
18: return  $\hat{\phi}^{(i)}$ 

```



(c)

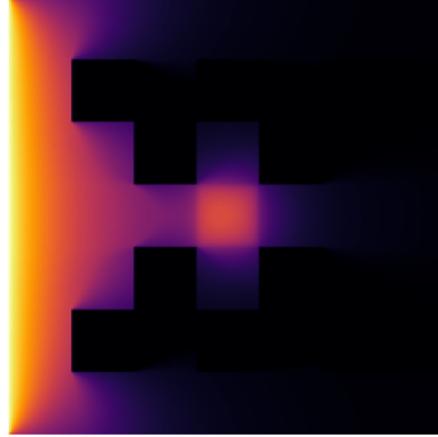
Figure 5.27: The HSM estimator $\hat{\phi}$ for the LL problem before iterating (a), after the first cycle (b), and at convergence (c), where 13 cycles were required to converge HSM using 500 million particles.

Algorithm Hybrid Second Moment

```

1: Input: user-provided boolean value HSM
2: if not HSM then
3:   scattering_events  $\leftarrow$  true
4:    $\hat{\phi} \leftarrow$  mc(q, scattering_events,  $\bar{\psi}$ )
5:   return  $\hat{\phi}$ 
6: end if
7: scattering_events  $\leftarrow$  false
8:  $\hat{\phi}^{(0)}, \hat{\mathbf{T}}^{(0)}, \hat{\beta}^{(0)} \leftarrow$  mc(q, scattering_events,  $\bar{\psi}$ )
9: i  $\leftarrow$  1
10: while not converged( $\hat{\phi}^{(i-1)}, \hat{\phi}^{(i)}$ ) do
11:    $\varphi^{(i)} \leftarrow$  sm(Q0, Q1,  $\hat{\mathbf{T}}^{(i-1)}, \hat{\beta}^{(i-1)}$ )
12:    $\hat{\phi}_{\text{temp}}, \hat{\mathbf{T}}_{\text{temp}}, \hat{\beta}_{\text{temp}} \leftarrow$  mc( $\varphi^{(i)}$ , scattering_events)
13:    $\hat{\phi}^{(i)} \leftarrow \hat{\phi}^{(0)} + \hat{\phi}_{\text{temp}}$ 
14:    $\hat{\mathbf{T}}^{(i)} \leftarrow \hat{\mathbf{T}}^{(0)} + \hat{\mathbf{T}}_{\text{temp}}$ 
15:    $\hat{\beta}^{(i)} \leftarrow \hat{\beta}^{(0)} + \hat{\beta}_{\text{temp}}$ 
16:   i  $\leftarrow$  i + 1
17: end while
18: return  $\hat{\phi}^{(i)}$ 

```



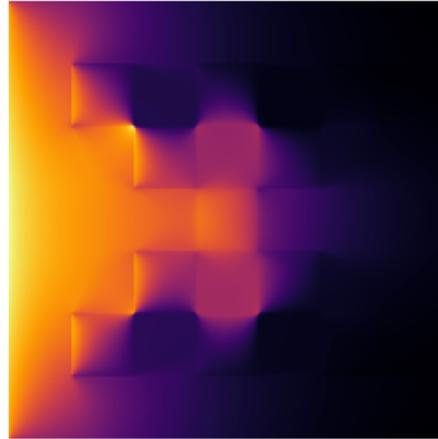
(a)

Algorithm Hybrid Second Moment

```

1: Input: user-provided boolean value HSM
2: if not HSM then
3:   scattering_events  $\leftarrow$  true
4:    $\hat{\phi} \leftarrow$  mc(q, scattering_events,  $\bar{\psi}$ )
5:   return  $\hat{\phi}$ 
6: end if
7: scattering_events  $\leftarrow$  false
8:  $\hat{\phi}^{(0)}, \hat{\mathbf{T}}^{(0)}, \hat{\beta}^{(0)} \leftarrow$  mc(q, scattering_events,  $\bar{\psi}$ )
9: i  $\leftarrow$  1
10: while not converged( $\hat{\phi}^{(i-1)}, \hat{\phi}^{(i)}$ ) do
11:    $\varphi^{(i)} \leftarrow$  sm(Q0, Q1,  $\hat{\mathbf{T}}^{(i-1)}, \hat{\beta}^{(i-1)}$ )
12:    $\hat{\phi}_{\text{temp}}, \hat{\mathbf{T}}_{\text{temp}}, \hat{\beta}_{\text{temp}} \leftarrow$  mc( $\varphi^{(i)}$ , scattering_events)
13:    $\hat{\phi}^{(i)} \leftarrow \hat{\phi}^{(0)} + \hat{\phi}_{\text{temp}}$ 
14:    $\hat{\mathbf{T}}^{(i)} \leftarrow \hat{\mathbf{T}}^{(0)} + \hat{\mathbf{T}}_{\text{temp}}$ 
15:    $\hat{\beta}^{(i)} \leftarrow \hat{\beta}^{(0)} + \hat{\beta}_{\text{temp}}$ 
16:   i  $\leftarrow$  i + 1
17: end while
18: return  $\hat{\phi}^{(i)}$ 

```



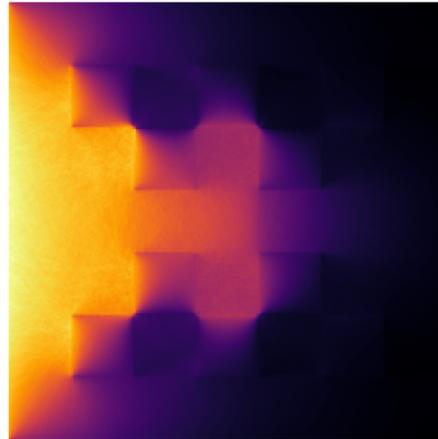
(b)

Algorithm Hybrid Second Moment

