

**A STUDY OF LOW ORDER SPHERICAL HARMONIC CLOSURES FOR
RAPID TRANSIENTS IN RADIATION TRANSPORT**

by

Kyeong Sam Oh

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Nuclear Engineering and Radiological Sciences)
in The University of Michigan
2009

Doctoral Committee:

Professor James P. Holloway, Chair
Professor Thomas J. Downar
Professor William R. Martin
Professor Kenneth G. Powell

© Kyeong Sam Oh 2009
All Rights Reserved

Dedicated to my advisor,
James P. Holloway

ACKNOWLEDGEMENTS

I would like to thank my advisor, Professor James Paul Holloway, for his support and advice. His encouragement and guidance during my studies were invaluable. Professor Holloway's attitude as a teacher and scientist, his enthusiasm and knowledge, and his insight to the mysteries of particle transport have all inspired me deeply. I would also like to thank him for carefully reviewing this manuscript.

I would like to thank the members of my doctoral committee, Professor Thomas Downar, Professor William Martin and Professor Kenneth Powell for their careful reviews of my dissertation and insightful suggestions.

I would like to express my gratitude to Professor John C. Lee for his comments and encouragement.

I would like to give my gratitude to my colleagues, Jae Cheon Kim, Troy Becker and Kaushik Banerjee for their practical help and suggestion to my research.

Finally, heartfelt thanks to my wife, Jeong Im Kim, my sons, Jason and Justin, parents and numerous friends. Without their love and encouragement, this thesis would not have come into existence.

TABLE OF CONTENTS

DEDICATION.....	ii
ACKNOWLEDGMENTS	iii
LIST OF FIGURES	vi
LIST OF TABLES	viii
LIST OF APPENDICES	ix
ABSTRACT.....	x

CHAPTER

1 Introduction.....	1
1.1 Background.....	1
1.2 Motivation.....	2
1.3 Importance of pulsed source problem.....	3
1.4 Key ideas.....	4
1.5 Outline of dissertation.....	4
1.6 References.....	6
2 Angular Discretization.....	7
2.1 Spherical harmonics method	7
2.2 P_N approximation to the transport equation	9
2.3 Discrete Ordinates method	11
2.4 Diffusion theory	12
2.5 Eddington approximation.....	14
2.6 References.....	16
3 Spatial and Temporal Discretization	17
3.1 Introduction.....	17
3.2 Hyperbolicity of Quasilinear and Nonlinear systems	17
3.3 Finite Volume Method for hyperbolic system.....	18
3.4 Roe type Riemann solver for cell interface flux calculation.....	20

3.5 High resolution scheme.....	23
3.6 Time discretization.....	25
3.7 Central difference scheme for the second order derivatives.....	26
3.8 Boundary conditions.....	27
3.9 References.....	29
4 P_3 with Quasi-Static Model	30
4.1 Introduction.....	30
4.2 P_3 equations for time-dependent transport.....	30
4.3 Quasi-Static Closure	31
4.4 Numerical results in 1D	33
4.5 Numerical results in 2D	37
4.6 References.....	43
5 Nonlinear P_1 Model.....	44
5.1 Introduction.....	44
5.2 Eddington tensor in 3D and Eddington factor	44
5.3 Properties of Eddington Factor Models	47
5.4 Construction of Nonlinear P_1 Eddington factor	54
5.5 Comparisons of various Eddington factors	56
5.6 Riemann Solver Construction for the Nonlinear P_1 closure.....	59
5.7 Numerical results in 1D	64
5.8 Numerical results in 2D	67
5.9 References.....	78
6 Conclusions.....	79
6.1 Our Goal.....	79
6.2 Summary	79
6.3 Future work.....	80
APPENDICES	82

LIST OF FIGURES

Figure

1.1 Schematic relations of transport solution methods	2
3.1 Reconstruction scheme with harmonic mean limiter	23
3.2 A typical finite difference grid.....	26
3.3 The left corner of a typical computational domain	28
4.1 Comparison of solution at 1 second after the pulse in 1D	34
4.2 ψ_2^0 of P_3QS solution as a function of time and space	34
4.3 Comparison of solution at 5 second after the pulse in 1D	36
4.4 Comparison of solution at 10 second after the pulse in 1D	36
4.5 Comparison of solution at 1 second after the pulse in 2D	38
4.6 Comparison of solution at 5 second after the pulse in 2D	38
4.7 Comparison of solution at 10 second after the pulse in 2D	39
4.8 Negative fluxes for P_3QS and P_1 solutions at 0.5 second after the pulse in 2D	40
4.9 Comparison of the volume integrals of negative fluxes for P_3QS and P_1	41
4.10 P_3QS solution for various scattering ratios at 1 second in 1D	42
4.11 P_3QS solution for various scattering ratios at 1 second in 2D	42
5.1 Plot of χ/η as a function of η	49
5.2 Plot of relaxation length for transport, P_1 and ME closures	51
5.3 Plot of the eigenvalues of the Jacobian.....	53
5.4 Comparison of χ/η as a function of η for various Eddington factors	57

5.5 Comparison of relaxation length as a function of η for various Eddington factors.....	58
5.6 Comparison of eigenvalues for various Eddington factors.....	59
5.7 Comparison of Transport, P_1 , ME, Kershaw and NLP_1 at 1 sec in 1D.....	64
5.8 Comparison of Transport, P_1 , ME, Kershaw and NLP_1 at 5 sec in 1D.....	65
5.9 Comparison of Transport, P_1 , ME, Kershaw and NLP_1 at 10 sec in 1D.....	66
5.10 Plot of difference from Transport at 10 sec in 1D	66
5.11 Comparison of P_1 , Kershaw and NLP_1 at 0.2 sec after the pulse in 2D	68
5.12 Comparison of P_1 , Kershaw and NLP_1 at 0.5 sec after the pulse in 2D	68
5.13 Comparison of P_1 , Kershaw and NLP_1 at 1 sec after the pulse in 2D.....	69
5.14 Comparison of P_1 and NLP_1 at 1 sec after the pulse in 2D	70
5.15 Comparison of P_1 , Kershaw and NLP_1 at 3 sec after the pulse in 2D.....	71
5.16 Comparison of transport, P_1 , Kershaw and NLP_1 at 5 sec after the pulse in 2D.....	71
5.17 NLP_1 solutions for various scattering ratios at 1 second in 2D	72
5.18 Problem of radiation propagation across a medium.	74
5.19 Radiation propagation for NLP_1 and P_1 across vacuum.....	75
5.20 Line-out ($z = 0$) for NLP_1 and P_1 across vacuum.....	75
5.21 Radiation propagation for NLP_1 and P_1 across pure absorber ($c = 0$)	76
5.22 Radiation propagation for NLP_1 and P_1 across pure absorber ($c = 0.5$)	76
5.23 Radiation propagation for NLP_1 and P_1 across pure absorber ($c = 1$).....	77
5.24 Radiation propagation for NLP_1 and P_1 with strong absorber in the center	77