```

1: Input: user-provided boolean value HSM
2: if not HSM then
3:   scattering_events  $\leftarrow$  true
4:    $\hat{\phi} \leftarrow$  mc(q, scattering_events,  $\bar{\psi}$ )
5:   return  $\hat{\phi}$ 
6: end if
7: scattering_events  $\leftarrow$  false
8:  $\hat{\phi}^{(0)}, \hat{\mathbf{T}}^{(0)}, \hat{\beta}^{(0)} \leftarrow$  mc(q, scattering_events,  $\bar{\psi}$ )
9: i  $\leftarrow$  1
10: while not converged( $\hat{\phi}^{(i-1)}, \hat{\phi}^{(i)}$ ) do
11:    $\varphi^{(i)} \leftarrow$  sm(Q0, Q1,  $\hat{\mathbf{T}}^{(i-1)}, \hat{\beta}^{(i-1)}$ )
12:    $\hat{\phi}_{\text{temp}}, \hat{\mathbf{T}}_{\text{temp}}, \hat{\beta}_{\text{temp}} \leftarrow$  mc( $\varphi^{(i)}$ , scattering_events)
13:    $\hat{\phi}^{(i)} \leftarrow \hat{\phi}^{(0)} + \hat{\phi}_{\text{temp}}$ 
14:    $\hat{\mathbf{T}}^{(i)} \leftarrow \hat{\mathbf{T}}^{(0)} + \hat{\mathbf{T}}_{\text{temp}}$ 
15:    $\hat{\beta}^{(i)} \leftarrow \hat{\beta}^{(0)} + \hat{\beta}_{\text{temp}}$ 
16:   i  $\leftarrow$  i + 1
17: end while
18: return  $\hat{\phi}^{(i)}$ 

```



(c)

Figure 5.28: The HSM estimator $\hat{\phi}$ for the LL problem before iterating (a), φ in the first cycle (b), and $\hat{\phi}$ after cycle 1 (c). The pseudocolor plots in (a) and (c) are identical to the pseudocolor plots in Fig. 5.27 (a) and (b), respectively.

Chapter 6

Deviatoric Numerical Results

In this chapter, I use the deviatoric estimator to solve the same problems that I solved in Chapter 5. To differentiate my HSM method, which utilizes the SMM data estimators $\hat{\mathbf{T}}$ and $\hat{\beta}$, from the version that employs the deviatoric estimators $\hat{\mathbf{T}}_{\text{new}}$ and $\hat{\beta}_{\text{new}}$, I refer to the latter as deviatoric hybrid second moment (DHSM). Where the acronym HSM appeared in the plots and figure captions in Chapter 5, the acronym DHSM is used in the plots and figure captions in this chapter.

The purpose of this chapter is to demonstrate that DHSM fixes the noise issue observed with HSM in Chapter 5. My calculations provide numerical evidence that the DHSM estimators require far fewer particles to converge than the HSM estimators require. The numerical evidence aligns with the results presented in Chapter 3, where I demonstrated that, in the TDL, the variance of the HSM estimator is $O(1/\epsilon)$, while the variance of the DHSM estimator is $O(\epsilon)$. In the TDL, $\epsilon \ll 1/\epsilon$ because $\epsilon \in (0, 1]$ and $\epsilon \rightarrow 0$ is the asymptotic limit that characterizes the TDL (see section 1.3.6 for a brief description of the TDL).

In section 3.8.2 Implementation Details of Deviatoric Estimator, I explain that the angle integrated intensity ϕ does not have a well-defined derivative for all problems. However, this is not an issue in section 6.1 and section 6.2, where the smoothness of the source functions q and $\bar{\psi}$, combined with a homogeneous medium, ensures that $\nabla\phi$ is well-defined everywhere. The issue caused by the non-differentiability of ϕ appears in sections 6.3 and 6.4, where we solve problems for which $\nabla\phi$ is not well-defined. Specifically, ϕ is not differentiable on the interface between the optically-thick and optically-thin materials.

The non-differentiability of ϕ invalidates my derivation in section 3.8.1 Derivation of Deviatoric Estimator, where I substituted $\psi = \varphi/(4\pi) + \tilde{\psi}$ into the linear transport equation. Thus, in sections 6.3 and 6.4, the DHSM method computes a solution to a linear transport equation for which $\nabla\phi$ is well-defined, which differs from the actual equation that we want to solve. The effect of this error is most easily seen along the material interface, where the DHSM solution is incorrect. In section 3.8.2, I suggest a way to fix the issue. My suggestion involves minimizing the Dirichlet energy to make the strong derivative of φ well-defined, though I have yet to implement my suggestion.

6.1 Method of Manufactured Solutions Problem

I verify that my DHSM method solves the linear transport equation correctly by solving the same MMS problem that I solved in section 5.1 using the HSM method. The MMS solution ψ_{MMS} is Eq. (5.1). The MMS problem has nonzero SMM data $\mathbf{T} \neq 0$ and $\beta \neq 0$. It also has nonzero inflow $\bar{\psi} \neq 0$, and so $J_{\text{in}} \neq 0$. Thus, the MMS problem verifies my implementation of the coupling of the deterministic and Monte Carlo components of the hybrid method, as well as my implementation of the boundary conditions.

I compare my numerical DHSM solution with $\bar{\phi}_{\text{MMS}}(x_1, x_2, y_1, y_2)$, which is the element-averaged angle integrated MMS solution defined by Eq. (5.4). I fix the number of elements and scale up the number of particles in successive calculations by a factor of four each time. I fix the number of particles and increase the mesh resolution by scaling down the element width by a factor of two each time. I compute the error of my numerical DHSM solution with respect to the MMS solution using Eqs. (5.14) and (5.15) for the latter and the former cases, respectively. I hypothesize that the error of the DHSM solution is $O(h) + O(N^{-1/2})$, which is the error that I hypothesized for HSM, based on arguments that I described in section 4.1. As before, h is the element width, and N is the number of MC particles.

Fig. 6.1 shows the results of these studies. The slight degradation in convergence at the 4^{12} and 4^{13} particle counts, appearing as liftoff above the dashed curve at the bottom-right of Fig. 6.1a, is expected behavior because I distribute more and more MC particles in a mesh with a fixed number of elements under sample size refinement. I confirmed that this was the case by running on a coarser mesh and observing earlier liftoff, as well as running on a finer mesh and observing delayed liftoff.

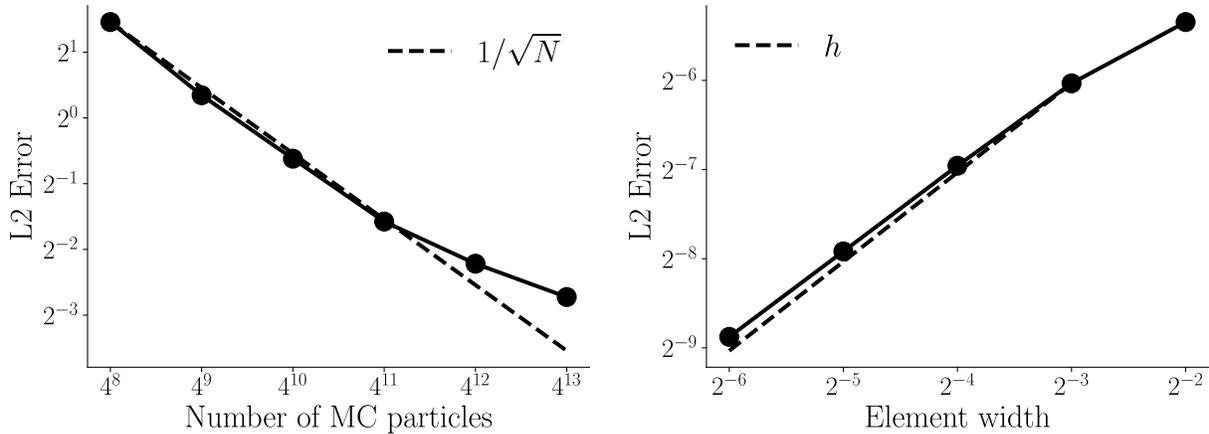
The very slight degradation in convergence below the hypothesized linear rate in h , appearing as liftoff above the dashed curve at the bottom-left of Fig. 6.1b, is also expected behavior, because I distribute a fixed number of MC particles across more and more elements under mesh refinement. I confirmed that this was the case by running fewer MC particles and observing earlier liftoff, as well as running more MC particles and observing delayed liftoff.

Interestingly, the HSM convergence under sample refinement, plotted in Fig. 5.4a, is nearly perfect, whereas the DHSM convergence, plotted in Fig. 6.1b, is degraded. I believe this is due to the constant on the $O(h)$ term in the error, which I suspect is larger in DHSM compared to HSM. I attribute this larger constant to the evaluation of φ on $\partial\mathcal{D}$, which is required in DHSM but not in HSM¹. Evaluating φ on $\partial\mathcal{D}$ introduces another place where the spatial discretization error caused by the mesh contributes to the $O(h)$ error term.

Another interesting difference between the HSM and DHSM convergence appears under mesh refinement. The HSM convergence, plotted in Fig. 5.4b, is nearly perfect until the finest mesh resolution, whereas the DHSM convergence is very slightly degraded, beginning with the mesh for which the element width is four times greater than the finest mesh. I

¹We need $\varphi(\mathbf{x})$ for $\mathbf{x} \in \partial\mathcal{D}$ in the Monte Carlo component of the DHSM method to integrate the variable boundary source, whereas the HSM method does not have a variable boundary source.

believe this reflects the influence of the variable boundary source, which requires the DHSM particle memory to be shared between the variable volume source and the variable boundary source during the DHSM iteration. In practice, I split the memory evenly between the two sources, which means that I use twice as many particles to integrate the variable volume source in my HSM calculations than in my DHSM calculations.



(a) Error as number of MC particles increases.

(b) Error as element width decreases.

Figure 6.1: Error of DHSM iterate $\hat{\phi}_{\text{new}}^{(i)}$ upon convergence.

6.2 Thick Diffusion Limit Problem

When I presented HSM results for the TDL in section 5.2, I mentioned that the thick diffusion limit is a challenging regime for transport methods. In Chapter 1, I explained why deterministic transport methods such as SI suffer from arbitrarily slow convergence in the TDL, and how Monte Carlo transport methods suffer from arbitrarily long runtimes due to extremely long MC particle histories in the TDL. The TDL also proved difficult for my HSM method, which we observed to have an estimator variance of $O(1/\epsilon)$ in the TDL. A quantity which scales as $O(1/\epsilon)$ is arbitrarily large because $\epsilon \rightarrow 0$ in the TDL. An unbounded variance is not a desirable quality for any numerical method.

The superiority of the deviatoric estimator comes from its $O(\epsilon)$ variance in the TDL. In section 3.8.3, we derived the $O(\epsilon)$ variance result. In this section, I give supporting numerical evidence which convincingly demonstrates the superiority of the deviatoric estimator for single-material transport problems in the TDL.

First, I calculate the DHSM solution for the four different optical-thickness parameter values $\epsilon = 10^{-1}, 10^{-2}, 10^{-3}$, and 10^{-4} . Fig. 6.2 shows that the DHSM solution appears to converge to the diffusion limit solution as $\epsilon \rightarrow 0$, as expected, because the lineouts appear to approach an equilibrium solution in the sense that successive lineouts differ less for smaller ϵ .

The DHSM solution in Fig. 6.2 is nearly perfect. The only visible imperfection is at $\epsilon = 10^{-1}$, where $\hat{\phi}_{\text{new}}(x_1)$ for $0.4 < x_1 < 0.5$ is just barely larger than $\hat{\phi}_{\text{new}}(x_2)$ for $0.5 < x_2 < 0.6$. These function values should be equal, as they are for smaller ϵ , and I observe visual equality when I increase the number of particles. The HSM solution in Fig. 5.6 is less perfect. It is visibly apparent that $|\hat{\phi}(x_1) - \hat{\phi}(x_2)| < |\hat{\phi}_{\text{new}}(x_1) - \hat{\phi}_{\text{new}}(x_2)|$ for $\epsilon = 10^{-1}$ as well as smaller values of ϵ , though the differences in the $\hat{\phi}$ values also go to zero as I increase the number of particles.

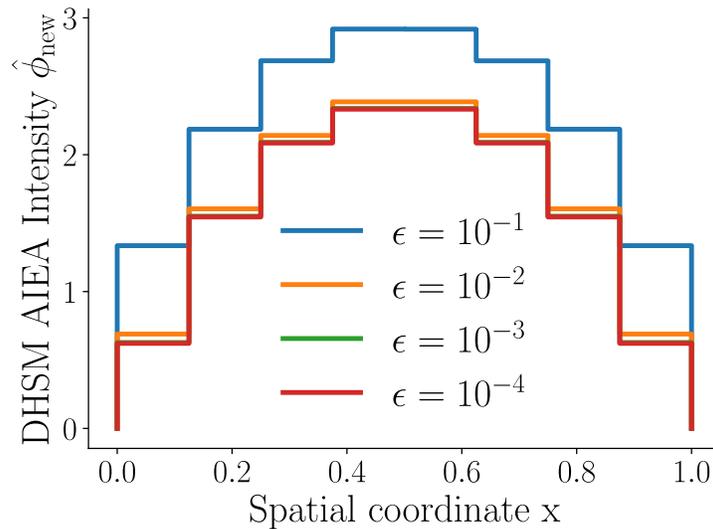


Figure 6.2: Lineouts at $y = 0.5$ of DHSM angle integrated element averaged (AIEA) intensity for calculations using four different values of the TDL scaling parameter ϵ .

The runtime of DHSM is $O(1)$ in the TDL, as can be seen in Fig. 6.3. The DHSM method runtime exceeds the HSM method runtime when $\epsilon = 1$, as can be seen by observing that the height of the rightmost cross in Fig. 6.3 exceeds the height of the rightmost cross in Fig. 5.7. This reflects the fact that DHSM requires more work than HSM, as it has an additional source—the variable boundary source—that must be sampled every cycle of the iteration. The DHSM method also needs derivatives and boundary values of φ , which HSM does not. However, the additional expense of DHSM is inconsequential in the TDL, where we see that the DHSM and HSM runtimes are equally small, as judged by the commensurate heights of the leftmost crosses in Figs. 5.7 and 6.3.

The metric for which DHSM significantly outperforms HSM is the variance. I estimate the variance of $\hat{\phi}_{\text{new}}$ in a single element by computing the squared difference of 600 realizations of $\hat{\phi}_{\text{new}}$ with the average of the 600 realizations, as in Eq. (5.22) (replace $\hat{\phi}$ with $\hat{\phi}_{\text{new}}$ in this equation). I repeat this procedure for five ϵ values: 10^{-1} , 10^{-2} , 10^{-3} , 10^{-4} , and 10^{-5} . Fig. 6.4 shows empirical agreement of $\text{Var}[\hat{\phi}_{\text{new}}]$ with the $O(\epsilon)$ result derived in section 3.8.3. Most importantly, $\text{Var}[\hat{\phi}_{\text{new}}] \ll \text{Var}[\hat{\phi}]$, as can be seen by comparing Figs. 5.8 and 6.4. Finally,

substituting the $O(1)$ runtime and $O(\epsilon)$ variance into Eq. (5.21) gives a FOM for DHSM of $O(1/\epsilon)$, which is significantly better than the $O(\epsilon)$ HSM FOM.

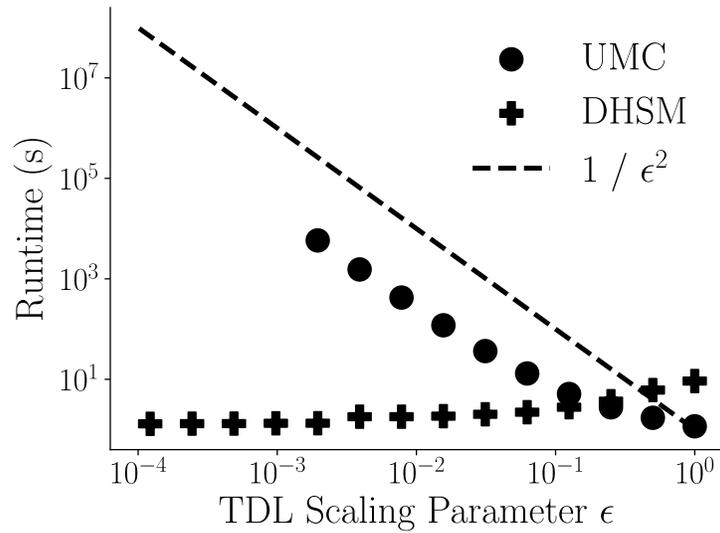


Figure 6.3: Runtime of unaccelerated Monte Carlo (UMC) and deviatoric hybrid second moment (DHSM) methods for calculations with different values of the TDL scaling parameter ϵ . The ϵ values are $2^0, 2^{-1}, 2^{-2}, \dots, 2^{-13}$.

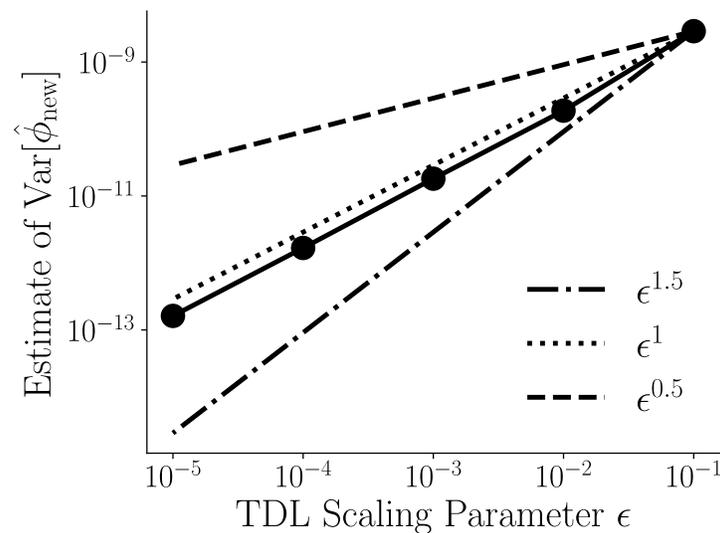


Figure 6.4: Estimate of the variance of the DHSM angle integrated element averaged intensity.

6.3 Linearized Crooked Pipe Problem

In section 5.3, I introduced the Crooked Pipe (CP) problem, which is commonly used to compare numerical methods for TRT. The CP problem definition comes from Graziani and LeBlanc [93]. It is an unsteady TRT problem. The Linearized Crooked Pipe (LCP) is a steady-state simplification of the CP problem that is equivalent to calculating one very large timestep of a backward Euler time integrator. The definition for the LCP problem that I solve comes from [39]. The LCP has a uniform volume source of $q = 10^{-7}$ everywhere in space, and an inflow source on the left boundary that is described by Eq. (5.24). Fig. 5.9 shows the LCP geometry, material data, and boundary conditions. Figs. 5.10 and 5.11 illustrate my meshing strategy. In this approach, I use the minimum number of equally-sized squares to tessellate the LCP geometry, ensuring that each square contains either optically-thick or optically-thin material, but not both. The result is a mesh with $14 \times 8 = 112$ elements, which I refine four times to produce the 224×128 mesh containing 28,672 equally-sized square elements.

In section 5.3, I ran HSM using 8 billion simulation particles, and plotted it alongside an unaccelerated Monte Carlo (UMC) solution calculated using 16 million simulation particles, and a radiation diffusion solution. Fig. 5.12 shows those results. Here, I replace the HSM solution plot with a plot of the DHSM solution using only 64 million particles. Fig. 6.5 demonstrates that 64 million particles were sufficient to eliminate any visibly-discernible noise from the DHSM solution. This represents over 100 times fewer particles than were required to produce the visibly-noiseless HSM solution shown in Fig. 5.12.

One significant difference between the HSM and DHSM calculations, apart from the substantial reduction in the number of simulation particles required by the DHSM method, is that the HSM solution closely resembles the UMC solution, whereas the DHSM solution appears distinctly different. That is, Fig. 5.12 (a) and (b) are very similar, whereas Fig. 6.5 (a) and (b) are not. This is a bad result for the DHSM method, because I consider the UMC solution in Figs. 5.12 and 6.5 (b) to be the reference solution². The disagreement of DHSM and UMC is most pronounced at the interface between the materials, where the DHSM solution is insufficiently bright or insufficiently dark relative to the UMC solution on the front and back walls, respectively. The DHSM solution also over-illuminates the right half of the pipe, and under-illuminates the left half. I believe the DHSM solution is incorrect, but not irreparable.

The issue is that the LCP solution has an undefined derivative at the interface between the two materials. In section 3.8.2, I described how this invalidates the substitution of $\psi = \varphi/(4\pi) + \tilde{\psi}$ into the transport Eq. (3.39a), and I suggested how to fix the issue. My suggestion involves replacing φ with some arbitrary function $\dot{\varphi}$ for which $\nabla\dot{\varphi}$ is well-defined. I also describe how solving a minimization problem could provide differentiability while preserving the crucial property that $\varphi - \dot{\varphi}$ must be $O(1/\sigma_t)$, as illustrated in Fig. 3.8. I have

²Not only does UMC involve fewer approximations than the hybrid methods, but my UMC solution closely resembles the S_N solution in Fig 9 (b) in Olivier *et al.* [39].

not yet implemented my suggestion.

6.3.1 Particle-Equivalent Comparisons with Varied Optical Thicknesses

Despite an improvement of two orders of magnitude in the number of particles required for the DHSM solution to be visually noiseless, 64 million is still four times more particles than the 16 million particles required by UMC. In this section, I run the same problems from section 5.3.1 using DHSM instead of HSM. I vary the opacities in the LCP so that the ratio $\sigma_t^{\text{thick}}/\sigma_t^{\text{thin}}$ is equal to 2, 10, 100, and 1000. The highest ratio corresponds to the opacity values in the LCP problem definition from Olivier *et al.* [39]. The other three values are choices that I made to investigate the behavior of my hybrid method. Table 5.1 shows the complete material data for the four optical-thickness ratios.

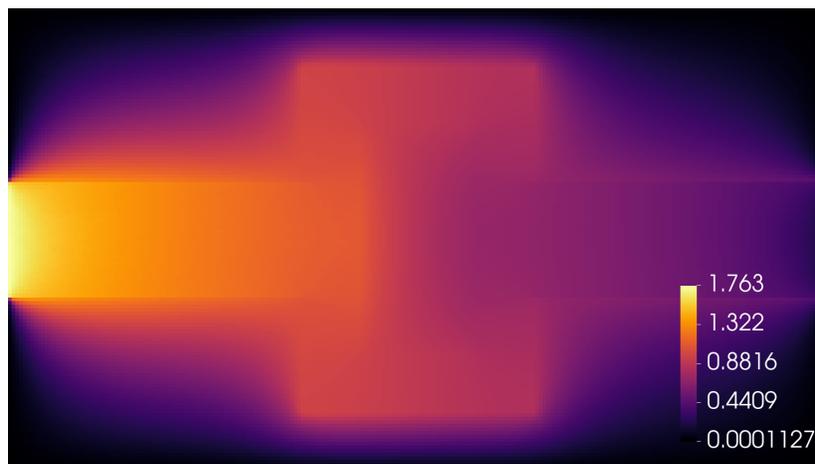
Figs. 6.6 to 6.9 show pseudocolor plots of the results of 24 calculations. Each of the four figures has three subfigures with results for calculations using (a) one million, (b) four million, and (c) sixteen million simulation particles. Each subfigure contains two plots. The top plot is DHSM and the bottom plot is UMC. I dedicate one paragraph of comments on each of the four figures.

1) Comparing Fig. 6.6 to Fig. 5.13, specifically the top half of the (a) subfigures, DHSM is noticeably more noisy than HSM. This indicates that HSM may be more efficient for optically-thin calculations. The DHSM solution also shows less agreement with the UMC solution than the HSM solution, which is likely related to the aforementioned non-differentiability issue on the material interface. Finally, the DHSM solution violates the non-negativity property of the transport Eq. (1.13a).

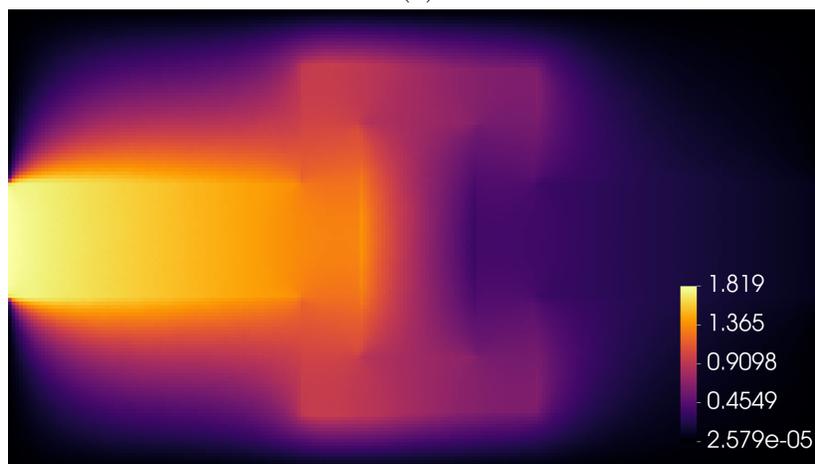
2) Comparing Fig. 6.7 to Fig. 5.14, specifically the top half of the (a) subfigures, DHSM and HSM have approximately commensurate amounts of visible noise. Interestingly, the DHSM noise appears to be lower frequency than the HSM noise, in the sense that the DHSM noise manifests as unphysical streaking in the solution, whereas the HSM noise presents as unphysical spotting in the solution.

3) Comparing Fig. 6.8 to Fig. 5.15, specifically the top half of the (a) subfigures, DHSM is less noisy than HSM. Increasing the number of HSM particles by a factor of 16 dramatically reduces the noise, though the resulting amount of noise is visually similar than the amount of noise in the DHSM calculation with 16 times fewer simulation particles.

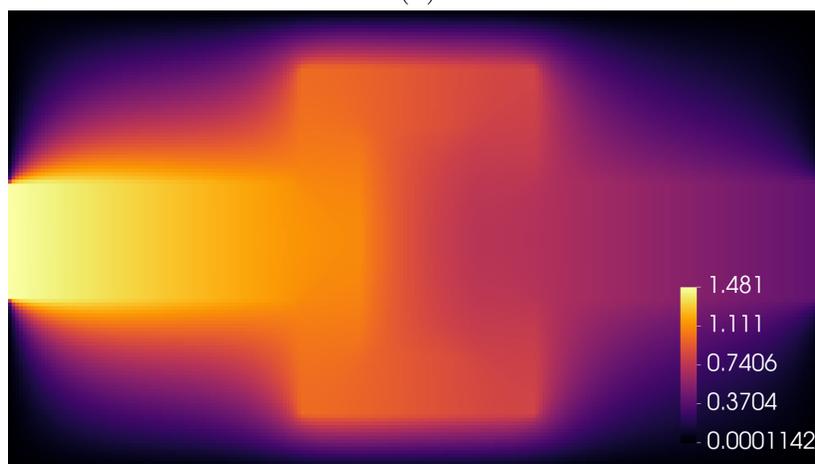
4) A comparison of Fig. 6.9 to Fig. 5.16 demonstrates the most dramatic difference between the DHSM and HSM method, which occurs due to the regions of very optically-thick material in the problem. Not only do the HSM and DHSM solutions have visible noise, but the UMC solution does as well. Inspecting the (a) subfigures reveals that DHSM converges to a reasonable solution at only one million simulation particles, whereas the HSM iteration diverges, resulting in an unphysical solution characterized by a single bright spot. The brightness of the unphysical bright spot is about twenty orders of magnitude too bright.



(a)



(b)



(c)

Figure 6.5: DHSM (a), UMC (b), and diffusion (c) solutions for the linearized crooked pipe.



(a)

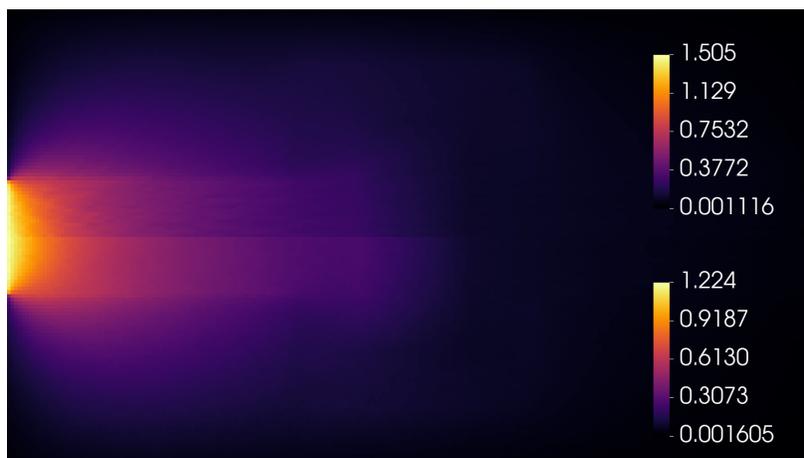


(b)

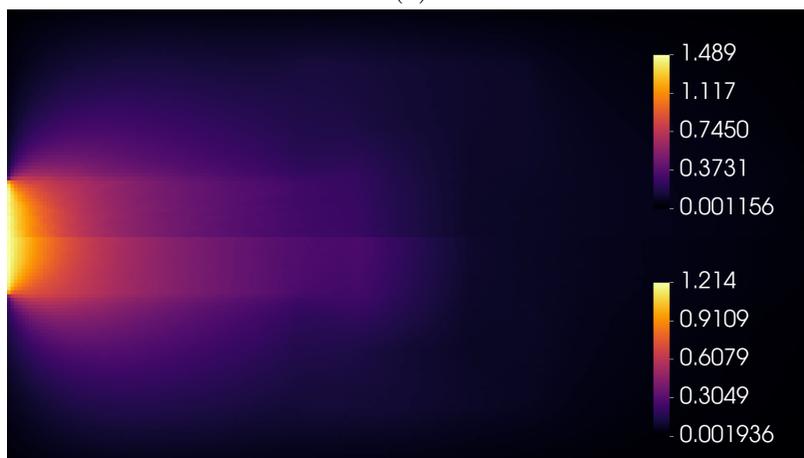


(c)

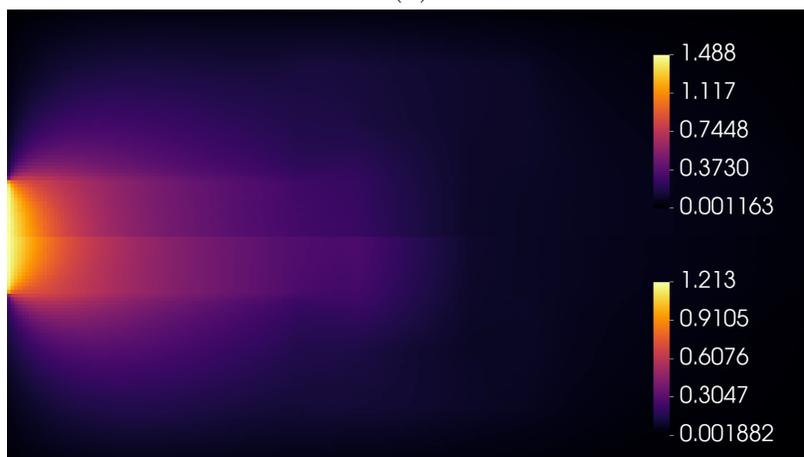
Figure 6.6: Solutions using 1e6 (a), 4e6 (b), and 16e6 (c) particles with DHSM (top of a,b,c) and UMC (bottom of a,b,c) for the LCP problem with $\sigma_t^{\text{thick}}/\sigma_t^{\text{thin}} = 2$.



(a)

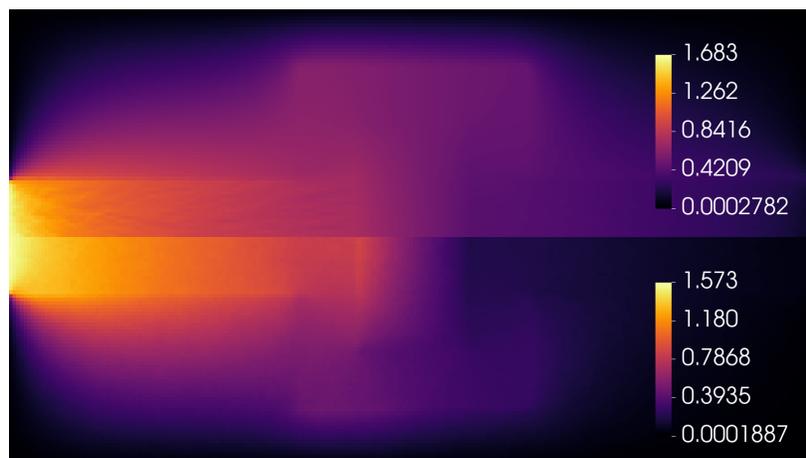


(b)

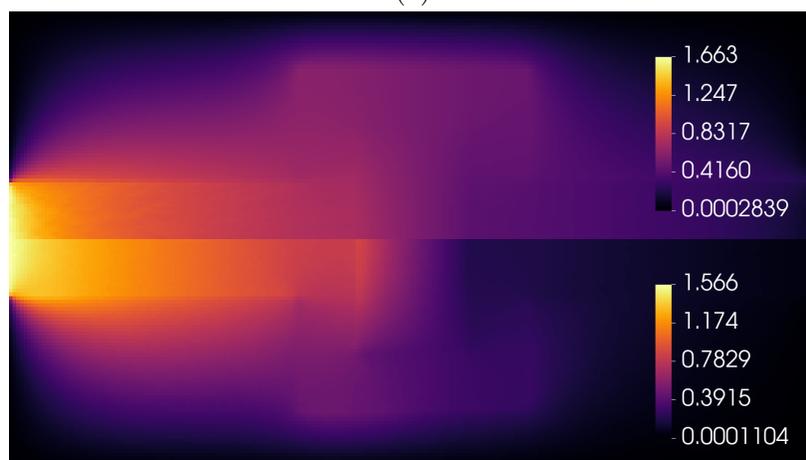


(c)

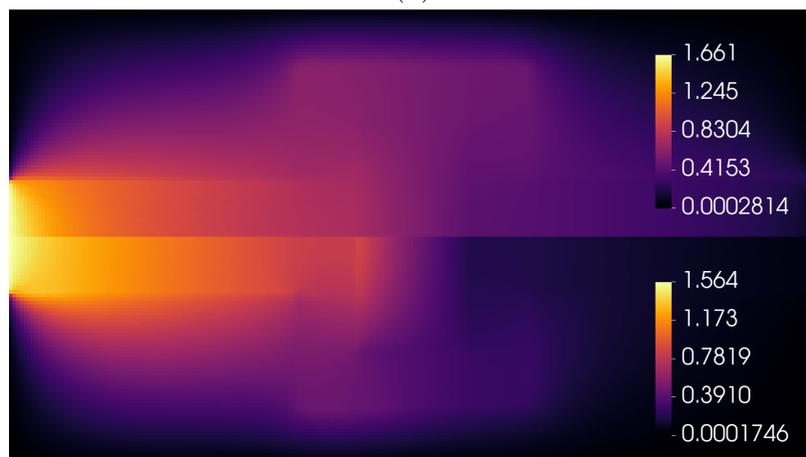
Figure 6.7: Solutions using 1e6 (a), 4e6 (b), and 16e6 (c) particles with DHSM (top of a,b,c) and UMC (bottom of a,b,c) for the LCP problem with $\sigma_t^{\text{thick}}/\sigma_t^{\text{thin}} = 10$.



(a)

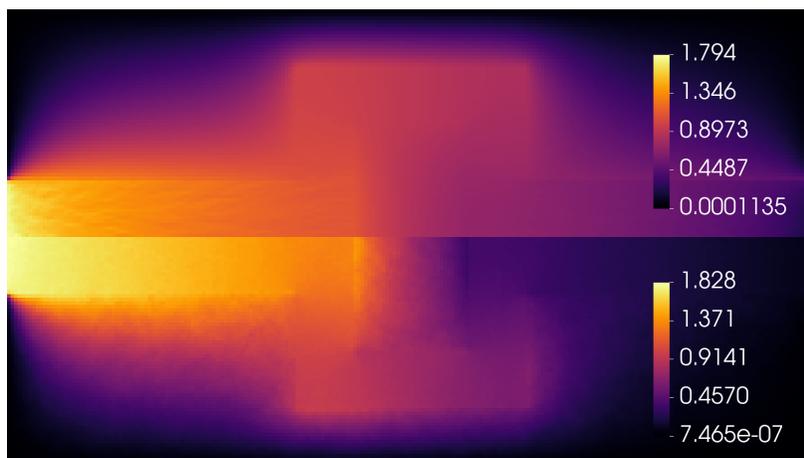


(b)

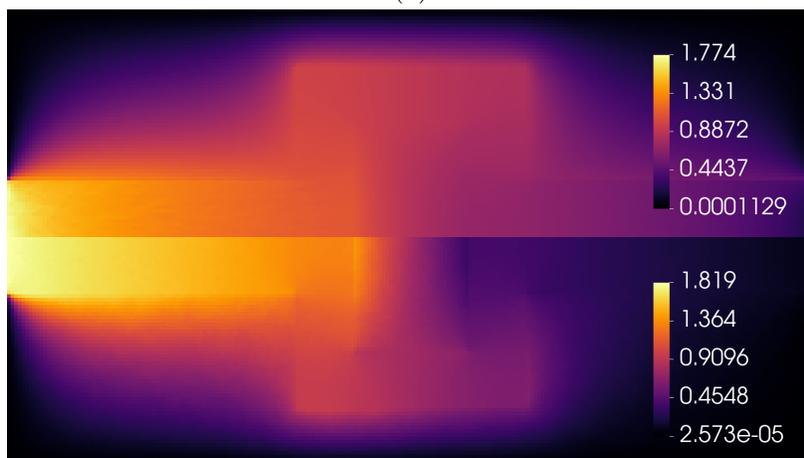


(c)

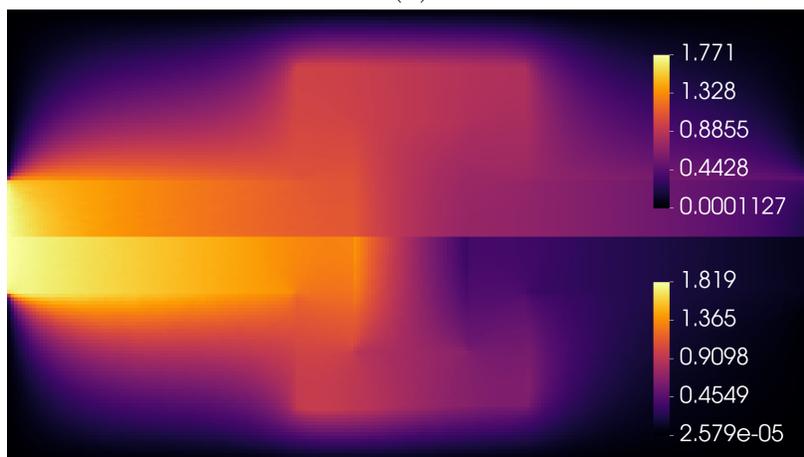
Figure 6.8: Solutions using $1e6$ (a), $4e6$ (b), and $16e6$ (c) particles with DHSM (top of a,b,c) and UMC (bottom of a,b,c) for the LCP problem with $\sigma_t^{\text{thick}}/\sigma_t^{\text{thin}} = 100$.



(a)



(b)



(c)

Figure 6.9: Solutions using 1e6 (a), 4e6 (b), and 16e6 (c) particles with DHSM (top of a,b,c) and UMC (bottom of a,b,c) for the LCP problem with $\sigma_t^{\text{thick}}/\sigma_t^{\text{thin}} = 1000$.

6.3.2 Confirming Avoidance of Noise Amplification

This section shows that the mixed FEM discretization of the SMM successfully avoids amplifying the noise in the SMM data for the LCP problem using the DHSM method. In section 5.3.2, I demonstrated amplification avoidance in the HSM method by plotting different HSM iterates for a calculation using 500 million particles. I do the same here with DHSM, but using only 500 thousand particles³. The calculations in this section use DHSM to solve the LCP problem as defined in [39], for which $\sigma_t^{\text{thick}}/\sigma_t^{\text{thin}} = 1000$. I plot unconverged quantities to show when the noise appears and how it changes during the DHSM iteration.

Below, I present plots of DHSM iterates alongside Algorithm 3, which is the HSM algorithm. The DHSM logic may be described using Algorithm 3 with the following changes,

- The `sm()` function must be modified to additionally compute and return the gradient of φ along with φ on the domain boundary. The new code in Algorithm 3 would be:

$$11: \varphi^{(i)}, \nabla\varphi^{(i)}, \varphi_{\Gamma_b}^{(i)} \leftarrow \text{sm}(Q_0, \mathbf{Q}_1, \hat{\mathbf{T}}^{(i-1)}, \hat{\beta}^{(i-1)})$$

- The `mc()` function must be modified to i) sample the DHSM variable volume source instead of the HSM variable volume source, and ii) additionally sample the variable boundary source⁴. The new code in Algorithm 3 would be:

$$12: \hat{\phi}_{\text{temp}}, \hat{\mathbf{T}}_{\text{temp}}, \hat{\beta}_{\text{temp}} \leftarrow \text{mc}(\varphi^{(i)}, \nabla\varphi^{(i)}, \varphi_{\Gamma_b}^{(i)}, \text{scattering_events})$$

- Finally, the return value must be modified because the DHSM estimator for the angle integrated intensity Eq. (1.17) is $\hat{\phi}_{\text{new}}$, which is defined by Eq. (3.195a). The new code in Algorithm 3 would be:

$$18: \text{return } \hat{\phi}^{(i)} + \varphi$$

Fig. 6.10 shows Algorithm 3 alongside pseudocolor plots of the unscattered estimator $\hat{\phi}_{\text{new}}^{(0)}$, the estimator first iterate $\hat{\phi}_{\text{new}}^{(1)}$, and the estimator at convergence $\hat{\phi}_{\text{new}}^{(6)}$. Fig. 6.11 shows the unscattered estimator $\hat{\phi}_{\text{new}}^{(0)}$ and the estimator first iterate $\hat{\phi}_{\text{new}}^{(1)}$ again, but with the second moment system solution iterate $\varphi^{(1)}$ interposed between them.

Fig. 6.10 (a) shows that the unscattered solution $\hat{\phi}_{\text{new}}^{(0)}$ is localized compared to the converged solution $\hat{\phi}_{\text{new}}^{(6)}$ in (c). Additionally, the noise appears to be preserved (neither dampened nor amplified) during the iteration, because the amount of noise in the converged iterate $\hat{\phi}_{\text{new}}^{(6)}$, shown in (c), is visibly similar to that of the first iterate $\hat{\phi}_{\text{new}}^{(1)}$, shown in (b).

The noise in the unscattered solution $\hat{\phi}_{\text{new}}^{(0)}$ is not amplified by the SMM solve, but rather slightly dampened, as judged by comparing Fig. 6.11 (a) to (b). Thus, the noise in the DHSM solution results from the Monte Carlo transport solve, not the moment system solve. Avoiding differentiation of the correction tensor by solving the first order moment system using a FEM to offload the derivative appears to be effective for the prevention of noise amplification in the DHSM method.

³I use a factor of 1000 fewer particles so that I can see the noise in the DHSM solution.

⁴Table 3.4 lists the variable sources for HSM and DHSM.

Algorithm Hybrid Second Moment

```

1: Input: user-provided boolean value HSM
2: if not HSM then
3:   scattering_events  $\leftarrow$  true
4:    $\hat{\phi} \leftarrow$  mc(q, scattering_events,  $\bar{\psi}$ )
5:   return  $\hat{\phi}$ 
6: end if
7: scattering_events  $\leftarrow$  false
8:  $\hat{\phi}^{(0)}, \hat{\mathbf{T}}^{(0)}, \hat{\beta}^{(0)} \leftarrow$  mc(q, scattering_events,  $\bar{\psi}$ )
9: i  $\leftarrow$  1
10: while not converged( $\hat{\phi}^{(i-1)}, \hat{\phi}^{(i)}$ ) do
11:    $\varphi^{(i)} \leftarrow$  sm(Q0, Q1,  $\hat{\mathbf{T}}^{(i-1)}, \hat{\beta}^{(i-1)}$ )
12:    $\hat{\phi}_{\text{temp}}, \hat{\mathbf{T}}_{\text{temp}}, \hat{\beta}_{\text{temp}} \leftarrow$  mc( $\varphi^{(i)}$ , scattering_events)
13:    $\hat{\phi}^{(i)} \leftarrow \hat{\phi}^{(0)} + \hat{\phi}_{\text{temp}}$ 
14:    $\hat{\mathbf{T}}^{(i)} \leftarrow \hat{\mathbf{T}}^{(0)} + \hat{\mathbf{T}}_{\text{temp}}$ 
15:    $\hat{\beta}^{(i)} \leftarrow \hat{\beta}^{(0)} + \hat{\beta}_{\text{temp}}$ 
16:   i  $\leftarrow$  i + 1
17: end while
18: return  $\hat{\phi}^{(i)}$ 

```



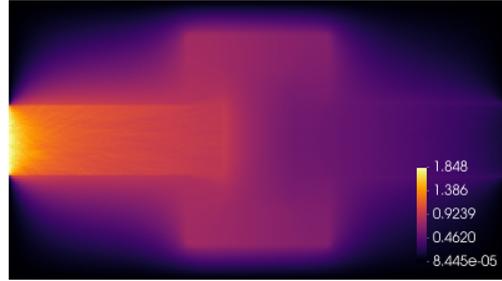
(a)

Algorithm Hybrid Second Moment

```

1: Input: user-provided boolean value HSM
2: if not HSM then
3:   scattering_events  $\leftarrow$  true
4:    $\hat{\phi} \leftarrow$  mc(q, scattering_events,  $\bar{\psi}$ )
5:   return  $\hat{\phi}$ 
6: end if
7: scattering_events  $\leftarrow$  false
8:  $\hat{\phi}^{(0)}, \hat{\mathbf{T}}^{(0)}, \hat{\beta}^{(0)} \leftarrow$  mc(q, scattering_events,  $\bar{\psi}$ )
9: i  $\leftarrow$  1
10: while not converged( $\hat{\phi}^{(i-1)}, \hat{\phi}^{(i)}$ ) do
11:    $\varphi^{(i)} \leftarrow$  sm(Q0, Q1,  $\hat{\mathbf{T}}^{(i-1)}, \hat{\beta}^{(i-1)}$ )
12:    $\hat{\phi}_{\text{temp}}, \hat{\mathbf{T}}_{\text{temp}}, \hat{\beta}_{\text{temp}} \leftarrow$  mc( $\varphi^{(i)}$ , scattering_events)
13:    $\hat{\phi}^{(i)} \leftarrow \hat{\phi}^{(0)} + \hat{\phi}_{\text{temp}}$ 
14:    $\hat{\mathbf{T}}^{(i)} \leftarrow \hat{\mathbf{T}}^{(0)} + \hat{\mathbf{T}}_{\text{temp}}$ 
15:    $\hat{\beta}^{(i)} \leftarrow \hat{\beta}^{(0)} + \hat{\beta}_{\text{temp}}$ 
16:   i  $\leftarrow$  i + 1
17: end while
18: return  $\hat{\phi}^{(i)}$ 

```



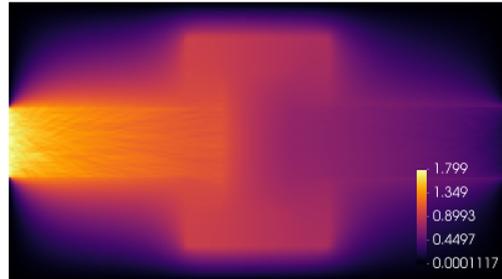
(b)

Algorithm Hybrid Second Moment

```

1: Input: user-provided boolean value HSM
2: if not HSM then
3:   scattering_events  $\leftarrow$  true
4:    $\hat{\phi} \leftarrow$  mc(q, scattering_events,  $\bar{\psi}$ )
5:   return  $\hat{\phi}$ 
6: end if
7: scattering_events  $\leftarrow$  false
8:  $\hat{\phi}^{(0)}, \hat{\mathbf{T}}^{(0)}, \hat{\beta}^{(0)} \leftarrow$  mc(q, scattering_events,  $\bar{\psi}$ )
9: i  $\leftarrow$  1
10: while not converged( $\hat{\phi}^{(i-1)}, \hat{\phi}^{(i)}$ ) do
11:    $\varphi^{(i)} \leftarrow$  sm(Q0, Q1,  $\hat{\mathbf{T}}^{(i-1)}, \hat{\beta}^{(i-1)}$ )
12:    $\hat{\phi}_{\text{temp}}, \hat{\mathbf{T}}_{\text{temp}}, \hat{\beta}_{\text{temp}} \leftarrow$  mc( $\varphi^{(i)}$ , scattering_events)
13:    $\hat{\phi}^{(i)} \leftarrow \hat{\phi}^{(0)} + \hat{\phi}_{\text{temp}}$ 
14:    $\hat{\mathbf{T}}^{(i)} \leftarrow \hat{\mathbf{T}}^{(0)} + \hat{\mathbf{T}}_{\text{temp}}$ 
15:    $\hat{\beta}^{(i)} \leftarrow \hat{\beta}^{(0)} + \hat{\beta}_{\text{temp}}$ 
16:   i  $\leftarrow$  i + 1
17: end while
18: return  $\hat{\phi}^{(i)}$ 

```



(c)

Figure 6.10: The DHSM estimator $\hat{\phi}_{\text{new}}$ for the LCP problem before iterating (a), after the first cycle (b), and at convergence (c), where 6 cycles were required to converge DHSM using 500 thousand particles.

Algorithm Hybrid Second Moment

```

1: Input: user-provided boolean value HSM
2: if not HSM then
3:   scattering_events  $\leftarrow$  true
4:    $\hat{\phi} \leftarrow$  mc(q, scattering_events,  $\bar{\psi}$ )
5:   return  $\hat{\phi}$ 
6: end if
7: scattering_events  $\leftarrow$  false
8:  $\hat{\phi}^{(0)}, \hat{\mathbf{T}}^{(0)}, \hat{\beta}^{(0)} \leftarrow$  mc(q, scattering_events,  $\bar{\psi}$ )
9: i  $\leftarrow$  1
10: while not converged( $\hat{\phi}^{(i-1)}, \hat{\phi}^{(i)}$ ) do
11:    $\varphi^{(i)} \leftarrow$  sm(Q0, Q1,  $\hat{\mathbf{T}}^{(i-1)}, \hat{\beta}^{(i-1)}$ )
12:    $\hat{\phi}_{\text{temp}}, \hat{\mathbf{T}}_{\text{temp}}, \hat{\beta}_{\text{temp}} \leftarrow$  mc( $\varphi^{(i)}$ , scattering_events)
13:    $\hat{\phi}^{(i)} \leftarrow \hat{\phi}^{(0)} + \hat{\phi}_{\text{temp}}$ 
14:    $\hat{\mathbf{T}}^{(i)} \leftarrow \hat{\mathbf{T}}^{(0)} + \hat{\mathbf{T}}_{\text{temp}}$ 
15:    $\hat{\beta}^{(i)} \leftarrow \hat{\beta}^{(0)} + \hat{\beta}_{\text{temp}}$ 
16:   i  $\leftarrow$  i + 1
17: end while
18: return  $\hat{\phi}^{(i)}$ 

```



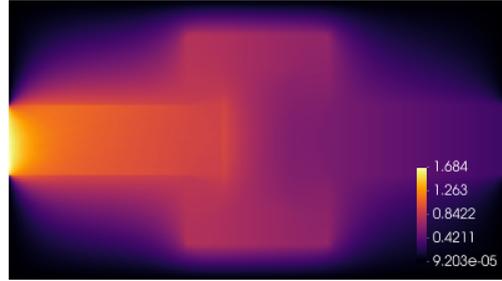
(a)

Algorithm Hybrid Second Moment

```

1: Input: user-provided boolean value HSM
2: if not HSM then
3:   scattering_events  $\leftarrow$  true
4:    $\hat{\phi} \leftarrow$  mc(q, scattering_events,  $\bar{\psi}$ )
5:   return  $\hat{\phi}$ 
6: end if
7: scattering_events  $\leftarrow$  false
8:  $\hat{\phi}^{(0)}, \hat{\mathbf{T}}^{(0)}, \hat{\beta}^{(0)} \leftarrow$  mc(q, scattering_events,  $\bar{\psi}$ )
9: i  $\leftarrow$  1
10: while not converged( $\hat{\phi}^{(i-1)}, \hat{\phi}^{(i)}$ ) do
11:    $\varphi^{(i)} \leftarrow$  sm(Q0, Q1,  $\hat{\mathbf{T}}^{(i-1)}, \hat{\beta}^{(i-1)}$ )
12:    $\hat{\phi}_{\text{temp}}, \hat{\mathbf{T}}_{\text{temp}}, \hat{\beta}_{\text{temp}} \leftarrow$  mc( $\varphi^{(i)}$ , scattering_events)
13:    $\hat{\phi}^{(i)} \leftarrow \hat{\phi}^{(0)} + \hat{\phi}_{\text{temp}}$ 
14:    $\hat{\mathbf{T}}^{(i)} \leftarrow \hat{\mathbf{T}}^{(0)} + \hat{\mathbf{T}}_{\text{temp}}$ 
15:    $\hat{\beta}^{(i)} \leftarrow \hat{\beta}^{(0)} + \hat{\beta}_{\text{temp}}$ 
16:   i  $\leftarrow$  i + 1
17: end while
18: return  $\hat{\phi}^{(i)}$ 

```



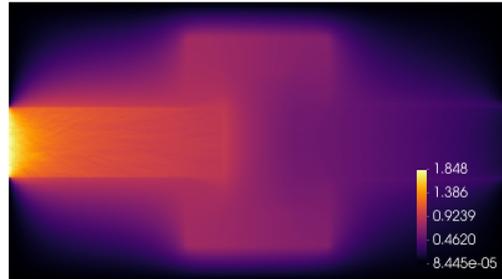
(b)

Algorithm Hybrid Second Moment