LIST OF TABLES

Table

4.1 Comparison of the number of unknowns for P_N vs. P_3QS	39
5.1 Volume integral of the difference of the scalar flux from transport	67

LIST OF APPENDICES

APPENDIX

A: CFL Condition	83
B: Roe matrix for a two moment Eddington type closure	86
C: 2D P_3QS Riemann Solver (coded by Kyeong)	87
D: 2D Nonlinear P_1 Riemann Solver (coded by Kyeong).....	92

ABSTRACT

The time-dependent radiation transport has been solved using the spherical harmonics method. One problem with this method is that the scalar flux computed with the time-dependent spherical harmonics in 2D can become negative for rapid transients. In this dissertation, two schemes are suggested to overcome this problem. One relies on relaxing hyperbolicity of the spherical harmonics closure by using a quasi-static approximation and the other relies on removing linearity by making the P_1 equation nonlinear.

A quasi-static approximation is normally made in P_1 theory to derive time-dependent diffusion theory, and this approximation maintains a positive flux. We have therefore explored such a quasi-static closure in P_3 in 2D to see if a positive flux can result. The resulting method is relatively accurate, producing better results than much higher order P_N methods, even while having far fewer unknowns. However the numerical results show that the scalar flux can still go negative in a pulsed source problem in 2D and this method is still not robust.

Next, using the eigenstructure of the variable Eddington factor equations to satisfy both the isotropic and beam-like limits we have derived a new Eddington factor as the simplest explicit polynomial of current-to-flux ratio. We have used the discontinuity structure and relaxation length as a way to compare the nonlinear Eddington factors. We have found that the existence of a minimum in the relaxation length implies the existence of discontinuity solution for nonlinear Eddington closures. The numerical results show that our Nonlinear P_1 closure produces very similar results with Maximum Entropy closure in 1D and remains positive for rapid transients in 2D. This method therefore provides a robust P_1 method in 1 and 2D.

CHAPTER 1

Introduction

1.1 Background

To determine the distribution of particles in a system we must investigate the process of radiation transport, that is, the motion of the particles including streaming, scattering, absorption and leakage out of the system [1]. This can be done by solving the radiation transport equation, often called the Boltzmann transport equation, first formulated for the study of the kinetic theory of gases [2]. Time-dependent transport can be described by the Boltzmann transport equation in an arbitrary volume \mathbf{R} :

$$\frac{1}{v} \frac{\partial}{\partial t} \psi(\mathbf{r}, \mathbf{\Omega}, t) + \mathbf{\Omega} \cdot \nabla \psi(\mathbf{r}, \mathbf{\Omega}, t) + \Sigma_t \psi(\mathbf{r}, \mathbf{\Omega}, t) = \frac{\Sigma_s}{4\pi} \int_{4\pi} \psi(\mathbf{r}, \mathbf{\Omega}', t) d\mathbf{\Omega}' + Q(\mathbf{r}, \mathbf{\Omega}, t)$$
$$\mathbf{r} \in \mathbf{R}, \quad |\mathbf{\Omega}| = 1, \quad t > 0 \quad (1.1)$$

where ψ is the angular flux, v is a particle speed, $\mathbf{\Omega}$ is the angular variable, Σ_t, Σ_s are the total and scattering cross section respectively, and Q is an independent or external source. The angular flux ψ is a function of seven variables: three position variables (\mathbf{r}), two angular variables ($\mathbf{\Omega}$), a time variable (t) and the particle speed (v). In this thesis, we suppose for convenience that all the particles in the system have the same speed $v = 1 \text{ cm/sec}$. The cross sections are assumed constant in space and time and scattering is isotropic in the lab frame. The derivation of the transport equation can be found in textbooks on transport theory [2, 3].

Figure 1.1 illustrates various solution techniques for transport problem [4]. There are two primary approaches for solving the transport equation, namely *stochastic* and

deterministic. One deterministic method used in solving transport problem is the spherical harmonics (P_N) method and it will be the subject of this dissertation.