```

1: Input: user-provided boolean value HSM
2: if not HSM then
3:   scattering_events  $\leftarrow$  true
4:    $\hat{\phi} \leftarrow$  mc(q, scattering_events,  $\bar{\psi}$ )
5:   return  $\hat{\phi}$ 
6: end if
7: scattering_events  $\leftarrow$  false
8:  $\hat{\phi}^{(0)}, \hat{\mathbf{T}}^{(0)}, \hat{\beta}^{(0)} \leftarrow$  mc(q, scattering_events,  $\bar{\psi}$ )
9: i  $\leftarrow$  1
10: while not converged( $\hat{\phi}^{(i-1)}, \hat{\phi}^{(i)}$ ) do
11:    $\varphi^{(i)} \leftarrow$  sm(Q0, Q1,  $\hat{\mathbf{T}}^{(i-1)}, \hat{\beta}^{(i-1)}$ )
12:    $\hat{\phi}_{\text{temp}}, \hat{\mathbf{T}}_{\text{temp}}, \hat{\beta}_{\text{temp}} \leftarrow$  mc( $\varphi^{(i)}$ , scattering_events)
13:    $\hat{\phi}^{(i)} \leftarrow \hat{\phi}^{(0)} + \hat{\phi}_{\text{temp}}$ 
14:    $\hat{\mathbf{T}}^{(i)} \leftarrow \hat{\mathbf{T}}^{(0)} + \hat{\mathbf{T}}_{\text{temp}}$ 
15:    $\hat{\beta}^{(i)} \leftarrow \hat{\beta}^{(0)} + \hat{\beta}_{\text{temp}}$ 
16:   i  $\leftarrow$  i + 1
17: end while
18: return  $\hat{\phi}^{(i)}$ 

```



(c)

Figure 6.11: The DHSM estimator $\hat{\phi}_{\text{new}}$ for the LCP problem before iterating (a), φ in the first cycle (b), and $\hat{\phi}_{\text{new}}$ after cycle 1 (c). The pseudocolor plots in (a) and (c) are identical to the pseudocolor plots in Fig. 6.10 (a) and (b), respectively.

6.4 Linearized Lattice Problem

In section 5.4, I introduced the Lattice problem, which is a new problem for comparing numerical methods for TRT. The Lattice problem definition comes from Brunner [94]. It is an unsteady TRT problem. I solve a steady-state problem inspired by the Lattice problem, which I call the Linearized Lattice (LL). Fig. 5.19 shows the LL geometry, material data, and boundary conditions. The opacities are the same as in the LCP, but the geometry is a lattice instead of a pipe, and the inflow boundary extends along the entire left boundary of the domain, instead of just a subset of the left boundary. Additionally, LL has a fixed volume source that is uniformly distributed over a 1-cm-by-1-cm square at the center of the lattice. The boundary source and volume source functions are defined by Eqs. (5.26) and (5.27), respectively. My meshing strategy is the same as for the LCP: tessellate the LL geometry using the minimum number of equally-sized squares such that each square contains either optically-thick or optically-thin material, but not both. The result is a mesh with $7 \times 7 = 49$ elements. I refine the mesh five times, producing a mesh with $224 \times 224 = 50,176$ equally-sized square elements. Figs. 5.20 and 5.21 illustrate my meshing strategy.

In section 5.4, I ran HSM using 8 billion simulation particles, and plotted it alongside an unaccelerated Monte Carlo (UMC) solution calculated using 16 million simulation particles, and a radiation diffusion solution. Fig. 5.22 shows those results. Here, I replace the HSM solution plot with a plot of the DHSM solution using only 64 million particles. Fig. 6.12 demonstrates that 64 million particles were sufficient to eliminate any visibly-discernible noise from the DHSM solution. This represents over 100 times fewer particles than were required to produce the visibly-noiseless HSM solution shown in Fig. 5.22.

Just as we saw in section 6.3 with the DHSM solution to the LCP problem, the DHSM solution to the LL problem disagrees with the UMC solution. I believe my UMC solution to be more accurate than my hybrid method solution because it involves fewer approximations, and because my UMC solution to the LCP problem closely agreed with the S_N solution in [39]. Thus, the disagreement between Fig. 6.12 (a) and (b) demonstrates that the issue of the undefined derivative at the interface between two materials is present regardless of the geometry, because the interface issue arises in both LCP and LL calculations using DHSM. In section 3.8.2, I suggested a way to fix the issue, though I have not yet implemented my suggestion.

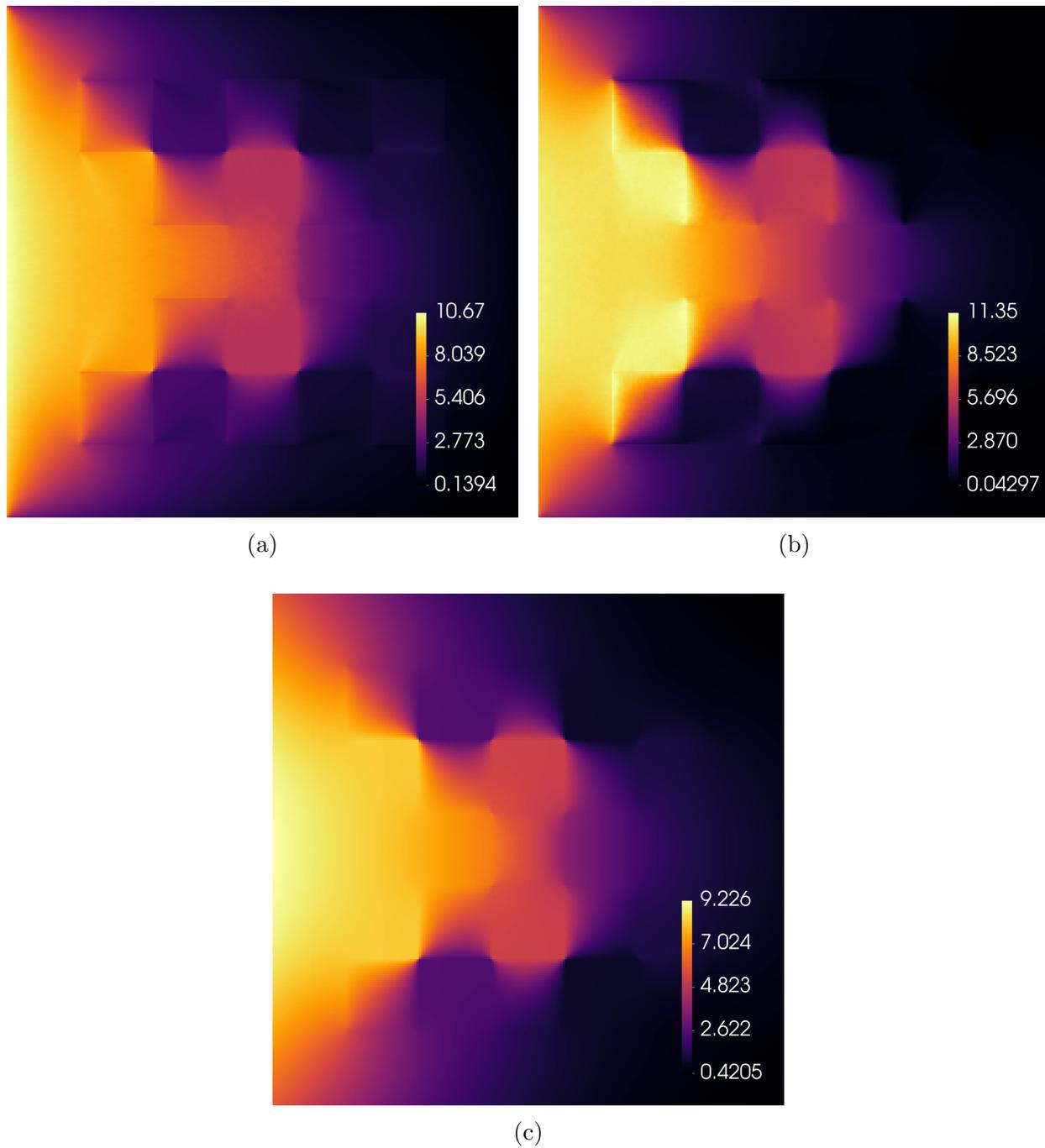


Figure 6.12: DHSM (a), UMC (b), and diffusion (c) solutions for the linearized lattice.

6.4.1 Particle-Equivalent Comparisons with Varied Optical Thicknesses

Despite an improvement of two orders of magnitude in the number of particles required for the DHSM solution to be visually noiseless, 64 million is still four times more particles than the 16 million particles required by UMC. In this section, I run the same problems from section 5.4.1 using DHSM instead of HSM. I vary the opacities in the LCP so that the ratio $\sigma_t^{\text{thick}}/\sigma_t^{\text{thin}}$ is equal to 2, 10, 100, and 1000. The highest ratio corresponds to the opacity values from the LL problem that I defined. The other three values are choices that I made to investigate the behavior of my hybrid method. Table 5.1 shows the complete material data for the four optical-thickness ratios.

Figs. 6.13 to 6.16 show pseudocolor plots of the results of 24 calculations. Each of the four figures has three subfigures with results for calculations using (a) one million, (b) four million, and (c) sixteen million simulation particles. Each subfigure contains two plots. The top plot is DHSM and the bottom plot is UMC. I dedicate one paragraph of comments on each of the four figures.

1) Comparing Fig. 6.13 to Fig. 5.23, specifically the top half of the (a) subfigures, DHSM is noticeably more noisy than HSM. This indicates that HSM may be more efficient for optically-thin calculations. The DHSM solution also shows less agreement with the UMC solution than the HSM solution, which is likely related to the aforementioned non-differentiability issue on the material interface. Finally, the DHSM solution violates the non-negativity property of the transport Eq. (1.13a).

2) Comparing Fig. 6.14 to Fig. 5.24, specifically the top half of the (a) subfigures, DHSM and HSM have approximately commensurate amounts of visible noise. Interestingly, the DHSM noise appears to be lower frequency than the HSM noise, in the sense that the DHSM noise manifests as unphysical streaking in the solution, whereas the HSM noise presents as unphysical spotting in the solution. However, the noise frequency difference between HSM and DHSM for the $\sigma_t^{\text{thin}} = 10$ LL calculation does not look as great as for the LCP calculation.

3) Comparing Fig. 6.15 to Fig. 5.25, specifically the top half of the (a) subfigures, DHSM is less noisy than HSM. Increasing the number of HSM particles by a factor of 16 significantly reduces the noise, though the resulting amount of noise is visually similar than the amount of noise in the DHSM calculation with 16 times fewer simulation particles.

4) A comparison of Fig. 6.16 to Fig. 5.26 demonstrates the most dramatic difference between the DHSM and HSM method, which occurs due to the regions of very optically-thick material in the problem. Not only do the HSM and DHSM solutions have visible noise, but the UMC solution does as well. Inspecting the (a) subfigures reveals that DHSM converges to a reasonable solution at only one million simulation particles, whereas the HSM iteration diverges, resulting in an unphysical solution characterized by a single bright spot, and the brightness of the unphysical bright spot is about six orders of magnitude too bright.

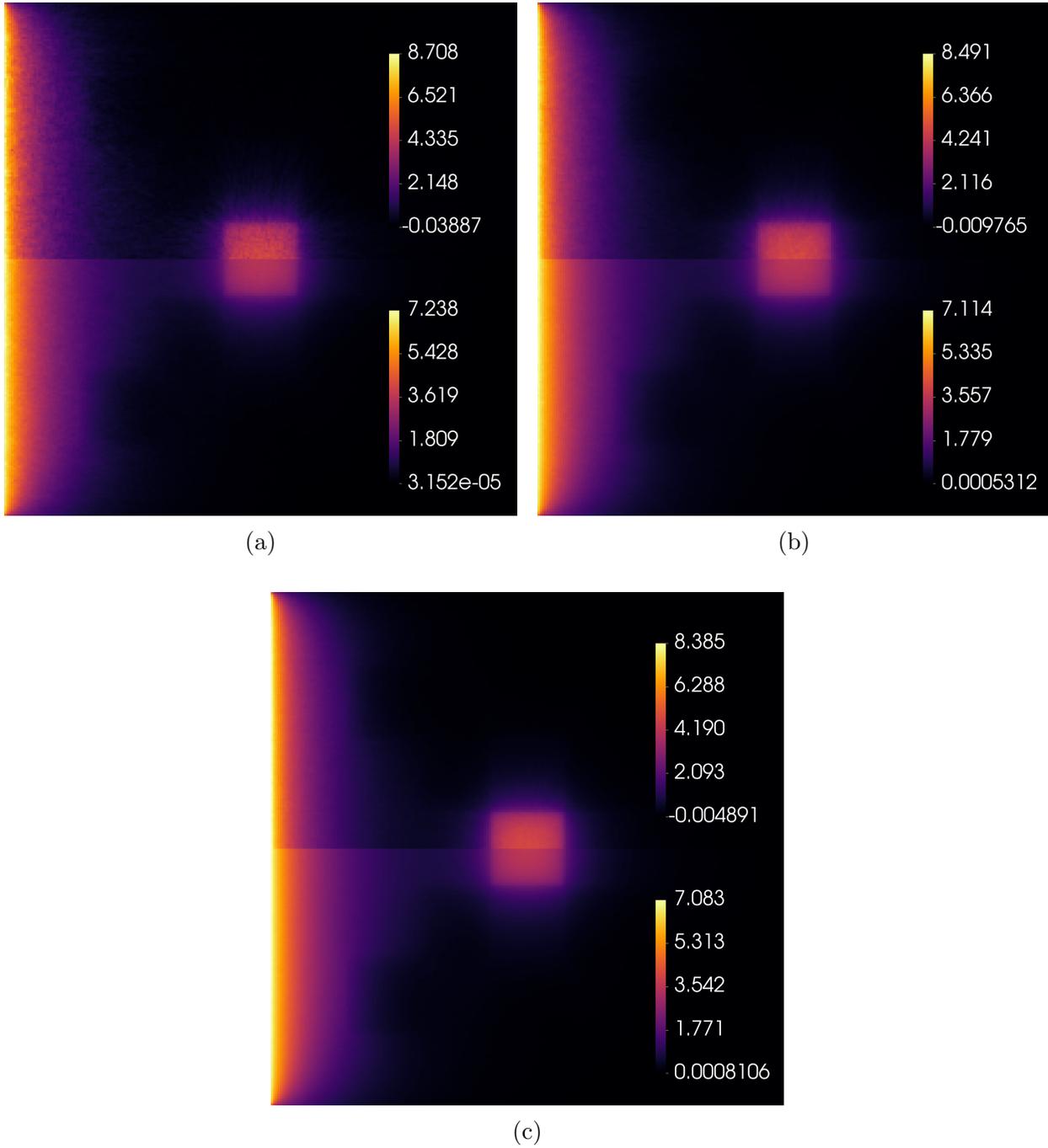


Figure 6.13: Solutions using 1e6 (a), 4e6 (b), and 16e6 (c) particles with DHSM (top of a,b,c) and UMC (bottom of a,b,c) for the LL problem with $\sigma_t^{\text{thick}}/\sigma_t^{\text{thin}} = 2$.

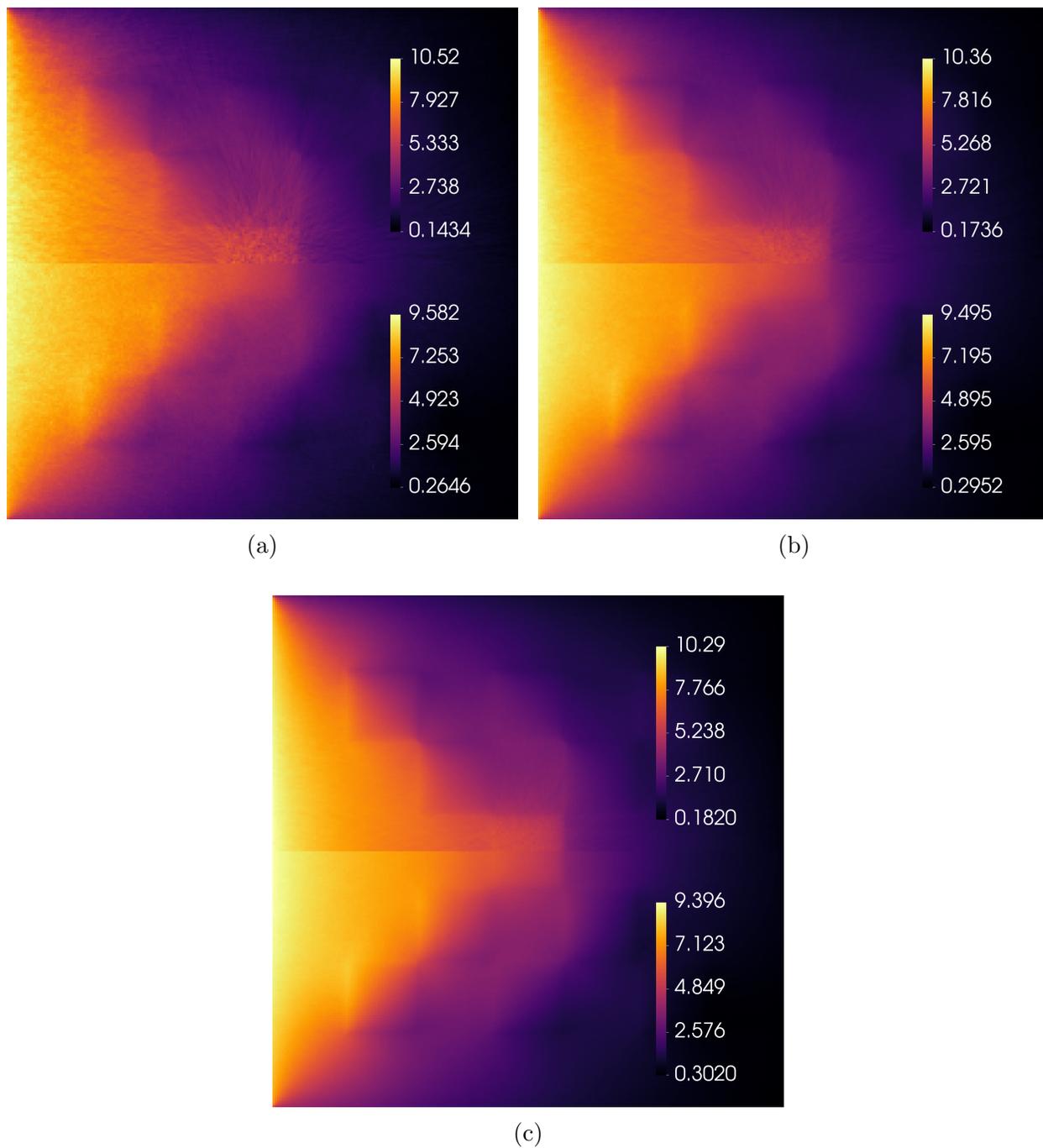


Figure 6.14: Solutions using 1e6 (a), 4e6 (b), and 16e6 (c) particles with DHSM (top of a,b,c) and UMC (bottom of a,b,c) for the LL problem with $\sigma_t^{\text{thick}}/\sigma_t^{\text{thin}} = 10$.

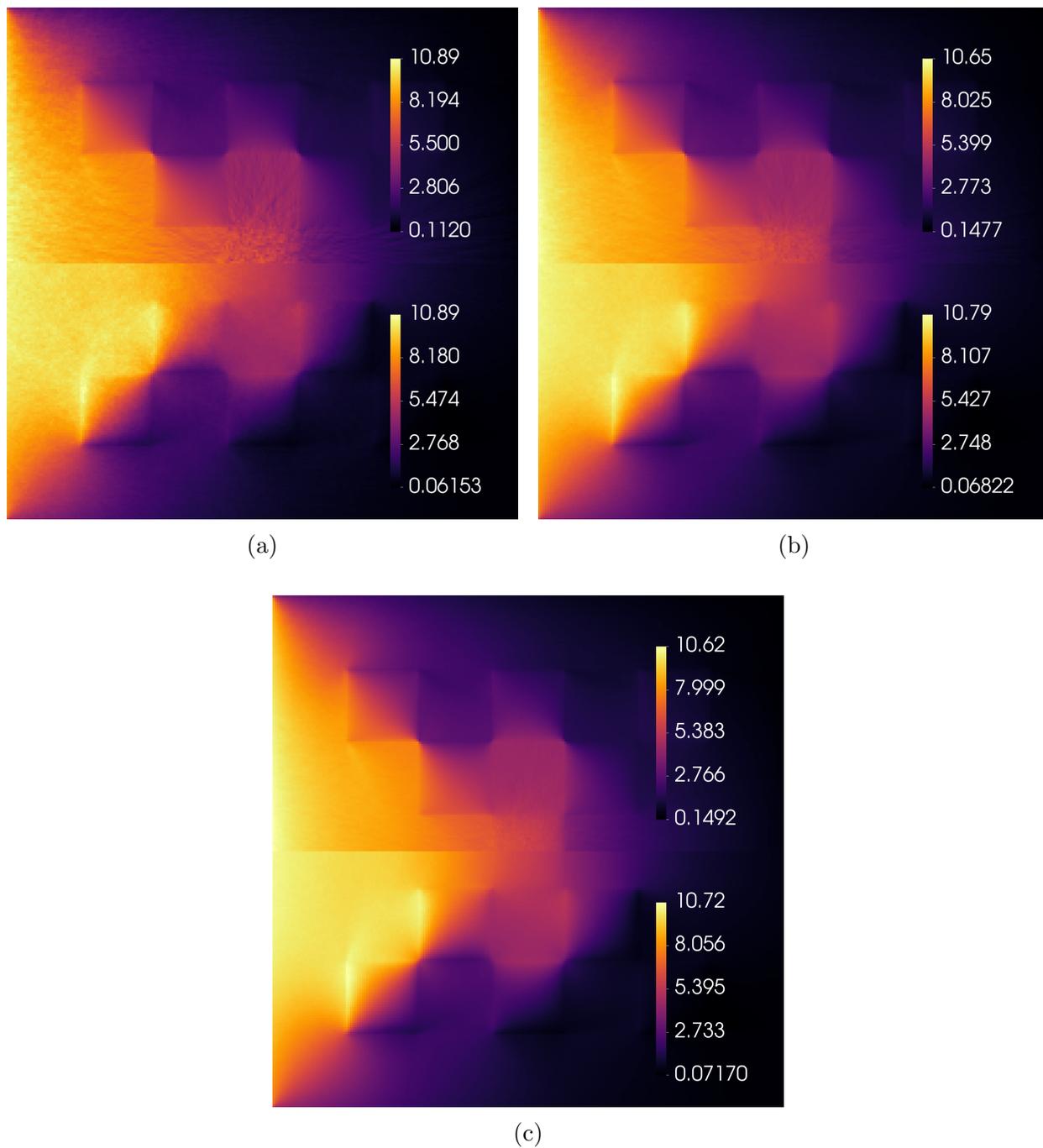


Figure 6.15: Solutions using 1e6 (a), 4e6 (b), and 16e6 (c) particles with DHSM (top of a,b,c) and UMC (bottom of a,b,c) for the LL problem with $\sigma_t^{\text{thick}}/\sigma_t^{\text{thin}} = 100$.

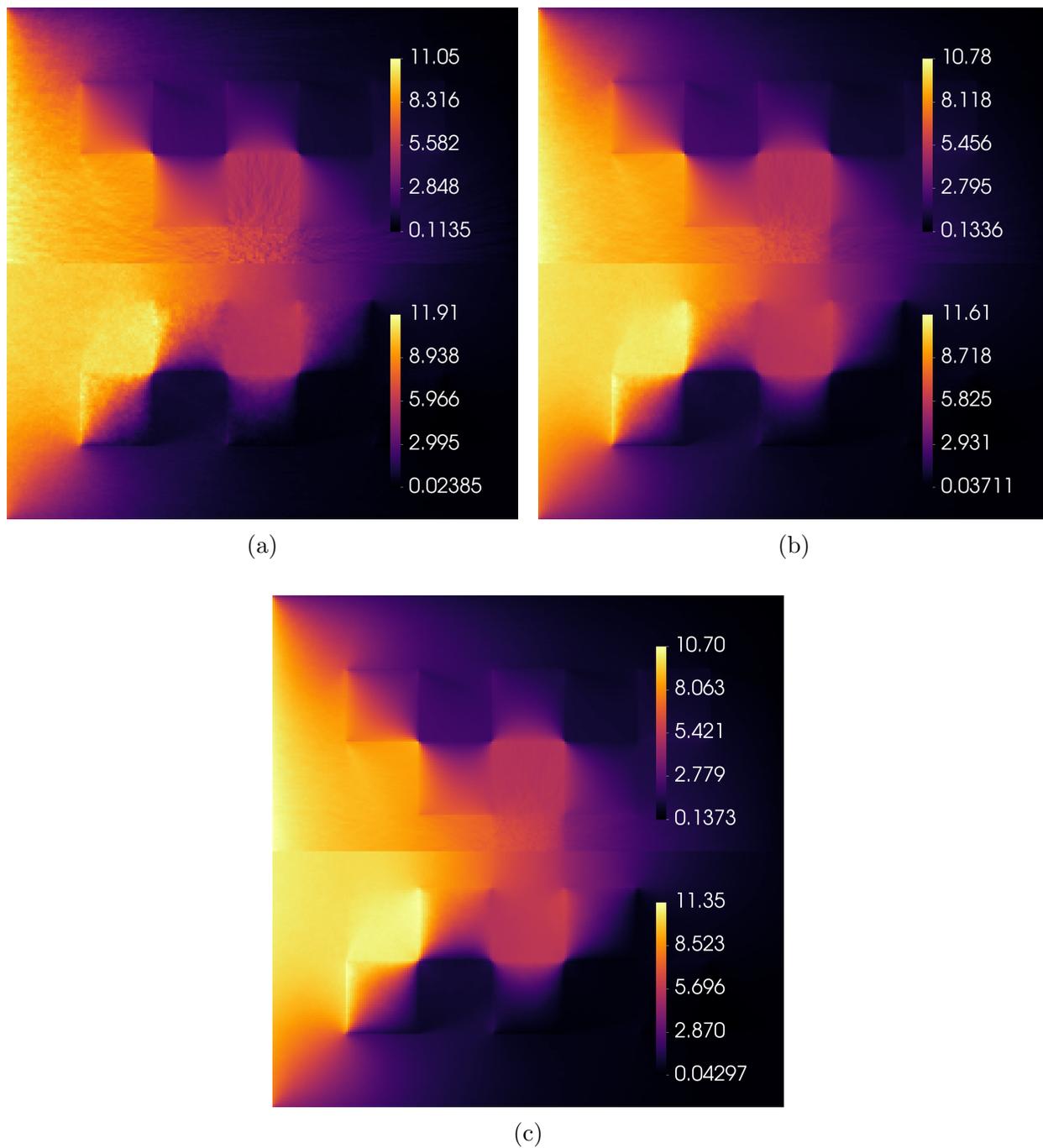


Figure 6.16: Solutions using 1e6 (a), 4e6 (b), and 16e6 (c) particles with DHSM (top of a,b,c) and UMC (bottom of a,b,c) for the LL problem with $\sigma_t^{\text{thick}}/\sigma_t^{\text{thin}} = 1000$.

6.4.2 Confirming Avoidance of Noise Amplification

This section shows that the mixed FEM discretization of the SMM successfully avoids amplifying the noise in the SMM data for the LL problem using the DHSM method. Combined with the results from section 6.3.2, which examined the LCP DHSM solution, the LL DHSM solution in this section provides confidence that DHSM avoids amplification regardless of geometry. This is because the LL is a lattice, whereas the LCP is a pipe.

In section 5.4.2, I demonstrated amplification avoidance in the HSM method by plotting different HSM iterates for a calculation using 500 million particles. I do the same here with DHSM, but using only 500 thousand particles⁵. The calculations in this section use DHSM to solve the LL problem that I defined in section 5.4, for which $\sigma_t^{\text{thick}}/\sigma_t^{\text{thin}} = 1000$. I plot unconverged quantities to show when the noise appears and how it changes during the DHSM iteration.

Below, I present plots of DHSM iterates alongside Algorithm 3, which is the HSM algorithm. In section 6.3.2, I showed how to modify Algorithm 3 so that it describes the DHSM logic instead of the HSM logic. These modifications also apply here in this section. The logic in Algorithm 3, and the DHSM modifications, are independent of the problem geometry.

Fig. 6.17 shows Algorithm 3 alongside pseudocolor plots of the unscattered estimator $\hat{\phi}_{\text{new}}^{(0)}$, the estimator first iterate $\hat{\phi}_{\text{new}}^{(1)}$, and the estimator at convergence $\hat{\phi}_{\text{new}}^{(8)}$. Fig. 6.18 shows the unscattered estimator $\hat{\phi}_{\text{new}}^{(0)}$ and the estimator first iterate $\hat{\phi}_{\text{new}}^{(1)}$ again, but with the second moment system solution iterate $\varphi^{(1)}$ interposed between them.

Fig. 6.17 (a) shows that the unscattered solution $\hat{\phi}_{\text{new}}^{(0)}$ is localized compared to the converged solution $\hat{\phi}_{\text{new}}^{(8)}$ in (c). Additionally, the noise appears to be preserved (neither dampened nor amplified) during the iteration, because the amount of noise in the converged iterate $\hat{\phi}_{\text{new}}^{(8)}$, shown in (c), is visibly similar to that of the first iterate $\hat{\phi}_{\text{new}}^{(1)}$, shown in (b).

The noise in the unscattered solution $\hat{\phi}_{\text{new}}^{(0)}$ is not amplified by the SMM solve, but rather slightly dampened, as judged by comparing Fig. 6.18 (a) to (b). Thus, the noise in the DHSM solution results from the Monte Carlo transport solve, not the moment system solve. Avoiding differentiation of the correction tensor by solving the first order moment system using a FEM to offload the derivative appears to be effective for the prevention of noise amplification in the DHSM method.

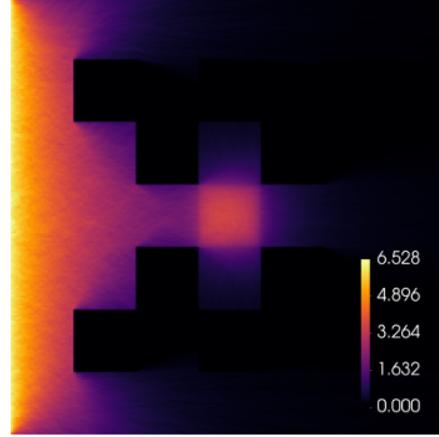
⁵I use a factor of 1000 fewer particles so that I can see the noise in the DHSM solution.

Algorithm Hybrid Second Moment