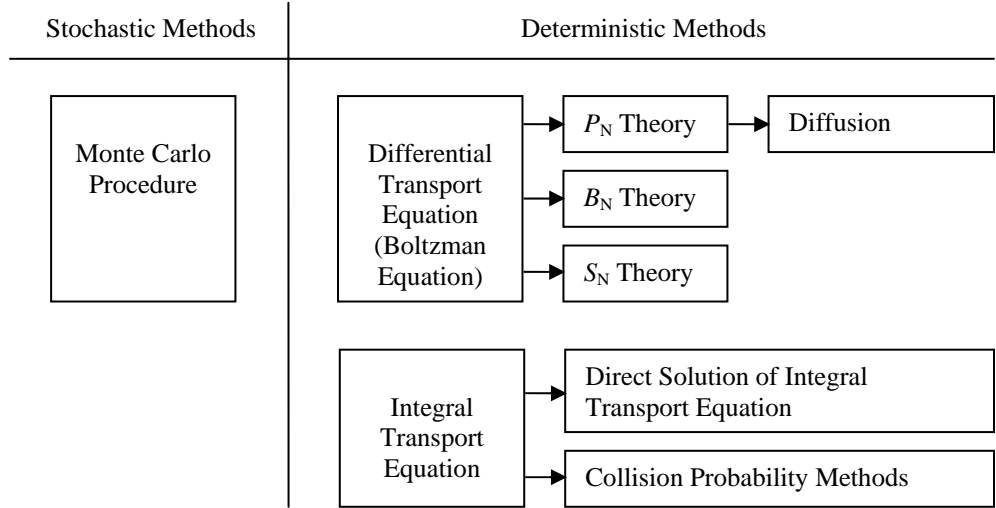


Figure 1.1 Schematic relations of transport solution methods

1.2 Motivation

The time-dependent transport equation has previously been solved in 2D using spherical harmonic expansions [5, 6, 7]. One interesting result of this work was the realization that the scalar flux in the time-dependent spherical harmonics equations in 2 and 3D can become negative in short time transients. The negative flux indicates that the approximation can be unphysical during short time transients. This has recently [8] been proven to be a consequence of any model that is linear, hyperbolic, rotationally invariant, and has finite angular resolution. There is therefore some interest in relaxing each of these conditions to see if a model can be developed for time-dependent transport based on low order angular information that nevertheless maintains a positive scalar flux in short time transients. This will be the focus of this dissertation.

Time dependent radiation transport is important to understanding and modeling a number of physical systems. Pulsed neutron sources, such as the Spallation Neutron Source at Oak Ridge National Laboratory produce a pulsed source of neutrons that have been modeled using time-dependent Monte Carlo. High energy density pulsed power machines such as the Z-machine at Sandia produce pulses of radiation that then couple to hydrodynamic phenomena. These are most routinely modeled using diffusion theory. This dissertation will contribute to the basic modeling tools for such systems by exploring some issues in building robust low-order-in-angle time-dependent radiation transport models.

1.3 Importance of Pulsed Source Problem

In many time-dependent transport problems, we need to solve a differential equation of the form

$$\mathbf{M}\phi(x,t) = \mathbf{S}(x,t) \quad (1.2)$$

where \mathbf{M} is a differential operator and \mathbf{S} is a source term. One well-known technique available to solve this kind of problem is the Green's function method. In this method the solution of Eq. (1.2) is constructed using the Green's function G that satisfies

$$\mathbf{M}G(x,x',t,t') = \delta(x-x',t-t'). \quad (1.3)$$

Eq. (1.3) can be understood physically as a problem driven by a pulsed source on the right hand side. The general solution to Eq. (1.3) is just

$$\phi(x,t) = \int dx' \int dt' G(x,x',t,t') \mathbf{S}(x',t'). \quad (1.4)$$

Proof:
$$\mathbf{M}\phi(x,t) = \int dx' \int dt' \underbrace{\mathbf{M}G(x,x',t,t')}_{\delta(x-x',t-t')} \mathbf{S}(x',t') = \mathbf{S}(x,t) \quad (1.5)$$

This implies that if we find a solution of the pulsed source problem we can simply obtain the general solution of a time-dependent transport problem we want to solve by integration. Solutions to all other time-dependent problems are just superposition of such Green's functions.

1.4 Key Ideas

In order to overcome the disadvantage of spherical harmonics in short time transients, two schemes are suggested and implemented in this dissertation. One is relaxing hyperbolicity by using a quasi-static approximation. The other one is relaxing linearity by using a new nonlinear Eddington factor. A quasi-static approximation is standard in P_1 to derive time-dependent diffusion theory, and this approximation is known to maintain a positive flux. Moreover, the Simplified P_N method does not provide a positive scalar flux even in 1D, but a quasi-static approximation to it does [9]. Therefore in this dissertation we explore a quasi-static approximation to P_3 theory; we call this new method P_3QS . Linearity has been relaxed in the Maximum Entropy closure by Brunner and Holloway [10] in 1D but it has not been tried for more than 1D because of the computational difficulties of the Maximum Entropy closure in 2D. Therefore we develop an approximation to the Maximum Entropy closure that is computationally tractable; we call this new method NLP_1 .

1.5 Outline of Dissertation

This dissertation begins with an introduction to the spherical harmonics method and various angular discretization methods in Chapter 2. Chapter 3 presents the spatial and time discretization method. In Chapter 4 we propose the P_3 with Quasi-Static Method (P_3QS) in which hyperbolicity has been given up and implement it for 1D and 2D for the pulsed source problem. In Chapter 5 we discuss various properties of Eddington factor methods and construct a new nonlinear Eddington factor model in which the property of linearity of regular P_1 has been given up. We then implement and explore this model in

1D and 2D for pulsed source problems. Conclusions and directions for future work are described in Chapter 6.

1.6 References

- [1] J. J. Duderstadt and L. J. Hamilton, *Nuclear Reactor Analysis*, Wiley & Sons, pp. 103, 1976.
- [2] E. E. Lewis, *Computational methods of Neutron Transport*, Wiley & Sons, pp. 1, 1984.
- [3] G. I. Bell and S. Glasstone, *Nuclear Reactor theory*, Kieger, 1970.
- [4] K. O. Ott, *Introductory Nuclear Reactor Statics*, American Nuclear Society, pp. 296, 1989.
- [5] T. A. Brunner, J. P. Holloway, "Two-dimensional time dependent Riemann solver for neutron transport," *Journal of Computational Physics*, vol. 210, pp. 386-399 2005.
- [6] R. G. McClarren, *Spherical Harmonics Methods for Thermal Radiation Transport*, Ph.D. Thesis, University of Michigan, Ann Arbor, Michigan, 2007.
- [7] R. G. McClarren, J. P. Holloway, T. A. Brunner, T. A. Mehlhorn, "A quasi-linear implicit Riemann solver for the time-dependent P_N equations," *Nuclear Science and Engineering*, vol. 155, pp. 290-299, 2007.
- [8] R. G. McClarren, J. P. Holloway, T. A. Brunner, "On Solutions to the P_N equations for thermal radiative transfer," *Journal of Computational Physics*, vol. 227, 2008.
- [9] M. Frank, E. W. Larsen, "Time-dependent simplified P_N approximation to the equations of radiative transfer" *Journal of Computational Physics*, vol. 226, pp. 386-399, 2005.
- [10] T. A. Brunner, *Riemann solvers for time-dependent transport based on the maximum entropy and spherical harmonics closures*, University of Michigan, Ph.D. thesis, pp. 39, 2000.