```

1: Input: user-provided boolean value HSM
2: if not HSM then
3:   scattering_events  $\leftarrow$  true
4:    $\hat{\phi} \leftarrow$  mc(q, scattering_events,  $\bar{\psi}$ )
5:   return  $\hat{\phi}$ 
6: end if
7: scattering_events  $\leftarrow$  false
8:  $\hat{\phi}^{(0)}, \hat{\mathbf{T}}^{(0)}, \hat{\beta}^{(0)} \leftarrow$  mc(q, scattering_events,  $\bar{\psi}$ )
9: i  $\leftarrow$  1
10: while not converged( $\hat{\phi}^{(i-1)}, \hat{\phi}^{(i)}$ ) do
11:    $\varphi^{(i)} \leftarrow$  sm(Q0, Q1,  $\hat{\mathbf{T}}^{(i-1)}, \hat{\beta}^{(i-1)}$ )
12:    $\hat{\phi}_{\text{temp}}, \hat{\mathbf{T}}_{\text{temp}}, \hat{\beta}_{\text{temp}} \leftarrow$  mc( $\varphi^{(i)}$ , scattering_events)
13:    $\hat{\phi}^{(i)} \leftarrow \hat{\phi}^{(0)} + \hat{\phi}_{\text{temp}}$ 
14:    $\hat{\mathbf{T}}^{(i)} \leftarrow \hat{\mathbf{T}}^{(0)} + \hat{\mathbf{T}}_{\text{temp}}$ 
15:    $\hat{\beta}^{(i)} \leftarrow \hat{\beta}^{(0)} + \hat{\beta}_{\text{temp}}$ 
16:   i  $\leftarrow$  i + 1
17: end while
18: return  $\hat{\phi}^{(i)}$ 

```

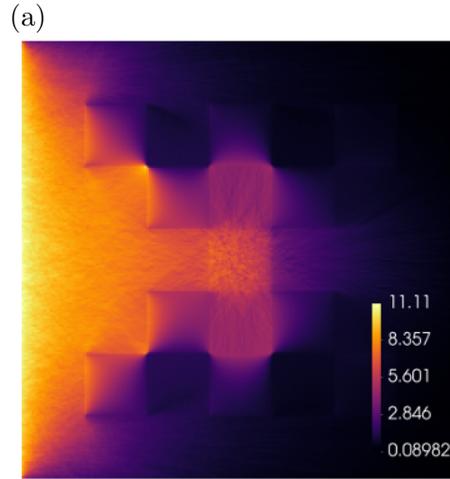


Algorithm Hybrid Second Moment

```

1: Input: user-provided boolean value HSM
2: if not HSM then
3:   scattering_events  $\leftarrow$  true
4:    $\hat{\phi} \leftarrow$  mc(q, scattering_events,  $\bar{\psi}$ )
5:   return  $\hat{\phi}$ 
6: end if
7: scattering_events  $\leftarrow$  false
8:  $\hat{\phi}^{(0)}, \hat{\mathbf{T}}^{(0)}, \hat{\beta}^{(0)} \leftarrow$  mc(q, scattering_events,  $\bar{\psi}$ )
9: i  $\leftarrow$  1
10: while not converged( $\hat{\phi}^{(i-1)}, \hat{\phi}^{(i)}$ ) do
11:    $\varphi^{(i)} \leftarrow$  sm(Q0, Q1,  $\hat{\mathbf{T}}^{(i-1)}, \hat{\beta}^{(i-1)}$ )
12:    $\hat{\phi}_{\text{temp}}, \hat{\mathbf{T}}_{\text{temp}}, \hat{\beta}_{\text{temp}} \leftarrow$  mc( $\varphi^{(i)}$ , scattering_events)
13:    $\hat{\phi}^{(i)} \leftarrow \hat{\phi}^{(0)} + \hat{\phi}_{\text{temp}}$ 
14:    $\hat{\mathbf{T}}^{(i)} \leftarrow \hat{\mathbf{T}}^{(0)} + \hat{\mathbf{T}}_{\text{temp}}$ 
15:    $\hat{\beta}^{(i)} \leftarrow \hat{\beta}^{(0)} + \hat{\beta}_{\text{temp}}$ 
16:   i  $\leftarrow$  i + 1
17: end while
18: return  $\hat{\phi}^{(i)}$ 

```

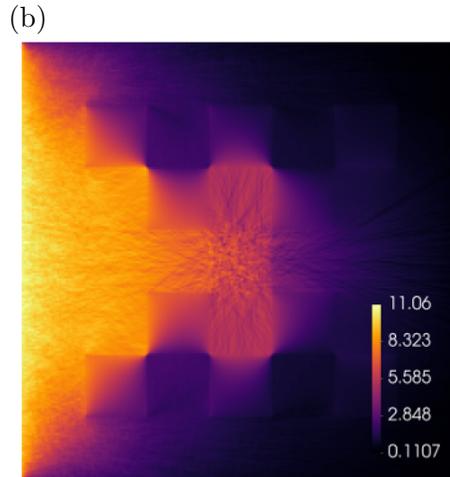


Algorithm Hybrid Second Moment

```

1: Input: user-provided boolean value HSM
2: if not HSM then
3:   scattering_events  $\leftarrow$  true
4:    $\hat{\phi} \leftarrow$  mc(q, scattering_events,  $\bar{\psi}$ )
5:   return  $\hat{\phi}$ 
6: end if
7: scattering_events  $\leftarrow$  false
8:  $\hat{\phi}^{(0)}, \hat{\mathbf{T}}^{(0)}, \hat{\beta}^{(0)} \leftarrow$  mc(q, scattering_events,  $\bar{\psi}$ )
9: i  $\leftarrow$  1
10: while not converged( $\hat{\phi}^{(i-1)}, \hat{\phi}^{(i)}$ ) do
11:    $\varphi^{(i)} \leftarrow$  sm(Q0, Q1,  $\hat{\mathbf{T}}^{(i-1)}, \hat{\beta}^{(i-1)}$ )
12:    $\hat{\phi}_{\text{temp}}, \hat{\mathbf{T}}_{\text{temp}}, \hat{\beta}_{\text{temp}} \leftarrow$  mc( $\varphi^{(i)}$ , scattering_events)
13:    $\hat{\phi}^{(i)} \leftarrow \hat{\phi}^{(0)} + \hat{\phi}_{\text{temp}}$ 
14:    $\hat{\mathbf{T}}^{(i)} \leftarrow \hat{\mathbf{T}}^{(0)} + \hat{\mathbf{T}}_{\text{temp}}$ 
15:    $\hat{\beta}^{(i)} \leftarrow \hat{\beta}^{(0)} + \hat{\beta}_{\text{temp}}$ 
16:   i  $\leftarrow$  i + 1
17: end while
18: return  $\hat{\phi}^{(i)}$ 

```



(c)

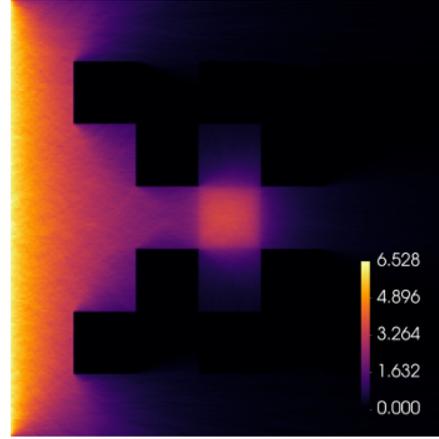
Figure 6.17: The DHSM estimator $\hat{\phi}_{\text{new}}$ for the LL problem before iterating (a), after the first cycle (b), and at convergence (c), where 8 cycles were required to converge DHSM using 500 thousand particles.

Algorithm Hybrid Second Moment