CHAPTER 2

Angular Discretization

2.1 Spherical Harmonics (P_N) Method

The term spherical harmonics was first introduced by Kelvin [1] in studying the theory of potential, which had been developed by Laplace, Legendre and others in the eighteenth century. The P_N approach in solving Eq. (1.1) consists of expanding the angular flux using the “spherical harmonic functions.”[2] If only one angle is required for the angular description, as is the case for one-dimensional problems in plane geometry, then the spherical harmonics become the Legendre Polynomials of the cosine of one angle $\mu = \cos(\theta)$. If two angles are required for the angular description, as is generally the case, the complete orthogonal expansion system of the angular dependence consists of the spherical harmonic functions $Y_l^m(\Omega)$.

The spherical harmonic function, $Y_l^m(\Omega)$ is defined by [3]

$$Y_l^m(\Omega) = \left[\frac{2l+1}{4\pi} \frac{(l-m)!}{(l+m)!} \right]^{1/2} P_l^m(\mu) e^{im\phi} \quad (2.2)$$

where $P_l^m(\mu)$ is the associated Legendre function

$$P_l^m(\mu) = (1-\mu^2)^{m/2} \left(\frac{d}{d\mu} \right)^m P_l(\mu) = P_l^{-m}(\mu). \quad (2.3)$$

From Eq. (2.2) it can be seen that

$$Y_l^{-m}(\boldsymbol{\Omega}) = (-1)^m \overline{Y_l^m}(\boldsymbol{\Omega}) \quad (2.4)$$

where $\overline{Y_l^m}(\boldsymbol{\Omega})$ is the complex conjugate of $Y_l^m(\boldsymbol{\Omega})$.

The spherical harmonic functions are orthogonal; they satisfy

$$\int_{4\pi} Y_l^m(\boldsymbol{\Omega}) \overline{Y_j^k}(\boldsymbol{\Omega}) d\boldsymbol{\Omega} = \delta_{l,j} \delta_{m,k} . \quad (2.5)$$

Any function $f_l^m(\boldsymbol{\Omega})$ on the unit sphere can be expanded in the series with spherical harmonic function as

$$f_l^m(\boldsymbol{\Omega}) = \sum_{l=0}^{\infty} \sum_{m=-l}^l f_l^m Y_l^m(\boldsymbol{\Omega}) \quad (2.6)$$

By using Eq. (2.5), Eq. (2.6) requires,

$$f_l^m = \int_{4\pi} f(\boldsymbol{\Omega}) \overline{Y_l^m}(\boldsymbol{\Omega}) d\boldsymbol{\Omega} . \quad (2.7)$$

Likewise, we can expand the spherical harmonic expansion of the angular flux $\psi(\mathbf{r}, \boldsymbol{\Omega}, t)$ in the transport equation as,

$$\psi(\mathbf{r}, \boldsymbol{\Omega}, t) = \sum_{l=0}^{\infty} \sum_{m=-l}^l \psi_l^m(\mathbf{r}, t) Y_l^m(\boldsymbol{\Omega}) \quad (2.8)$$

with

$$\psi_l^m(\mathbf{r}, t) = \int_{4\pi} \psi(\mathbf{r}, \boldsymbol{\Omega}, t) \overline{Y_l^m}(\boldsymbol{\Omega}) d\boldsymbol{\Omega} . \quad (2.9)$$

where the N is called the order of spherical harmonics. As the N increases, the solutions will become more accurate and further as N goes to infinity, the solution of spherical harmonic expansion converges to the exact solution.

2.2 P_N Approximation to the Transport Equation

Suppose there is no independent source Q and then Eq. (1.1) becomes,

$$\frac{\partial}{\partial t} \psi(\mathbf{r}, \boldsymbol{\Omega}, t) + \boldsymbol{\Omega} \cdot \nabla \psi(\mathbf{r}, \boldsymbol{\Omega}, t) + \Sigma_t \psi(\mathbf{r}, \boldsymbol{\Omega}, t) = \int_{4\pi} \frac{\Sigma_s}{4\pi} \psi(\mathbf{r}, \boldsymbol{\Omega}, t) d\boldsymbol{\Omega}. \quad (2.10)$$

By integrating Eq. (2.10) with $\int_{4\pi} (\cdot) \bar{Y}_l^m(\boldsymbol{\Omega}) d\boldsymbol{\Omega}$ we get,

$$\frac{\partial}{\partial t} \psi_l^m(\mathbf{r}, t) + \int_{4\pi} \boldsymbol{\Omega} \cdot \nabla \psi(\mathbf{r}, \boldsymbol{\Omega}, t) \bar{Y}_l^m(\boldsymbol{\Omega}) d\boldsymbol{\Omega} + \Sigma_t \psi_l^m(\mathbf{r}, t) = \Sigma_s \psi_l^m(\mathbf{r}, t) \delta_{l0} \delta_{m0} \quad (2.11)$$

In Eq. (2.11), integrating the streaming term $\boldsymbol{\Omega} \cdot \nabla \psi$ is very complicated and we will simply report Brunner's result [4]

$$\begin{aligned} \int_{4\pi} \boldsymbol{\Omega} \cdot \nabla \psi(\mathbf{r}, \boldsymbol{\Omega}, t) \bar{Y}_l^m(\boldsymbol{\Omega}) d\boldsymbol{\Omega} = & + \frac{1}{2} \frac{\partial}{\partial x} (-C_{l-1}^{m-1} \psi_{l-1}^{m-1} + D_{l+1}^{m-1} \psi_{l+1}^{m-1} + E_{l-1}^{m+1} \psi_{l-1}^{m+1} - F_{l+1}^{m+1} \psi_{l+1}^{m+1}) \\ & + \frac{1}{2} i \frac{\partial}{\partial y} (C_{l-1}^{m-1} \psi_{l-1}^{m-1} - D_{l+1}^{m-1} \psi_{l+1}^{m-1} + E_{l-1}^{m+1} \psi_{l-1}^{m+1} - F_{l+1}^{m+1} \psi_{l+1}^{m+1}) \quad (2.12) \\ & + \frac{\partial}{\partial z} (A_{l-1}^m \psi_{l-1}^m + B_{l+1}^m \psi_{l+1}^m) \end{aligned}$$

where,

$$\begin{aligned} A_l^m &= \sqrt{\frac{(l-m+1)(l+m+1)}{(2l+3)(2l+1)}}, & B_l^m &= \sqrt{\frac{(l-m)(l+m)}{(2l+1)(2l-1)}} \\ C_l^m &= \sqrt{\frac{(l+m+1)(l+m+2)}{(2l+3)(2l+1)}}, & D_l^m &= \sqrt{\frac{(l-m)(l+m-1)}{(2l+1)(2l-1)}} \\ E_l^m &= \sqrt{\frac{(l-m+1)(l-m+2)}{(2l+3)(2l+1)}}, & F_l^m &= \sqrt{\frac{(l+m)(l+m-1)}{(2l+1)(2l-1)}}. \end{aligned} \quad (2.13)$$

Substituting Eq. (2.12) into Eq. (2.11) yields the 3D P_N equations

$$\begin{aligned}
& \frac{\partial}{\partial t} \psi_l^m + \frac{1}{2} \frac{\partial}{\partial x} (-C_{l-1}^{m-1} \psi_{l-1}^{m-1} + D_{l+1}^{m-1} \psi_{l+1}^{m-1} + E_{l-1}^{m+1} \psi_{l-1}^{m+1} - F_{l+1}^{m+1} \psi_{l+1}^{m+1}) \\
& + \frac{i}{2} \frac{\partial}{\partial y} (C_{l-1}^{m-1} \psi_{l-1}^{m-1} - D_{l+1}^{m-1} \psi_{l+1}^{m-1} + E_{l-1}^{m+1} \psi_{l-1}^{m+1} - F_{l+1}^{m+1} \psi_{l+1}^{m+1}) \\
& + \frac{\partial}{\partial z} (A_{l-1}^m \psi_{l-1}^m + B_{l+1}^m \psi_{l+1}^m) + \Sigma_t \psi_l^m = \Sigma_s \psi_0^0 \delta_{l0} \delta_{m0} .
\end{aligned} \tag{2.14}$$

We obtain 2D P_N equations by dropping y - dependence in Eq. (2.14), yielding

$$\begin{aligned}
& \frac{\partial}{\partial t} \psi_l^m + \frac{1}{2} \frac{\partial}{\partial x} (-C_{l-1}^{m-1} \psi_{l-1}^{m-1} + D_{l+1}^{m-1} \psi_{l+1}^{m-1} + E_{l-1}^{m+1} \psi_{l-1}^{m+1} - F_{l+1}^{m+1} \psi_{l+1}^{m+1}) \\
& + \frac{\partial}{\partial z} (A_{l-1}^m \psi_{l-1}^m + B_{l+1}^m \psi_{l+1}^m) + \Sigma_t \psi_l^m = \Sigma_s \psi_0^0 \delta_{l0} \delta_{m0} .
\end{aligned} \tag{2.15}$$

Further dropping the x -dependence in Eq.(2.15) provides the 1D P_N equations as

$$\frac{\partial}{\partial t} \psi_l^m + \frac{\partial}{\partial z} (A_{l-1}^m \psi_{l-1}^m + B_{l+1}^m \psi_{l+1}^m) + \Sigma_t \psi_l^m = \Sigma_s \psi_0^0 \delta_{l0} \delta_{m0} . \tag{2.16}$$

Since l is defined as $l = 0, 1, 2, \dots, N$, the P_N equations can be of odd-order or even-order. Unfortunately, solutions of even-order P_N equations have an undesirable property: at a material interface between two media - where the cross sections are discontinuous - the even order P_N solutions are incorrectly discontinuous [5]. However, the odd-order P_N equations are continuous at a material interface - which is physically correct. For this reason, odd-order P_N approximations are used in practice, and even-order P_N approximations are typically not used.

2.3 Discrete Ordinates (S_N) Method

In the P_N equations, the angular flux is approximated as a finite sum of spherical harmonic functions and the transport equations are developed for the space and time dependent expansion coefficients. In the Discrete Ordinates Method, a finite set of discrete ordinate directions is chosen and each discrete ordinate has an associated angular weight on the unit sphere.

$$\int_{4\pi} \psi(\mathbf{\Omega}) d\mathbf{\Omega} \approx \sum_{n=1}^N \psi_n w_n \quad (2.17)$$

where ψ_n represents $\psi(\mathbf{\Omega}_n)$ and $\mathbf{\Omega}_n$ is a discrete ordinate and w_n is an angular weight. The set of $\mathbf{\Omega}_n$ and w_n together are termed a quadrature set. The discrete ordinates approximation to the transport equation is obtained as

$$\frac{\partial}{\partial t} \psi_n(\mathbf{r}, t) + \mathbf{\Omega}_n \cdot \nabla \psi_n(\mathbf{r}, t) + \Sigma_t \psi_n(\mathbf{r}, t) = \frac{\Sigma_s}{4\pi} \sum_{n=1}^N \psi_n w_n + Q_n(\mathbf{r}, t) \quad (2.18)$$

The discrete ordinates method is a widely used method of transport analysis. However, in multidimensional calculations with optically thin regions and highly localized sources, the S_N solutions can exhibit a nonphysical oscillation called ray effects [6, 7]. These effects arise because the S_N equations allow propagation of the solution only along a discrete set of directions rather than the continuum of directions [8]. The S_N equations lack rotational invariance. The ray effects may be reduced by increasing the order N of the S_N calculation. However this may be expensive for the computational effort. While a higher order S_N calculation will reduce ray effects, this will not eliminate them altogether. Limitation of ray effects can only be accomplished by a method that has rotational invariance, such as the Spherical Harmonics Method [9].