```

1: Input: user-provided boolean value HSM
2: if not HSM then
3:   scattering_events  $\leftarrow$  true
4:    $\hat{\phi} \leftarrow$  mc(g, scattering_events,  $\bar{\psi}$ )
5:   return  $\hat{\phi}$ 
6: end if
7: scattering_events  $\leftarrow$  false
8:  $\hat{\phi}^{(0)}, \hat{\mathbf{T}}^{(0)}, \hat{\beta}^{(0)} \leftarrow$  mc(g, scattering_events,  $\bar{\psi}$ )
9: i  $\leftarrow$  1
10: while not converged( $\hat{\phi}^{(i-1)}, \hat{\phi}^{(i)}$ ) do
11:    $\varphi^{(i)} \leftarrow$  sm(Q0, Q1,  $\hat{\mathbf{T}}^{(i-1)}, \hat{\beta}^{(i-1)}$ )
12:    $\hat{\phi}_{\text{temp}}, \hat{\mathbf{T}}_{\text{temp}}, \hat{\beta}_{\text{temp}} \leftarrow$  mc( $\varphi^{(i)}$ , scattering_events)
13:    $\hat{\phi}^{(i)} \leftarrow \hat{\phi}^{(0)} + \hat{\phi}_{\text{temp}}$ 
14:    $\hat{\mathbf{T}}^{(i)} \leftarrow \hat{\mathbf{T}}^{(0)} + \hat{\mathbf{T}}_{\text{temp}}$ 
15:    $\hat{\beta}^{(i)} \leftarrow \hat{\beta}^{(0)} + \hat{\beta}_{\text{temp}}$ 
16:   i  $\leftarrow$  i + 1
17: end while
18: return  $\hat{\phi}^{(i)}$ 

```



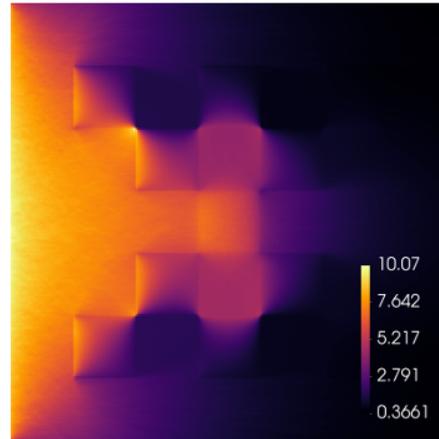
(a)

Algorithm Hybrid Second Moment

```

1: Input: user-provided boolean value HSM
2: if not HSM then
3:   scattering_events  $\leftarrow$  true
4:    $\hat{\phi} \leftarrow$  mc(g, scattering_events,  $\bar{\psi}$ )
5:   return  $\hat{\phi}$ 
6: end if
7: scattering_events  $\leftarrow$  false
8:  $\hat{\phi}^{(0)}, \hat{\mathbf{T}}^{(0)}, \hat{\beta}^{(0)} \leftarrow$  mc(g, scattering_events,  $\bar{\psi}$ )
9: i  $\leftarrow$  1
10: while not converged( $\hat{\phi}^{(i-1)}, \hat{\phi}^{(i)}$ ) do
11:    $\varphi^{(i)} \leftarrow$  sm(Q0, Q1,  $\hat{\mathbf{T}}^{(i-1)}, \hat{\beta}^{(i-1)}$ )
12:    $\hat{\phi}_{\text{temp}}, \hat{\mathbf{T}}_{\text{temp}}, \hat{\beta}_{\text{temp}} \leftarrow$  mc( $\varphi^{(i)}$ , scattering_events)
13:    $\hat{\phi}^{(i)} \leftarrow \hat{\phi}^{(0)} + \hat{\phi}_{\text{temp}}$ 
14:    $\hat{\mathbf{T}}^{(i)} \leftarrow \hat{\mathbf{T}}^{(0)} + \hat{\mathbf{T}}_{\text{temp}}$ 
15:    $\hat{\beta}^{(i)} \leftarrow \hat{\beta}^{(0)} + \hat{\beta}_{\text{temp}}$ 
16:   i  $\leftarrow$  i + 1
17: end while
18: return  $\hat{\phi}^{(i)}$ 

```



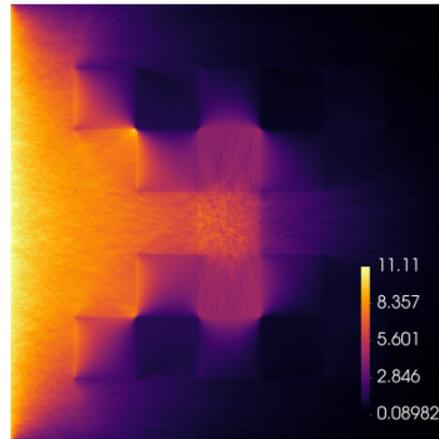
(b)

Algorithm Hybrid Second Moment