2.4 Diffusion Theory

If we define the first two moments of angular flux, the scalar flux ϕ and the current \mathbf{J} , as

$$\phi(\mathbf{r}, t) = \int_{4\pi} \psi(\mathbf{r}, \boldsymbol{\Omega}, t) d\boldsymbol{\Omega} \quad (2.19)$$

$$\mathbf{J}(\mathbf{r}, t) = \int_{4\pi} \boldsymbol{\Omega} \psi(\mathbf{r}, \boldsymbol{\Omega}, t) d\boldsymbol{\Omega} \quad (2.20)$$

and then operate on the transport equation by $\int_{4\pi} (\cdot) d\boldsymbol{\Omega}$ and $\int_{4\pi} \boldsymbol{\Omega} (\cdot) d\boldsymbol{\Omega}$ we obtain two moment equations

$$\frac{\partial}{\partial t} \phi + \nabla \cdot \mathbf{J} + \Sigma_t \phi = \Sigma_s \phi + Q_0 \quad (2.21)$$

$$\frac{\partial}{\partial t} \mathbf{J} + \nabla \cdot \int_{4\pi} \boldsymbol{\Omega} \boldsymbol{\Omega} \psi(\boldsymbol{\Omega}) d\boldsymbol{\Omega} + \Sigma_t \mathbf{J} = \mathbf{Q}_1. \quad (2.22)$$

These equations are still exact without any approximations. The diffusion approximation is a consequence of the following different approximations [10]:

- D1.** The angular flux is represented by only a linearly anisotropic angular dependence,
- D2.** Isotropic sources,
- D3.** The neutron current density changes slowly on a time scale compared to the mean collision time.

Approximation **D1** in the P_1 approximation implies that the angular flux is represented as

$$\psi(\mathbf{r}, \boldsymbol{\Omega}, t) \cong \frac{1}{4\pi} \phi + \frac{3}{4\pi} \boldsymbol{\Omega} \cdot \mathbf{J}(\mathbf{r}, t) \quad (2.23)$$

Applying Eq. (2.23) to the leakage term in the second moment equation Eq. (2.22), we find

$$\int_{4\pi} \boldsymbol{\Omega} \boldsymbol{\Omega} \psi(\mathbf{r}, \boldsymbol{\Omega}, t) d\boldsymbol{\Omega} = \frac{1}{3} \phi(\mathbf{r}, t) \quad (2.24)$$

Also the source term $\mathbf{Q}_1(\mathbf{r}, t)$ in the Eq. (2.22) vanishes by **D2**

$$\mathbf{Q}_1(\mathbf{r}, t) = 0. \quad (2.25)$$

Next, **D3** asserts that we assume that we can neglect the time derivative $\partial \mathbf{J} / \partial t$ in comparison with the remaining terms in Eq. (2.22). This would imply $\partial \mathbf{J} / \partial t \ll \Sigma_t \mathbf{J}$ and so

$$\frac{\partial \mathbf{J}}{\partial t} \approx 0. \quad (2.26)$$

That is, that the rate of time variation of the current density is much slower than the collision frequency. Note that an extremely rapid time variation of the current would invalidate this assumption so the diffusion approximation is not valid for the rapid transients of interest to us. By Eq. (2.23) – (2.26), Eq. (2.22) becomes,

$$\frac{1}{3} \nabla \phi + \Sigma_t \mathbf{J} = 0 \quad (2.27)$$

which gives Fick's law

$$\mathbf{J} = -(1/3\Sigma_t) \nabla \phi. \quad (2.28)$$

Then substituting Eq. (2.28) into Eq. (2.21), gives us the time-dependent diffusion Equation,

$$\frac{\partial}{\partial t} \phi - \nabla \cdot D(\mathbf{r}) \nabla \phi + \Sigma_a \phi = Q_0(\mathbf{r}, t) \quad (2.29)$$

where

$$D(\mathbf{r}) = [3\Sigma_t]^{-1}. \quad (2.30)$$

The time-dependent diffusion theory is commonly used in reactor core simulation. Note however that it is not useful in a vacuum or for rapid transients.

2.5 Eddington Approximation

In order to obtain a closed set of moment equations, it is necessary to relate one or more of the higher angular moments to lower moments. In lowest order, in which only two moment equations are considered, this approximate closure technique leads to the variable Eddington approximation.

Consider again the two moment equations of Eq. (2.21)-(2.22).

$$\frac{\partial}{\partial t} \phi + \nabla \cdot \mathbf{J} + \Sigma_t \phi = \Sigma_s \phi + Q_0 \quad (2.31)$$

$$\frac{\partial}{\partial t} \mathbf{J} + \nabla \cdot \int_{4\pi} \mathbf{\Omega} \mathbf{\Omega} \psi(\mathbf{\Omega}) d\mathbf{\Omega} + \Sigma_t \mathbf{J} = \mathbf{Q}_1. \quad (2.32)$$

To obtain a closed set of moment equations, it is necessary to approximate Eq. (2.32) with an equation which contains only ϕ and \mathbf{J} . The variable Eddington approach is to rewrite Eq. (2.31) and (2.32) as

$$\frac{\partial}{\partial t} \phi + \nabla \cdot \mathbf{J} + \Sigma_t \phi = \Sigma_s \phi + Q_0 \quad (2.33)$$

$$\frac{\partial}{\partial t} \mathbf{J} + \nabla \cdot \boldsymbol{\chi} \phi + \Sigma_t \mathbf{J} = \mathbf{Q}_1 \quad (2.34)$$

where we call $\boldsymbol{\chi}$ the Eddington tensor and it is defined as

$$\boldsymbol{\chi} = \frac{1}{\phi} \int_{4\pi} \mathbf{\Omega} \mathbf{\Omega} \psi(\mathbf{\Omega}) d\mathbf{\Omega}. \quad (2.35)$$

One approach to compute $\boldsymbol{\chi}$ approximately is to postulate a simple model for the $\mathbf{\Omega}$ dependence of the angular flux. As a technique for solving transport problems, the variable Eddington approximation is known as more accurate than the diffusion

approximation and much faster than transport calculations [11]. Several kinds of Eddington factors will be introduced and studied in Chapter 5.

2.7 References

- [1] E. W. Hobson, *The theory of Spherical and Ellipsoidal Harmonics*, Cambridge, pp.1, 1931.
- [2] K. O. Ott, *Introductory Nuclear Reactor Statics*, American Nuclear Society, pp. 297, 1989.
- [3] G. I. Bell and S. Glasstone, “*Nuclear Reactor theory*,” Kieger, 1970.
- [4] T. A. Brunner, *Riemann solvers for time-dependent transport based on the maximum entropy and spherical harmonics closures*, University of Michigan, Ph.D. thesis, pp. 30, 2000.
- [5] E. W. Larsen, *Advanced Nuclear Reactor Theory*, University of Michigan, NERS 543, pp. 83, 2005. (Unpublished)
- [6] E. E. Lewis and W. F. Miller, *Computational Methods of Neutron Transport*, John Wiley and Sons, pp. 194, 1984.
- [7] W. F. Miller and W. H. Reed, “Ray-Effect Mitigation Methods for Two-Dimensional Neutron Transport Theory,” *Nuclear Science and Engineering*, vol. 62, pp. 391, 1977.
- [8] J. E. Morel et al, “Analysis of Ray –Effect Mitigation Techniques,” *Nuclear Science and Engineering*, vol. 144, pp. 1-22, 2003.
- [9] K. D. Lathrop, “Remedies for Ray Effects,” *Nuclear Science and Engineering*, vol. 45, pp. 255, 1971.
- [10] J. J. Duderstadt and L.J. Hamilton, *Nuclear Reactor Analysis*, Wiley & Sons, pp. 103, 1976.
- [11] G. N. Minerbo, “Maximum Entropy Eddington Factors” *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol.20, pp. 541, 1978.

CHAPTER 3

Spatial and Temporal Discretization

3.1 Introduction

For completeness in this chapter, we introduce standard numerical methods along with definitions and analyses used to obtain a solution of our proposed models later in the dissertation. In Section 3.2, we consider the meaning of a hyperbolic system of PDE's. We describe the Finite Volume Method for spatial discretization scheme in Section 3.3. In Section 3.4, we review the Roe type Riemann solver for cell interface flux calculation. We describe in Section 3.5 the high resolution scheme. In Section 3.6 we discuss the time discretization and the stability condition. We present in Section 3.7 the central difference scheme for higher order derivatives. Boundary conditions are described in Section 3.8.

3.2 Hyperbolicity of Quasilinear and Nonlinear Systems

Consider a PDE system of the form

$$\frac{\partial u}{\partial t} + A \frac{\partial u}{\partial x} = 0 \quad (3.1)$$

where $u(x,t)$ is a vector with m components representing the unknown functions (flux, current, pressure, etc.) and $A(u, x, t)$ is a Jacobian matrix that has $m \times m$ real elements. This system is called “hyperbolic” [1] if A is diagonalizable with real eigenvalues so that we can decompose

$$R^{-1}AR = \Lambda \quad \text{and} \quad A = R\Lambda R^{-1} \quad (3.2)$$

where Λ is a diagonal matrix of eigenvalues and R is the matrix of right eigenvectors which is defined with $AR = \Lambda R$. We can rewrite Eq. (3.1) with Eq. (3.2) as

$$R^{-1} \frac{\partial u}{\partial t} + R^{-1} A R R^{-1} \frac{\partial u}{\partial x} = 0 \quad (3.3)$$

If we define $w = R^{-1}u$, then this takes the form

$$\frac{\partial w}{\partial t} + \Lambda \frac{\partial w}{\partial x} = 0 \quad (3.4)$$

Since Λ is diagonal, this system decouples into m independent advection equations for the components w_n of w :

$$\frac{\partial w_n}{\partial t} + \lambda_n \frac{\partial w_n}{\partial x} = 0 \quad \text{for } n=1, 2, \dots, m. \quad (3.5)$$

Since each λ_n is real, these advection equations make sense physically and can be used to solve the original system of equations (3.1). Each decoupled equation has corresponding eigenvalue λ_n which gives the speed at which each solution component propagates. For example, the Jacobian matrix A of the one-dimensional P_1 equation has eigenvalues $\pm 1/\sqrt{3}$; these are the speeds at which particles travel along the axis when moving with direction cosines $\mu = \pm 1/\sqrt{3}$.

3.3 Finite Volume Method for Hyperbolic System

The Finite Volume Method (FVM) is a discretization of the governing equation integrated over a computational cell in a meshed geometry. “Finite volume” refers to the small volume of each mesh cell, centered on a computational node point. Because the flux entering a given volume is identical to that leaving the adjacent volume, these methods are conservative.

Consider a general hyperbolic system

$$\frac{\partial u}{\partial t} + \nabla \cdot F(u) = 0 \quad (3.6)$$

where $u(x, t)$ is a vector and $F(u)$ is the corresponding flux tensor. We sub-divide the spatial domain into finite volumes or cells and then take the volume integral over each cell, which gives

$$\int_{v_i} \frac{\partial u}{\partial t} dv + \int_{v_i} \nabla \cdot F(u) dv = 0 \quad (3.7)$$

If we define cell average value as

$$u_i = \frac{1}{v_i} \int_{v_i} u(x, t) dv \quad (3.8)$$

and convert the divergence term to surface integrals using the divergence theorem [2], the Eq. (3.7) becomes

$$\frac{\partial u_i}{\partial t} + \frac{1}{v_i} \oint_{S_i} F(u) \cdot \mathbf{n} dv = 0 \quad (3.9)$$

where S_i represents the total surface area of the cell and \mathbf{n} is a unit vector normal to the surface and pointing outward. The first term represents the time rate of change of the conserved quantities stored in the volume. The second term, an integral over the surface of the volume, represents the net rate at which the conserved quantities are conducted out through the surface of the volume.

Specializing the FVM for a two-dimensional Cartesian time-dependent conservation law described as

$$\frac{\partial u}{\partial t} + \frac{\partial F(u)}{\partial x} + \frac{\partial F(u)}{\partial z} = 0, \quad (3.10)$$

and operation on Eq. (3.10) with $\frac{1}{\Delta x \Delta z} \int_{x_{i,j-1/2}}^{x_{i,j+1/2}} \int_{z_{i-1/2,j}}^{z_{i+1/2,j}} (\cdot) dz dx$ then we get the finite volume form

$$\frac{\partial u_{i,j}}{\partial t} + \frac{1}{\Delta x} (F_{i+1/2,j} - F_{i-1/2,j}) + \frac{1}{\Delta z} (F_{i,j+1/2} - F_{i,j-1/2}) = 0 \quad (3.11)$$

where

$$u_{i,j} = \frac{1}{\Delta x \Delta z} \int_{z_{i,j-1/2}}^{z_{i,j+1/2}} \int_{x_{i-1/2,j}}^{x_{i+1/2,j}} u(x, z, t) dz dx. \quad (3.12)$$

The FVM is concise if only first order derivatives are present [3] like in the advection equations. So if second or higher order derivatives occur in the equations then additional manipulation will be required to discretize it.