```

1: Input: user-provided boolean value HSM
2: if not HSM then
3:   scattering_events  $\leftarrow$  true
4:    $\hat{\phi} \leftarrow$  mc(g, scattering_events,  $\bar{\psi}$ )
5:   return  $\hat{\phi}$ 
6: end if
7: scattering_events  $\leftarrow$  false
8:  $\hat{\phi}^{(0)}, \hat{\mathbf{T}}^{(0)}, \hat{\beta}^{(0)} \leftarrow$  mc(g, scattering_events,  $\bar{\psi}$ )
9: i  $\leftarrow$  1
10: while not converged( $\hat{\phi}^{(i-1)}, \hat{\phi}^{(i)}$ ) do
11:    $\varphi^{(i)} \leftarrow$  sm(Q0, Q1,  $\hat{\mathbf{T}}^{(i-1)}, \hat{\beta}^{(i-1)}$ )
12:    $\hat{\phi}_{\text{temp}}, \hat{\mathbf{T}}_{\text{temp}}, \hat{\beta}_{\text{temp}} \leftarrow$  mc( $\varphi^{(i)}$ , scattering_events)
13:    $\hat{\phi}^{(i)} \leftarrow \hat{\phi}^{(0)} + \hat{\phi}_{\text{temp}}$ 
14:    $\hat{\mathbf{T}}^{(i)} \leftarrow \hat{\mathbf{T}}^{(0)} + \hat{\mathbf{T}}_{\text{temp}}$ 
15:    $\hat{\beta}^{(i)} \leftarrow \hat{\beta}^{(0)} + \hat{\beta}_{\text{temp}}$ 
16:   i  $\leftarrow$  i + 1
17: end while
18: return  $\hat{\phi}^{(i)}$ 

```



(c)

Figure 6.18: The DHSM estimator $\hat{\phi}_{\text{new}}$ for the LL problem before iterating (a), φ in the first cycle (b), and $\hat{\phi}_{\text{new}}$ after cycle 1 (c). The pseudocolor plots in (a) and (c) are identical to the pseudocolor plots in Fig. 6.17 (a) and (b), respectively.

Chapter 7

Conclusion

In this final chapter, I provide concluding remarks which make use of the physical and mathematical language developed in the preceding chapters. Thus, this chapter contains a synopsis of the dissertation which is more technical than what I provided in the dissertation abstract, and one which is more thorough as well. I provide this synopsis in sections 7.1 and 7.2. I end this conclusion with suggestions for future work in section 7.3, and a dissertation coda in section 7.4.

7.1 Hybrid Second Moment Method

My hybrid second moment (HSM) method is a novel method for solving the equations of thermal radiative transfer (TRT). The kernel of the TRT system is a linear Boltzmann transport equation for the radiation intensity, which arises from linearizations of the TRT system, an important example of which is the implicit Monte Carlo (IMC) linearization. My HSM method takes the standard, unaccelerated Monte Carlo (UMC) method for solving the transport equation using the IMC linearization and replaces it with the iteration in Algorithm 3. The iteration couples the transport equation with the moment system from the Second Moment Method (SMM). Fig. 1.3 shows the transport equation, the moment system, and the coupling terms: \mathbf{T} , β , and φ . In my HSM method, I use a deterministic finite element method (FEM) to compute φ in a procedure described in Chapter 2, and I compute estimates of \mathbf{T} and β using the Monte Carlo method for linear transport in a procedure described in Chapter 3. Algorithm 1 and Algorithm 2 are pseudocode for the deterministic finite element method procedure and the Monte Carlo procedure, respectively.

A good method for Boltzmann transport requires verification to ensure that the numerical method accurately computes the solution of the transport equation, and to ensure that the implementation achieves the expected convergence order, which I confirmed in section 5.1 with my presentation of the scaling study results in Fig. 5.4. It must also perform well in the thick diffusion limit (TDL), a challenging regime for transport methods characterized by arbitrarily optically-thick media. I presented TDL calculations in section 5.2, where I

showed that HSM dramatically outperforms UMC runtimes in Fig. 5.7, but also provided an empirical demonstration of the undesirable dependence of the HSM variance on the TDL parameter in Fig. 5.8.

My final HSM calculations, which considered the optically-thick and optically-thin multimaterial Linearized Crooked Pipe problem in section 5.3, and Linearized Lattice problem in section 5.4, demonstrate that HSM could be a competitive alternative to UMC were it not for the noise issue in the HSM solution. Fig. 5.12 demonstrated that my HSM, UMC, and diffusion implementations appear to be correct, because of how closely my pseudocolor triptych matches Fig 9 of Olivier *et al.* [39]. Finally, my Lattice problem pseudocolor triptych in Fig. 5.22 shows that the HSM noise issue appears to be independent of the geometry, because both the Linearized Crooked Pipe and the Linearized Lattice demonstrate the issue.

Both the Linearized Crooked Pipe (LCP) problem results in Figs. 5.17 and 5.18, and the Linearized Lattice (LL) problem results in Figs. 5.27 and 5.28 confirm that the mixed finite element solve of the SMM system does not amplify the Monte Carlo noise in the SMM data estimators $\hat{\mathbf{T}}$ and $\hat{\beta}$. I did this by presenting unconverged, or intermediate results en route to the converged HSM solution, which demonstrate that the noise issue is absent during sampling of the fixed source, absent during the first moment system solve, and finally manifests during the first sampling of the scattering source. Furthermore, the noise seeded at this point in the calculation is maintained for all following cycles through convergence, in the sense that the amplitude of the noise appears to be the same in the first and last iterates. This indicates that my choice to solve the first order SMM system using a FEM was effective at avoiding noise amplification.

7.2 Deviatoric Hybrid Second Moment Method

My deviatoric hybrid second moment (DHSM) method is a variation of my HSM method which fixes the noise issue. The main difference between the two methods is that DHSM involves computing the deviation of the intensity from isotropy, and this causes the variance of the DHSM estimator of the intensity to be significantly smaller than that of the HSM estimator.

In Chapter 6, I presented DHSM calculations for the same problems that I solved in Chapter 5 using HSM. My results in Chapter 6 show significantly less noise for a given number of Monte Carlo simulation particles, from which I conclude that DHSM fixes the HSM noise issue. The noise reduction provided by DHSM aligns with my expectation, as established by my derivation of the variance of the HSM and DHSM estimators in sections 3.6 and 3.8.3, respectively. My derivations show that the variance of the latter is significantly smaller than that of the former.

However, while addressing the noise issue, I inadvertently introduced another issue that arises in DHSM but not HSM: convergence to an incorrect solution around material interfaces. This new issue is caused by the non-differentiability of the intensity on material interfaces. In section 3.8.2, I describe how non-differentiability invalidates my deviatoric

estimator derivation. In the same section, I suggest how one might fix the issue by solving a minimization problem. I have yet to implement my suggestion. I suspect that its implementation could result in a method that is not only competitive with other IMC acceleration techniques, but potentially superior.

7.3 Future Work

The highest priority for future work on either my novel HSM method or its deviatoric variation (DHSM), is to fix DHSM so that it does not converge to an incorrect solution around material interfaces. The approach that I suggest attempts to fix the non-differentiability of the intensity by solving a minimization problem. If that works as expected, the next step would be to implement time dependence, calculate time dependent Crooked Pipe and Lattice problem solutions, and compare with UMC. If my method continues to perform well, then adding frequency dependence—the final piece required to create a full-dimensional TRT solver—seems reasonable. The path of future work would then open into areas which one could choose to pursue simultaneously:

1. Hydrodynamics Coupling. Hot matter not only radiates, but it can also flow, like a fluid. Hydrodynamic calculations can incorporate the effect of the radiation field through radiation pressure and radiation momentum terms, which appear as sources in the mass and momentum conservation equations in the Euler equations for fluid mechanics, respectively. Forming these terms simply requires computing the zeroth and first angular moments of the radiation intensity, which are Eqs. (1.17) and (1.18), respectively.
2. Physical Scattering. The TRT equations I considered in this dissertation do not have a scattering term. Thus, the scattering in my linear transport equation is a proxy for effective scattering, not physical scattering. Common physical scattering mechanisms present in astrophysical phenomena and applied physics experiments include Thomson scattering and Compton scattering. The latter process can shift the radiation frequency spectrum dramatically. Incorporating physical scattering processes such as these can be essential for predictive science modeling.
3. Graphics Processors. The architectural shift in general-purpose computing from central processors to application-specific integrated circuits, initiated by successes using graphics processors for calculations which have nothing to do with computer graphics, seems to indicate that TRT methods which can run on different architectures are more useful than single-architecture methods restricted to central processors. The Monte Carlo component of HSM and DHSM, which consumes a majority of the calculation runtime, would be the focus of a graphics processor porting effort. One might consult existing research on the subject of porting Monte Carlo transport to graphics processors [76, 95–102]. If the moment solve becomes the bottleneck, then one might consider

re-computing the right-hand side and solving the linear system on the device. One example of a linear solver for the mixed FEM problem that was expressly designed for graphics processors is Pazner *et al.* [62].

4. Spectral Line Transport. The applied physics experiment described in Appendix A contains plasma which is not collisionally dominated, therefore *thermal* radiative transfer is an inadequate approximation of the radiative transfer problem that one must solve in order to accurately model the experiment. One could correct this problem by augmenting the TRT system with additional equations describing the populations of the ion excitation states. Chapter 9 “Spectral line transport” in Castor [3] describes how to do this in general, and one could adapt Castor’s description to HSM.
5. High Order. My HSM method, and most of its deviatoric variation (DHSM), are lowest-order. All the quantities in HSM, and nearly all quantities in DHSM, are piecewise constant, meaning that the quantities are single-valued within an element. Are there any advantages to using linear, quadratic, or higher order representations? One example where there could be an advantage in runtime and physics fidelity is when the hydrodynamics capability with which one couples a TRT solver, such as HSM or DHSM, computes a high order solution on a curved mesh, as in Dobrev *et al.* [103]. Coupling lowest-order HSM or DHSM to such a capability requires a refinement-and-coarsening operation which adds to the calculation runtime and introduces additional numerical error. For an example of refinement-and-coarsening, see Section 4.2.4 “HO to LOR to HO” in [104].

The future is full of interesting work for novel TRT methods like HSM and its deviatoric variation, DHSM.

7.4 Coda

In this dissertation, I describe a novel method for TRT, called HSM, and a variation called DHSM. I demonstrate the viability of my method by calculating the solution to gray, steady-state, linear transport problems. I verified that my HSM method can accurately calculate the transport solution, and that the accuracy increases under mesh refinement and MC sample augmentation, meaning that the numerical error and uncertainty in HSM can be systematically attenuated. The same is true for DHSM, except around material interfaces, where DHSM converges to an incorrect solution. I showed that my choice to solve the first order second moment system using a FEM successfully avoided amplification of MC noise in the correction tensor. I discovered a noise issue in HSM which worsens with optical thickness. I fixed the noise issue by deriving and implementing a variation of HSM, called DHSM, in which I compute an alternative estimator for the SMM correction tensor which has significantly lower variance. I concluded my dissertation with a description of future work that must be completed in order to turn my HSM method, or its deviatoric variation, into a

production capability for the simulation of hot matter in multiphysics models of astrophysical and applied physics phenomena, an example of which I describe in Appendix A.

Bibliography

- [1] *Review of equilibria*, https://casper.astro.berkeley.edu/astrobaki/index.php/Review_of_Equilibria, Written by the faculty and staff of the University of California, Berkeley for the “Radiative Processes in Astronomy” course website. Accessed: 2024-07-01.
- [2] D. Attwood and A. Sakdinawat, *X-Rays and Extreme Ultraviolet Radiation: Principles and Applications*, 2nd ed. Cambridge University Press, 2017.
- [3] J. I. Castor, *Radiation Hydrodynamics*. Cambridge University Press, 2004. DOI: 10.1017/CB09780511536182.
- [4] G. Pomraning, *The Equations of Radiation Hydrodynamics*. Dover Publications, 1973.
- [5] J. A. Fleck and J. D. Cummings, “An implicit monte carlo scheme for calculating time and frequency dependent nonlinear radiation transport,” *J. Comput. Phys.*, vol. 8, pp. 313–342, 1971.
- [6] A. B. Wollaber, “Four decades of implicit monte carlo,” *Journal of Computational and Theoretical Transport*, Feb. 2016. DOI: 10.1080/23324309.2016.1138132.
- [7] S. S. Olivier, “High-order moment methods for thermal radiative transfer,” PhD dissertation, University of California, Berkeley, Berkeley, CA, 2022.
- [8] E. Lewis and W. Miller, *Computational Methods of Neutron Transport*. Wiley, 1984.
- [9] E. W. Larsen, J. E. Morel, and W. F. Miller Jr., “Asymptotic solutions of numerical transport problems in optically thick, diffusive regimes,” *J. Comput. Phys.*, vol. 69, no. 2, pp. 283–324, 1987. DOI: [https://doi.org/10.1016/0021-9991\(87\)90170-7](https://doi.org/10.1016/0021-9991(87)90170-7).
- [10] R. E. Alcouffe, “Diffusion synthetic acceleration methods for the diamond-differenced discrete-ordinates equations,” *Nuclear Science and Engineering*, vol. 64, no. 2, pp. 344–355, 1977. DOI: 10.13182/NSE77-1.
- [11] E. W. Larsen, “Diffusion-synthetic acceleration methods for discrete-ordinates problems,” *Transport Theory and Statistical Physics*, vol. 13, no. 1-2, pp. 107–126, 1984.
- [12] V. Y. Gol’din, “A quasi-diffusion method of solving the kinetic equation,” *USSR Computational Mathematics and Mathematical Physics*, vol. 4, pp. 136–149, 1964.
- [13] M. L. Adams and E. W. Larsen, “Fast iterative methods for discrete-ordinates particle transport calculations,” *Progress in Nuclear Energy*, vol. 40, no. 1, pp. 3–159, 2002.

- [14] 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,” *J. Comput. Phys.*, vol. 54, pp. 508–523, 1984.
- [15] N. Gentile, “Implicit monte carlo diffusion - an acceleration method for monte carlo time-dependent radiative transfer simulations,” *J. Comput. Phys.*, vol. 172, pp. 543–571, 2001.
- [16] M. A. Cleveland, N. A. Gentile, and T. S. Palmer, “An extension of implicit monte carlo diffusion: Multigroup and the difference formulation,” *Journal of Computational Physics*, vol. 229, no. 16, pp. 5707–5723, 2010. DOI: <https://doi.org/10.1016/j.jcp.2010.04.004>.
- [17] M. A. Cleveland and N. Gentile, “Mitigating teleportation error in frequency-dependent hybrid implicit monte carlo diffusion methods,” *Journal of Computational and Theoretical Transport*, vol. 43, no. 1-7, pp. 6–37, 2014. DOI: [10.1080/00411450.2014.909850](https://doi.org/10.1080/00411450.2014.909850).
- [18] J. D. Densmore, T. J. Urbatsch, T. M. Evans, and M. W. Buksas, “A hybrid transport-diffusion method for monte carlo radiative-transfer simulations,” *Journal of Computational Physics*, vol. 222, no. 2, pp. 485–503, 2007. DOI: <https://doi.org/10.1016/j.jcp.2006.07.031>.
- [19] J. D. Densmore, K. G. Thompson, and T. J. Urbatsch, “A hybrid transport-diffusion monte carlo method for frequency-dependent radiative-transfer simulations,” *Journal of Computational Physics*, vol. 231, no. 20, pp. 6924–6934, 2012. DOI: <https://doi.org/10.1016/j.jcp.2012.06.020>.
- [20] K. D. Lathrop, “Spatial differencing of the transport equation: Positivity vs. accuracy,” *J. Comput. Phys.*, vol. 4, no. 4, pp. 475–498, Dec. 1969. DOI: [10.1016/0021-9991\(69\)90015-1](https://doi.org/10.1016/0021-9991(69)90015-1).
- [21] F. B. Brown, “Monte carlo techniques for nuclear systems - theory lectures,” Los Alamos National Laboratory, Tech. Rep., 2016, LA-UR-16-29043.
- [22] J. Spanier, “Monte carlo methods,” in *Nuclear Computational Science: A Century in Review*. Dordrecht: Springer Netherlands, 2010, pp. 117–165. DOI: [10.1007/978-90-481-3411-3_3](https://doi.org/10.1007/978-90-481-3411-3_3).
- [23] J. Halton, “On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals,” *Numerische Mathematik*, vol. 2, pp. 84–90, 1960.
- [24] L. Chacón *et al.*, “Multiscale high-order/low-order (holo) algorithms and applications,” *Journal of Computational Physics*, vol. 330, pp. 21–45, 2017. DOI: <https://doi.org/10.1016/j.jcp.2016.10.069>.
- [25] J. Duderstadt and W. Martin, *Transport Theory* (Wiley-Interscience Publications). New York, 1979.

- [26] T. A. Brunner, “Forms of approximate radiation transport,” Sandia National Lab, USA, Tech. Rep., Jun. 2002. DOI: 10.2172/800993.
- [27] M. A. Cooper and E. W. Larsen, “Automated weight windows for global monte carlo particle transport calculations,” *Nuclear Science and Engineering*, vol. 137, pp. 1–13, 2001.
- [28] V. N. Novellino and D. Y. Anistratov, “Analysis of hybrid mc/deterministic methods for transport problems based on low-order equations discretized by finite volume schemes,” *Transaction of American Nuclear Society*, vol. 130, 2024.
- [29] V. N. Novellino and D. Y. Anistratov, “Convergence of multi-level hybrid monte carlo methods for 1-d particle transport problems,” in *Proceedings of the International Conference on Mathematics and Computational Methods Applied to Nuclear Science & Engineering (M&C 2025)*, Denver, Colorado, 2025.
- [30] H. Park, J. Densmore, A. Wollaber, D. Knoll, and R. Rauenzahn, “Monte carlo solution methods in a moment-based scale-bridging algorithm for thermal radiative transfer problems: Comparison with fleck and cummings,” in *Proceedings of the International Conference on Mathematics and Computational Methods Applied to Nuclear Science & Engineering (M&C 2013)*, Sun Valley, Idaho, 2013.
- [31] H. Park, D. A. Knoll, R. M. Rauenzahn, A. B. Wollaber, and J. D. Densmore, “A consistent, moment-based, multiscale solution approach for thermal radiative transfer problems,” *Transport Theory and Statistical Physics*, vol. 41, no. 3-4, pp. 284–303, 2012. DOI: 10.1080/00411450.2012.671224.
- [32] J. Willert, “Hybrid deterministic/monte carlo methods for solving the neutron transport equation and k-eigenvalue problem,” PhD dissertation, North Carolina State University, Raleigh, North Carolina, Jun. 2013.
- [33] D. A. Knoll, H. Park, and K. Smith, “Application of the jacobian-free newton-krylov method to nonlinear acceleration of transport source iteration in slab geometry,” *Nuclear Science and Engineering*, vol. 167, no. 2, pp. 122–132, 2011. DOI: 10.13182/NSE09-75.
- [34] A. Q. Lam, “Diffusion accelerated implicit monte carlo via nonlinear elimination for thermal radiative transfer,” PhD dissertation, Oregon State University, Corvallis, OR, 2021.
- [35] S. Pasmann, “Development of the iterative quasi-monte carlo (iqmc) method for neutron transport simulations,” PhD dissertation, University of Notre Dame, Notre Dame, IA, May 2024. DOI: 10.7274/25607787.v1.
- [36] T. S. Haut, P. G. Maginot, V. Z. Tomov, B. S. Southworth, T. A. Brunner, and T. S. Bailey, “An efficient sweep-based solver for the S_N equations on high-order meshes,” *Nuclear Science and Engineering*, 2019. DOI: 10.1080/00295639.2018.1562778.

- [37] B. C. Yee, S. S. Olivier, T. S. Haut, M. Holec, V. Z. Tomov, and P. G. Maginot, “A quadratic programming flux correction method for high-order DG discretizations of S_N transport,” *Journal of Computational Physics*, vol. 419, p. 109 696, 2020. DOI: <https://doi.org/10.1016/j.jcp.2020.109696>.
- [38] T. S. Haut, B. S. Southworth, P. G. Maginot, and V. Z. Tomov, “Diffusion synthetic acceleration preconditioning for discontinuous Galerkin discretizations of S_N transport on high-order curved meshes,” *SIAM Journal on Scientific Computing*, vol. 42, no. 5, B1271–B1301, 2020. DOI: 10.1137/19M124993X.
- [39] S. Olivier, W. Pazner, T. S. Haut, and B. C. Yee, “A family of independent variable eddington factor methods with efficient preconditioned iterative solvers,” *Journal of Computational Physics*, vol. 473, p. 111 747, 2023. DOI: <https://doi.org/10.1016/j.jcp.2022.111747>.
- [40] D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini, “Unified analysis of discontinuous Galerkin methods for elliptic problems,” *SIAM Journal on Numerical Analysis*, vol. 39, no. 5, pp. 1749–1779, 2002. DOI: 10.1137/S0036142901384162.
- [41] J. R. Shewchuk, “An introduction to the conjugate gradient method without the agonizing pain,” Carnegie Mellon University, USA, Tech. Rep., 1994.
- [42] G. R. Cefus and E. W. Larsen, “Stability analysis of the quasidiffusion and second moment methods for iteratively solving discrete-ordinates problems,” *Transport Theory and Statistical Physics*, vol. 18, no. 5-6, pp. 493–511, 1989. DOI: 10.1080/00411458908204700.
- [43] A. Quarteroni and A. Valli, *Numerical Approximation of Partial Differential Equations*. Springer, Berlin, Heidelberg, 1994. DOI: 10.1007/978-3-540-85268-1.
- [44] D. Boffi, F. Brezzi, and M. Fortin, *Mixed Finite Element Methods and Applications*. Springer, Berlin, Heidelberg, 2013. DOI: 10.1007/978-3-642-36519-5.
- [45] Raviart, P. A. and Thomas, J. M., “A mixed finite element method for 2-nd order elliptic problems,” in *Mathematical Aspects of Finite Element Methods: Proceedings of the Conference Held in Rome, December 10–12, 1975*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1977, pp. 292–315.
- [46] P. Raviart and J. Thomas, “Dual finite element models for second order elliptic problems,” *Energy Methods in Finite Element Analysis*, pp. 175–191, 1979.
- [47] J. H. Bramble and S. R. Hilbert, “Estimation of linear functionals on sobolev spaces with application to fourier transforms and spline interpolation,” *SIAM Journal on Numerical Analysis*, vol. 7, no. 1, pp. 112–124, 1970. DOI: 10.1137/0707006.
- [48] C. C. Paige and M. A. Saunders, “Solution of sparse indefinite systems of linear equations,” *SIAM Journal on Numerical Analysis*, vol. 12, no. 4, pp. 617–629, 1975. DOI: 10.1137/0712047.

- [49] Y. Saad and M. H. Schultz, “GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems,” *SIAM Journal on Scientific and Statistical Computing*, vol. 7, pp. 856–869, 1986.
- [50] J. W. Demmel, *Applied Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, 1997. DOI: 10.1137/1.9781611971446.
- [51] M. R. Hestenes and E. Stiefel, “Methods of conjugate gradients for solving linear systems,” *Journal of Research of the National Bureau of Standards*, vol. 49, no. 6, pp. 409–436, 1952.
- [52] M. R. Hestenes, *Conjugate Direction Methods in Optimization*. Springer-Verlag, New York, 1980.
- [53] A. Brandt, S. McCormick, and J. Ruge, “Algebraic multigrid (AMG) for sparse matrix equations,” *Sparsity and Its Applications*, pp. 257–284, 1984.
- [54] S. F. McCormick, *Multigrid Methods*, S. F. McCormick, Ed. Society for Industrial and Applied Mathematics, 1987. DOI: 10.1137/1.9781611971057.
- [55] W. L. Briggs, V. E. Henson, and S. F. McCormick, *A Multigrid Tutorial, Second Edition*, Second. Society for Industrial and Applied Mathematics, 2000. DOI: 10.1137/1.9780898719505.
- [56] B. Muldoon and M. Pozulp, “Fast solvers for the finite element method,” University of California, Berkeley, USA, Tech. Rep., 2022, <https://mike.pozulp.com/2022ffem.pdf>.
- [57] R. D. Falgout and U. M. Yang, “Hypre: A library of high performance preconditioners,” in *Computational Science — ICCS 2002*, P. M. A. Sloot, A. G. Hoekstra, C. J. K. Tan, and J. J. Dongarra, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 632–641.
- [58] H. A. van der Vorst, “Bi-cgstab: A fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems,” *SIAM Journal on Scientific and Statistical Computing*, vol. 13, no. 2, pp. 631–644, 1992. DOI: 10.1137/0913035.
- [59] A. H. Baker, E. R. Jessup, and T. Manteuffel, “A technique for accelerating the convergence of restarted gmres,” *SIAM Journal on Matrix Analysis and Applications*, vol. 26, no. 4, pp. 962–984, 2005. DOI: 10.1137/S0895479803422014.
- [60] Y. Saad, “A flexible inner-outer preconditioned gmres algorithm,” *SIAM Journal on Scientific Computing*, vol. 14, no. 2, pp. 461–469, 1993. DOI: 10.1137/0914028.
- [61] X. S. Li, “An overview of superlu: Algorithms, implementation, and user interface,” *ACM Trans. Math. Softw.*, vol. 31, no. 3, pp. 302–325, 2005. DOI: 10.1145/1089014.1089017.
- [62] W. Pazner, T. Kolev, and P. S. Vassilevski, “Matrix-free high-performance saddle-point solvers for high-order problems in $\mathbf{H}(\mathbf{div})$,” *SIAM Journal on Scientific Computing*, vol. 46, no. 3, pp. B179–B204, 2024. DOI: 10.1137/23M1568806.

- [63] L. M. Leemis, *Probability*. Lightning Source, 2011, <http://www.math.wm.edu/~leemis>.
- [64] I. Lux and L. Koblinger, *Monte Carlo Particle Transport Methods: Neutron and Photon Calculations*. 1991. DOI: 10.1201/9781351074834.
- [65] J. Spanier and E. Gelbard, *Monte Carlo Principles and Neutron Transport Problems*. Dover Publications, 2008.
- [66] J. A. Kulesza *et al.*, “Mcnp code version 6.3.1 theory & user manual,” May 2024. DOI: 10.2172/2372634.
- [67] Y. Azmy and E. Sartori, *Nuclear Computational Science, A Century in Review*. Jan. 2010, pp. 1–470. DOI: 10.1007/978-90-481-3411-3.
- [68] A. Burden, R. Burden, and J. Faires, *Numerical Analysis, 10th ed.* Jan. 2016. DOI: 10.13140/2.1.4830.2406.
- [69] Y. Nakatsukasa, *Approximate and integrate: Variance reduction in monte carlo integration via function approximation*, 2018.
- [70] A. Q. Lam, T. S. Palmer, T. A. Brunner, and R. M. Vega, “A monte carlo thermal radiative transfer solver with nonlinear elimination,” *Journal of Computational and Theoretical Transport*, vol. 52, no. 3, pp. 221–245, 2023. DOI: 10.1080/23324309.2023.2223410.
- [71] J. Askew, “A characteristics formulation of the neutron transport equation in complicated geometries,” United Kingdom Atomic Energy Authority, Tech. Rep., 1972.
- [72] J. R. Tramm, K. S. Smith, B. Forget, and A. R. Siegel, “The random ray method for neutral particle transport,” *Journal of Computational Physics*, vol. 342, pp. 229–252, 2017. DOI: <https://doi.org/10.1016/j.jcp.2017.04.038>.
- [73] D. E. Knuth, *The Art of Computer Programming, Volume 2: Seminumerical Algorithms*, 3rd. Reading, MA: Addison-Wesley, 1997.
- [74] C. R. Harris *et al.*, “Array programming with numpy,” *Nature*, vol. 585, no. 7825, pp. 357–362, 2020. DOI: 10.1038/s41586-020-2649-2.
- [75] F. B. Brown and W. R. Martin, “Monte carlo methods for radiation transport analysis on vector computers,” *Progress in Nuclear Energy*, vol. 14, no. 3, pp. 269–299, 1984. DOI: [https://doi.org/10.1016/0149-1970\(84\)90024-6](https://doi.org/10.1016/0149-1970(84)90024-6).
- [76] M. Pozulp *et al.*, “Progress porting llnl monte carlo transport codes to nvidia gpus,” in *Proceedings of the International Conference on Mathematics and Computational Methods Applied to Nuclear Science & Engineering (M&C 2023)*, Niagara Falls, Canada, 2023.
- [77] B. W. Kernighan and D. M. Ritchie, *The C programming language*. USA: Prentice Hall Press, 1989.

- [78] B. Stroustrup, *The C++ Programming Language*, 4th. Addison-Wesley Professional, 2013.
- [79] R. Anderson *et al.*, “Mfem: A modular finite element methods library,” *Computers & Mathematics with Applications*, vol. 81, pp. 42–74, 2021, Development and Application of Open-source Software for Problems with Numerical PDEs. DOI: <https://doi.org/10.1016/j.camwa.2020.06.009>.
- [80] D. K. Panda, H. Subramoni, C.-H. Chu, and M. Bayatpour, “The mvapich project: Transforming research into high-performance mpi library for hpc community,” *Journal of Computational Science*, vol. 52, p. 101 208, 2021, Case Studies in Translational Computer Science. DOI: <https://doi.org/10.1016/j.jocs.2020.101208>.
- [81] *Llnl rzwhippet cluster*, <https://hpc.llnl.gov/hardware/compute-platforms/rzwhippet>, Accessed: 2025-02-24.
- [82] E. A. León *et al.*, “Toss-2020: A commodity software stack for hpc,” in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, ser. SC '20, Atlanta, Georgia: IEEE Press, 2020.
- [83] R. Ierusalimsky, *Programming in Lua, Third Edition*, 3rd. Lua.Org, 2013.
- [84] L. Busby, *Irep*, Aug. 2016. DOI: 10.11578/dc.20180718.22.
- [85] *Hierarchical data format, version 5*, <https://github.com/HDFGroup/hdf5>.
- [86] *Gperftools: Google performance tools*, <https://github.com/gperftools/gperftools>.
- [87] *Pymfem: Mfem python interface*, <https://github.com/mfem/PyMFEM>.
- [88] M. Pozulp, T. Haut, P. Brantley, and J. Vujic, “An implicit monte carlo acceleration scheme,” in *Proceedings of the International Conference on Mathematics and Computational Methods Applied to Nuclear Science & Engineering (M&C 2023)*, Niagara Falls, Canada, 2023.
- [89] M. Pozulp, T. Haut, P. Brantley, S. Olivier, and J. Vujic, “A hybrid second moment method for thermal radiative transfer,” in *Proceedings of the International Conference on Mathematics and Computational Methods Applied to Nuclear Science & Engineering (M&C 2025)*, Denver, Colorado, 2025.
- [90] P. Roache, *Verification and Validation in Computational Science and Engineering*. Hermosa Publishers, 1998.
- [91] K. Salari and P. Knupp, “Code verification by the method of manufactured solutions,” Jun. 2000. DOI: 10.2172/759450.
- [92] P. J. Roache, “Code verification by the method of manufactured solutions,” *Journal of Fluids Engineering*, vol. 124, no. 1, pp. 4–10, Nov. 2001. DOI: 10.1115/1.1436090.
- [93] F. Graziani and J. LeBlanc, “The crooked pipe test problem,” UCRL-MI-143393, Tech. Rep., 2000.

- [94] T. A. Brunner, “A family of multi-dimensional thermal radiative transfer test problems,” Lawrence Livermore National Laboratory (LLNL), Livermore, CA (United States), Tech. Rep., Dec. 2023, LLNL-TR-858450. DOI: [10.2172/2280904](https://doi.org/10.2172/2280904).
- [95] S. Dawson *et al.*, “Porting llnl monte carlo transport codes to the amd instict mi300a apu,” in *Proceedings of the International Conference on Mathematics and Computational Methods Applied to Nuclear Science & Engineering (M&C 2025)*, Denver, Colorado, 2025.
- [96] R. M. Bergmann and J. L. Vujić, “Algorithmic choices in warp – a framework for continuous energy monte carlo neutron transport in general 3d geometries on gpus,” *Annals of Nuclear Energy*, vol. 77, pp. 176–193, 2015. DOI: <https://doi.org/10.1016/j.anucene.2014.10.039>.
- [97] R. Bergmann, K. Rowland, N. Radnović, R. Slaybaugh, and J. Vujić, “Performance and accuracy of criticality calculations performed using warp – a framework for continuous energy monte carlo neutron transport in general 3d geometries on gpus,” *Annals of Nuclear Energy*, vol. 103, 334–349, May 2017. DOI: [10.1016/j.anucene.2017.01.027](https://doi.org/10.1016/j.anucene.2017.01.027).
- [98] R. Bleile, P. Brantley, M. O’Brien, and H. Childs, “Algorithmic improvements for portable event-based monte carlo transport using the nvidia thrust library,” *Transactions American Nuclear Society*, vol. 115, pp. 535–538, 2016.
- [99] S. P. Hamilton, S. R. Slattery, and T. M. Evans, “Multigroup monte carlo on gpus: Comparison of history- and event-based algorithms,” *Annals of Nuclear Energy*, vol. 113, pp. 506–518, 2018.
- [100] S. Hamilton and T. Evans, “Continuous-energy monte carlo neutron transport on gpus in the shift code,” *Annals of Nuclear Energy*, vol. 128, pp. 236–247, 2019.
- [101] N. Choi, K. M. Kim, and H. Joo, “Optimization of neutron tracking algorithms for gpu-based continuous energy monte carlo calculation,” *Annals of Nuclear Energy*, vol. 162, p. 108508, Nov. 2021. DOI: [10.1016/j.anucene.2021.108508](https://doi.org/10.1016/j.anucene.2021.108508).
- [102] Tramm, John *et al.*, “Performance portable monte carlo particle transport on intel, nvidia, and amd gpus,” *EPJ Web Conf.*, vol. 302, p. 04010, 2024. DOI: [10.1051/epjconf/202430204010](https://doi.org/10.1051/epjconf/202430204010).
- [103] V. Dobrev, T. Kolev, and R. Rieben, “High-order curvilinear finite element methods for Lagrangian hydrodynamics,” *SIAM Journal on Scientific Computing*, vol. 34, B606–B641, 2012.
- [104] R. Anderson *et al.*, “The multiphysics on advanced platforms project,” Lawrence Livermore National Lab. (LLNL), Livermore, CA (United States), Tech. Rep., Nov. 2020. DOI: [10.2172/1724326](https://doi.org/10.2172/1724326).
- [105] J. Nuckolls, L. Wood, A. Thiessen, and G. Zimmerman, “Laser compression of matter to super-high densities: Thermonuclear (ctr) applications,” *Nature*, vol. 239, pp. 139–142, 1972.

- [106] J. D. Moody *et al.*, “Progress in hohlraum physics for the National Ignition Facility,” *Physics of Plasmas*, vol. 21, no. 5, p. 056317, May 2014. DOI: 10.1063/1.4876966.
- [107] E. L. Dewald *et al.*, “Dante soft x-ray power diagnostic for National Ignition Facility,” *Review of Scientific Instruments*, vol. 75, no. 10, pp. 3759–3761, Oct. 2004. DOI: 10.1063/1.1788872.
- [108] K. M. Campbell *et al.*, “Omega Dante soft x-ray power diagnostic component calibration at the National Synchrotron Light Source,” *Review of Scientific Instruments*, vol. 75, no. 10, pp. 3768–3771, Oct. 2004. DOI: 10.1063/1.1789603.
- [109] M. J. May *et al.*, “Understanding reconstructed Dante spectra using high resolution spectroscopy,” *Review of Scientific Instruments*, vol. 87, no. 11, 11E330, Aug. 2016. DOI: 10.1063/1.4961267.
- [110] O. A. Hurricane *et al.*, “Approaching a burning plasma on the nif,” *Physics of Plasmas*, vol. 26, no. 5, May 2019. DOI: 10.1063/1.5087256.
- [111] J. D. Lawson, “Some criteria for a power producing thermonuclear reactor,” *Proc. Phys. Soc. (London)*, vol. 70, no. 1, p. 6, Jan. 1957. DOI: 10.1088/0370-1301/70/1/303.
- [112] C. Tarter, *The American Lab: An Insider’s History of the Lawrence Livermore National Laboratory* (Johns Hopkins Nuclear History and Contemporary Affairs). Johns Hopkins University Press, 2018.
- [113] N. R. Council, *Review of the Department of Energy’s Inertial Confinement Fusion Program: The National Ignition Facility*. Washington, DC: The National Academies Press, 1997. DOI: 10.17226/5730.
- [114] H. Abu-Shawareb *et al.*, “Achievement of target gain larger than unity in an inertial fusion experiment,” *Phys. Rev. Lett.*, vol. 132, p. 065102, 6 2024. DOI: 10.1103/PhysRevLett.132.065102.
- [115] *From ‘60 minutes to ‘snl,’ fusion ignition news thrusts llnl into the zeitgeist*, <https://www.llnl.gov/article/50011/60-minutes-snl-fusion-ignition-news-thrusts-llnl-zeitgeist>, Accessed: 2024-06-26.
- [116] H. Abu-Shawareb *et al.*, “Lawson criterion for ignition exceeded in an inertial fusion experiment,” *Phys. Rev. Lett.*, vol. 129, p. 075001, 7 2022. DOI: 10.1103/PhysRevLett.129.075001.
- [117] A. B. Zylstra *et al.*, “Burning plasma achieved in inertial fusion,” *Nature (London)*, vol. 601, no. 7894, Jan. 2022. DOI: 10.1038/s41586-021-04281-w.
- [118] M. D. Rosen, “The long road to ignition: An eyewitness account,” *Physics of Plasmas*, vol. 31, no. 9, p. 090501, Sep. 2024. DOI: 10.1063/5.0221005.
- [119] J. Lindl, “Development of the indirect-drive approach to inertial confinement fusion and the target physics basis for ignition and gain,” *Physics of Plasmas*, vol. 2, no. 11, pp. 3933–4024, Nov. 1995. DOI: 10.1063/1.871025.

- [120] O. A. Hurricane *et al.*, “Energy principles of scientific breakeven in an inertial fusion experiment,” *Phys. Rev. Lett.*, vol. 132, p. 065103, 6 2024. DOI: 10.1103/PhysRevLett.132.065103.
- [121] L. Hopkins *et al.*, “Toward a burning plasma state using diamond ablator inertially confined fusion (icf) implosions on the national ignition facility (nif),” *Plasma Physics and Controlled Fusion*, vol. 61, 2019. DOI: 10.1088/1361-6587/aad97e.
- [122] D. S. Clark *et al.*, “Exploring implosion designs for increased compression on the National Ignition Facility using high density carbon ablaters,” *Physics of Plasmas*, vol. 29, no. 5, p. 052710, May 2022. DOI: 10.1063/5.0087052.

Appendix A

Laser Fusion Experiment

This appendix describes a laboratory astrophysics experiment that uses the world’s most energetic laser to achieve stellar conditions and induce nuclear fusion. The plasma in the experiment emits mostly soft x-ray radiation. The TRT equations described in this dissertation, which I solve with my novel hybrid method, are useful for modeling this spectral band. Some of the plasma in the experiment, for some of the duration of the experiment, does not have a well-defined temperature in the sense that I described in section 1.1.1. This means that modelling the experiment with TRT requires augmenting the TRT system with additional equations describing the populations of the ion excitation states. Solving the augmented TRT system is outside the scope of this dissertation. Nonetheless, the experiment highlights the importance of models like TRT, and the importance of having methods which can efficiently and accurately solve the TRT equations.

Why was the first fusion experiment to achieve target gain exceeding unity not realized until 2022, even though the technique used in the experiment was proposed in 1972 [105]? One reason was that important material data at the stellar conditions necessary to conduct the experiment had yet to be collected¹. Another reason was that the computer models used to simulate the experiment did not resolve all the relevant physics. Some examples are: laser energy deposition at high intensities where laser plasma instability is significant, time- and frequency-dependent radiation transport through heterogeneous media in which the photon mean free path varies over many orders of magnitude, and turbulent hydrodynamics in a highly convergent geometry starting from an initial condition containing substantial and subtle asphericity. Simulating this physics remains challenging today.

The physical quantities that must be calculated to model the experiment cannot be calculated in isolation. For instance, the stellar conditions of the experiment prevent the material temperature from being calculated without also calculating the intensity of the radiation. This requires coupling the equations for compressible hydrodynamics with the radiative transfer equation in a discipline known as radiation hydrodynamics.

¹Examples include a material constitutive relation for the material stress, another material constitutive relation for determining the material pressure given the material density and material internal energy, and also the material opacity given the material temperature, material density, and photon frequency.

Consider first the complications associated with calculating the material temperature. The distribution of kinetic energies of the particles that constitute the matter in the experiment cannot be characterized by a single value, which means that multiple “temperatures” must be calculated. Nor can it be characterized by two values, so calculations which assume that the electrons and ions have Maxwellian velocity distributions at distinct temperatures will be incorrect unless efforts are made to correct them.

Computing the radiation intensity is also complicated. The distribution of directions traveled by the photons which constitute the radiation in the experiment is not isotropic, therefore Eddington’s approximation is an incorrect simplification, and so is the radiation diffusion approximation, because Eddington’s approximation follows from the diffusion approximation [3]. This means that solving a radiation diffusion approximation to the radiative transfer equation will be incorrect unless efforts are made to correct the diffusion approximation.

Using the radiative transfer equation instead of its diffusion approximation obviates the need for these corrections and provides the computer model of the experiment with greater accuracy. However, the commonly-used simplification known as “thermal” radiative transfer, also known as the local thermodynamic equilibrium (LTE) regime approximation, will be incorrect because the transfer of energy between particles is *not* dominated by particle collisions. This means that the ionization and excitation distribution of the plasma is *not* Saha-Boltzmann, and the emission is *not* Planckian. Instead, the energy transfer is dominated by photonics, collisional-radiative models must be used to describe the ionization and excitation, and the emission is more complicated than the Planck function. Opacity is also complicated by the absence of a well-defined plasma temperature.

A.1 Description of the laser fusion experiment

The first fusion experiment to achieve target gain exceeding unity occurred in 2022 at the National Ignition Facility (NIF) within Lawrence Livermore National Laboratory (LLNL) located in Livermore, California, United States of America. The principal apparatus at NIF is the world’s most energetic laser, and the primary purpose of NIF is to collect experimental data necessary for the stewardship of the United States nuclear weapon stockpile. The facility cost about 3.5 billion dollars, required about twenty years to construct, and has been in operation for about ten years. The NIF annual budget is about 350 million dollars and the laser is fired about 350 times per year, so each experiment costs about one million dollars.

The most important NIF application aside from conducting experiments related to stockpile stewardship is to demonstrate the laser fusion concept by showing that laser-induced fusion reactions can produce more energy than the amount of energy that was required to induce the reactions. The NIF laser fusion experiment differs in some ways from the original laser fusion proposal, which was made over fifty years ago in 1972 [105], but the fundamental idea of imploding a fusion fuel capsule using a high energy laser remains the same. Specifically, the NIF laser fusion experiment operates in the following manner: a large bank of

capacitors is charged with about 300 MJ of electrical energy in a one minute period, and then discharged in a few hundred microseconds to energize flashlamps which induce a population inversion in a neodymium doped glass lasing medium to produce about 3 MJ of 1.064 micron infrared light. After a few trips through the lasing medium, half of the 192 beams are directed towards the top of a target chamber, which is an evacuated sphere ten meters in diameter, and the other half towards the bottom.

Immediately before entering the target chamber, the light passes through optics which convert the 3 MJ of infrared light to 2 MJ of 347 nanometer ultraviolet light, tripling its frequency. Energy is sacrificed to increase frequency because of the results of experiments in which gold disks were illuminated with high intensity laser light. The experiments demonstrated that higher frequency light achieved greater absorption, greater x-ray conversion efficiency, and reduced suprathreshold x-ray conversion efficiency for a fixed intensity². The 2 MJ of ultraviolet light enters the target chamber and illuminates the interior walls of a hollow cylinder target approximately 1 centimeter in height nearly uniformly. The cylinder is made of gold-lined depleted uranium. The cylinder is a solid until illumination, which turns the illuminated portion into a plasma, which radiates mostly soft x-rays.

The x-rays that do not escape from the cylinder or get absorbed in the gold travel towards the center of the cylinder and deposit their energy in a spherical fusion fuel capsule with a radius of about 1 millimeter. A thin membrane holds the capsule in the center of the cylinder. The outside of the capsule is made of dense carbon doped with tungsten. The dopant shields the fuel from hard x-rays, which would otherwise pass through the dense carbon exterior of the fuel capsule and heat the fuel at this early stage during which fuel heating is not desired³. The inside of the fusion fuel capsule contains equimolar deuterium and tritium gas, which is filled to a density that exceeds the capacity of the pressure vessel, therefore a cryogenic system creates a frozen fuel layer between the gaseous fuel and the dense carbon.

Energy deposition due to the absorption of soft x-rays in the carbon causes the carbon layer to explode outwards, inducing an inward implosion of the fusion fuel⁴. At peak compression, the radius of the fuel capsule is thirty times smaller than the original radius, and the volume is about 27,000 times smaller than the original volume. The kinetic energy of the implosion has been converted to internal energy of the fuel⁵. The fusion reaction rate accelerates in a central hotspot⁶. The dominant fusion reaction is,



in which deuterium and tritium fuse to produce a neutron with about 14.1 MeV of energy and an alpha particle with about 3.5 MeV of energy⁷. If the cold fuel surrounding the hotspot

²See Figure 8.30 through 8.32 in section 8.7.6 Laser Wavelength Trends in [2].

³Premature heating is called “preheat” and can also be caused by hot (“suprathreshold”) electrons.

⁴This phase of the experiment is called “the implosion”.

⁵This phase of the experiment is called “stagnation.”

⁶This phase of the experiment is called “burn”.

⁷An alpha particle is a helium nucleus, which is a deuterium-tritium (“DT”) fusion product.

is sufficiently dense, then the alpha particles will deposit their energy in the cold fuel, thus heating it and creating a burn wave that propagates outward from the hotspot. The burning fuel explodes outwards, ending the experiment.

About ten nanoseconds are required for the laser to deliver all 2 MJ of light to the target. The laser is held at a very low power for about half a nanosecond to burn the membranes which cover the top and bottom of the hollow cylinder. Then, the laser power increases in three jumps, with the powers of the jumps and the time between them chosen so that the three shocks which are triggered by the three jumps overtake each other shortly after they travel into the fusion fuel. The peak power of about 450 terawatts (TW), which is achieved as a result of the third and final jump, is sustained for about four nanoseconds. At 400 TW, the laser light intensity at peak power is about 5×10^{14} W/cm² at the focus of each group of four beams [106]. The 192 beams are divided into 48 groups of 4 called quads. Each beam in a quad is focused at the same spot on the target.

A panoply of measurements are used to evaluate the success of the experiment. The experimental measurement devices are called experimental “diagnostics” and their complexity approaches that of the experiment itself. Radiographs of the hotspot revealed oblate asphericity of the implosion, which was corrected through techniques that deliver relatively more laser energy to the equator of the fuel capsule, such as cross-beam energy transfer (CBET) from outer beams to inner beams. An 18-channel spectrometer, called “Dante”, measures the time-resolved x-ray intensity spectrum between 50 eV and 20 keV using a complicated arrangement of x-ray filters, mirrors, and diodes [107–109]. Ideally, the hollow cylinder that is illuminated by the laser re-emits the laser energy in a continuum radiation spectrum that closely fits a Planckian at about 300 eV, which has a peak intensity at a frequency of $2.82 * 300 = 846$ eV. The spectrum that Dante measures is an empirical assessment of the effectiveness of experimental design methodologies for the suppression of undesirable emission deviating from the ideal 300 eV Planckian.

The hotspot dynamics obey a power balance equation,

$$c_{\text{DT}} \frac{dT}{dt} = f_{\alpha} P_{\alpha} - f_b P_b - P_e - \frac{P}{m} \frac{dV}{dt}, \quad (\text{A.2})$$

where c_{DT} is the heat capacity of the equimolar deuterium-tritium fuel and T is the hotspot temperature, so the left-hand side is hotspot power [110]. On the right-hand side, $f_{\alpha} \leq 1$ is the fraction of alpha particles stopped in the hotspot, P_{α} is the power produced per deuterium-tritium fusion, f_b is the fraction of hotspot self-emission x-rays that escape the hotspot, P_b is the power due to bremsstrahlung emission of the hotspot, P_e is the power due to electron conduction, P is the hotspot pressure, m is the hotspot mass, and V is the hotspot volume.

The right-hand side of Eq. (A.2) is the sum of power gain and power loss terms. The first right-hand side term accounts for power gain due to alpha heating, the second for loss due to self-emission of bremsstrahlung radiation, and the third is loss due to electron conduction. The fourth term is a power gain during the implosion because the sign of the hotspot volume

time derivative is negative. The sign flips during the explosion and it becomes a power loss term.

The relative sizes of the terms in Eq. (A.2) are used in the definition of important and desirable experimental regimes. A “burning plasma” is achieved when the alpha heating term exceeds the other power gain term, which is power gain due to the conversion of the kinetic energy of the implosion to internal energy. The “Lawson criterion” is satisfied when the alpha heating term exceeds not only the other power gain term, but also substantially exceeds the modulus of the power loss terms⁸. Satisfaction of the Lawson criterion is also called “ignition”. In an influential report, the United States National Academy of Sciences (NAS) defined ignition as “target gain exceeding unity” [113].

The most important metric of experimental success is the target gain,

$$G_{\text{target}} = Y/E_{\text{target}} . \quad (\text{A.3})$$

The yield Y is the total energy produced by fusion reactions and E_{target} is the laser energy delivered to the target. The yield can be measured by counting the 14.1 MeV neutrons emitted by the DT fusion reactions, and then multiplying by the total energy output of a single DT fusion, which is about 17.6 MeV. This is the sum of the 14.1 MeV and the 3.5 MeV of kinetic energy carried by the neutron and the alpha particle, respectively. The energy of the reaction products is due to their velocities, so the neutron travels at about one sixth of the speed of light. Its mean free path is large with respect to the extent of the plasma, so nearly all of the neutrons escape, and thus the neutrons may be counted in a neutron detector directed at the plasma. Each neutron that arrives at the detector corresponds to one fusion reaction, so the number of fusion reactions is equal to the number of neutrons counted.

Experiments at NIF are identified with an alphanumeric character sequence of the format Nymmdd where N is used to indicate that the experiment was conducted at NIF, yy is replaced with the last two digits of the four digit year in which the experiment was conducted, mm is replaced with two digits designating the month, and dd is replaced with two digits designating the day. Experiments are colloquially called “shots”. The goal $G_{\text{target}} > 1$ was achieved after a little over ten years of NIF operations with $Y \approx 3$ MJ for $E_{\text{target}} \approx 2$ MJ, thus providing $G_{\text{target}} \approx 1.5$ in an experiment denoted N221205 (meaning NIF experiment in the year **2022**, month **12**, and day **05**) [114]. N221205 achieved the NAS definition of ignition ($G_{\text{target}} > 1$) and the experiment was reported in “tens of thousands of news stories that reached billions of people around the globe” [115]. N221205 produced about 10^{18} neutrons, which was about twice as many neutrons as the best previous shot.

The best shot preceding N221205 was the first shot to satisfy the Lawson criterion: N210808 [116]. This shot had $Y \approx 1.37$ MJ for $E_{\text{target}} = 1.9$ MJ and thus $G_{\text{target}} < 1$.

⁸Lawson’s criterion [111] is alternatively explained on page 153 of Tarter [112] as the product of the ion number density and the confinement time. This product must be large at a suitably high ion temperature, about 10 keV or so. Lawson’s criterion is applicable to both magnetic and laser fusion but achieved very differently. Magnetic fusion confines low density plasma for seconds and laser fusion confines high density plasma for nanoseconds.

N221205 produced nearly an order of magnitude more neutrons than the best shot preceding it, which were the first shots to achieve a burning plasma, which are described in a Nature article [117]. This diagram summarizes the shot progression just described,

$$\max(\text{all shots before N210808}) \xrightarrow{\sim 10x} \text{N210808} \xrightarrow{\sim 2x} \text{N221205},$$

where the $\sim 10x$ and $\sim 2x$ above the arrows denotes the approximate multiplicative increase in neutron yield. Table A.1 summarizes the definitional categories satisfied by the aforementioned shots. The journey that was required to achieve the results in table A.1, the beginning of which preceded the NIF, is illuminated in Rosen [118].

	$\max(\text{all shots before N210808})$	N210808	N221205
burning plasma	✓	✓	✓
Lawson criterion		✓	✓
$G_{\text{target}} > 1$			✓

Table A.1: Satisfaction of definitional categories for selected NIF shots.

A.2 Computer model of the laser fusion experiment

Computer models of the fusion experiment have been in development for over half a century. The largest section of the 1972 publication proposing the laser fusion experiment is titled “Computer Calculations.” It included results from a computer model which was used to predict the outcome of the proposed experiment and demonstrate that the experiment could succeed [105]. In the same year, LLNL created an internal collaboration and organizational entity colloquially known as the “laser program” to work towards achieving the proposed experiment. The program funded lasers that became the predecessors of the NIF laser. Generations of lasers were distinguished by their energy: a new generation of laser was signified by an increase in the laser energy delivered to the target.

The physics of laser fusion is highly non-linear and this makes it difficult to extrapolate experimental results to higher laser energies. A computer model which solves the non-linear equations governing the physics of the experiment could be used to predict the outcome of extrapolated experiments. This is what motivated the authors in [105] to include computer calculations. Their practice persists: every significant advancement towards achieving the experiment that they proposed has been accompanied by computer calculations.

Experiments on LLNL lasers preceding the NIF laser, accompanied by computer models that allowed extrapolation to laser energies exceeding the energy of any existing laser, resulted in a NIF design specification that its designers hoped would be sufficient to achieve the scientific milestones of ignition and target gain [119]. Fifteen years later, NIF construction was completed, the laser achieved “first light”, and the targets proposed in [119] were

fabricated, fueled, and fielded. The designers were disappointed: the yield of the best of these first shots was only 1 kJ, which was three orders of magnitude less than the 2 MJ gain threshold. Hundreds more shots, fielded over the next ten years, were required to get to ignition and target gain: three years to achieve 10 kJ, seven more years to achieve 100 kJ, a few more months to achieve 1 MJ and then 1 more year to exceed the 2 MJ gain threshold⁹.

The computer models which solved the non-linear equations governing the physics were overly-optimistic in their predicted yield. A target which burned on the computer twenty-five years ago fizzled when fielded at NIF. The computer models were deficient, but they were amended, and they served as essential tools for testing design variations. One important example of this computer-assisted experimental design process is the biggest target design modification in the history of NIF. Designers replaced high-gas-fill cylinders and plastic capsules with low-gas-fill cylinders and diamond capsules [121]. Computer models demonstrated improved implosion characteristics that were realized with the new design [122].

Computer models are also important because they are required for interpreting the results of the experiment. Consider once more Eq. (A.2). The four terms on the right-hand side are not accurately measurable. Instead, they must be computed in a simulation which is initialized with the experimental parameters supplying the initial condition, boundary conditions, and observed laser pulse. The four terms are computed in the calculation by solving 1) charged particle transport, 2) radiative transfer, 3) electron conduction, and 4) compressible hydrodynamics equations. The relative size of the four terms determines the categorization of the experiment (see table A.1).

⁹For a time-series plot of fusion yield from the year 2010 through 2025, see Figure 5 in [120].