3.4 Roe type Riemann solver for cell interface flux calculation

The Riemann problem consists of a conservation law together with piecewise constant initial data having a single discontinuity. It appears in a natural way in finite volume methods for the solution of systems of conservation laws due to the discreteness. Typically the Riemann problem requires a solution of a nonlinear system. A significant gain in efficiency can be realized if a solution to a linear problem approximating the original nonlinear Riemann problem can be obtained. This is the basis for Roe's scheme.

Consider a Riemann problem for a one-dimensional hyperbolic system of conservation laws

$$\frac{\partial u}{\partial t} + \frac{\partial F(u)}{\partial x} = 0 \quad (3.13)$$

with the initial values

$$u(x, 0) = \begin{cases} u_L & x < 0, \\ u_R & x > 0, \end{cases} \quad (3.14)$$

where u_L and u_R are the left and right state vectors.

Roe solved the Riemann problem Eq. (3.13) – (3.14) approximately by introducing the Jacobian matrix

$$A(u) = \frac{\partial F(u)}{\partial u} \quad (3.15)$$

and using the chain rule in Eq. (3.13) so that it may be written as

$$\frac{\partial u}{\partial t} + A \frac{\partial u}{\partial x} = 0. \quad (3.16)$$

Roe's approach replaces the Jacobian matrix A in Eq. (3.16) by a constant Jacobian matrix

$$R = R(u_L, u_R) \quad (3.17)$$

The components of R are evaluated using u_L and u_R . In this way the original Riemann problem Eq. (3.13) – (3.14) is then replaced by the approximate linear problem

$$\frac{\partial u}{\partial t} + R \frac{\partial u}{\partial x} = 0 \quad (3.18)$$

with the initial values

$$u(x, 0) = \begin{cases} u_L & x < 0, \\ u_R & x > 0, \end{cases} \quad (3.19)$$

where the initial conditions are the same as those in the original problem and R is called the Roe matrix. The Roe matrix is chosen to satisfy certain conditions, so that a solution of the linear problem becomes an approximate solution of the nonlinear Riemann problem. These conditions include the following properties [5]:

R1. $R(u_R, u_L)$ has a complete set of real eigenvalues and eigenvectors.

R2. As $u_R, u_L \rightarrow u$, $R(u_R, u_L) = A(u)$

R3. $R(u_R, u_L) \cdot (u_R - u_L) = F(u_R) - F(u_L)$

Property **R1** ensures hyperbolicity. Property **R2** guarantees that the approximate solution tends to the exact solution when u_L and u_R approach each other. Property **R3** ensures the conservation across discontinuities. Based on these properties, Roe's numerical flux can be written as [1, 4, 6, 7, 8].

$$F(u_L, u_R) = \frac{1}{2} [F(u_L) + F(u_R)] - \frac{1}{2} |R| (u_R - u_L) \quad (3.20)$$

with

$$|R| = \sum_k \mathbf{r}_k |\lambda_k| \mathbf{l}_k \quad (3.21)$$

where $\mathbf{r}_k, \mathbf{l}_k$ and λ_k are the k -th right and left eigenvectors and eigenvalues of R , (k runs from 1 to the number of equations of the system). The first term of $F(u_L, u_R)$ in Eq. (3.20) leads to central differencing and the second term provides upwind bias, which introduces numerical dissipation and stability.

Now using Eq. (3.20), the fluxes at each face of a cell evaluate to

$$F_{i+1/2, j} = \frac{1}{2} [F(u_{i+1, j}) + F(u_{i, j})] - \frac{1}{2} |R(u_{i+1, j}, u_{i, j})| (u_{i+1, j} - u_{i, j}) \quad (3.22)$$

$$F_{i-1/2, j} = \frac{1}{2} [F(u_{i, j}) + F(u_{i-1, j})] - \frac{1}{2} |R(u_{i, j}, u_{i-1, j})| (u_{i, j} - u_{i-1, j}) \quad (3.23)$$

$$F_{i, j+1/2} = \frac{1}{2} [F(u_{i, j+1}) + F(u_{i, j})] - \frac{1}{2} |R(u_{i, j+1}, u_{i, j})| (u_{i, j+1} - u_{i, j}) \quad (3.24)$$

$$F_{i,j-1/2} = \frac{1}{2} [F(u_{i,j}) + F(u_{i,j-1})] - \frac{1}{2} |R(u_{i,j}, u_{i,j-1})| (u_{i,j} - u_{i,j-1}) \quad (3.25)$$

Using these cell interface fluxes we can write Eq. (3.11) as

$$\begin{aligned} \frac{\partial u_{i,j}}{\partial t} + \frac{1}{2\Delta x} \{ & (F(u_{i+1,j}) + F(u_{i,j})) - (F(u_{i,j}) + F(u_{i-1,j})) \} \\ & - \frac{1}{2\Delta x} \{ |R(u_{i+1,j}, u_{i,j})| (u_{i+1,j} - u_{i,j}) - |R(u_{i,j}, u_{i-1,j})| (u_{i,j} - u_{i-1,j}) \} \\ & + \frac{1}{2\Delta z} \{ (F(u_{i,j+1}) + F(u_{i,j})) - (F(u_{i,j}) + F(u_{i,j-1})) \} \\ & - \frac{1}{2\Delta z} \{ |R(u_{i,j+1}, u_{i,j})| (u_{i,j+1} - u_{i,j}) - |R(u_{i,j}, u_{i,j-1})| (u_{i,j} - u_{i,j-1}) \} = 0 \end{aligned} \quad (3.26)$$

3.5 High Resolution scheme

The Roe approximate Riemann solver presented in Eq. (3.20) is a first order method in space. In order to make it a better than first order, we need to reconstruct a slope inside each cell using a nonlinear method.

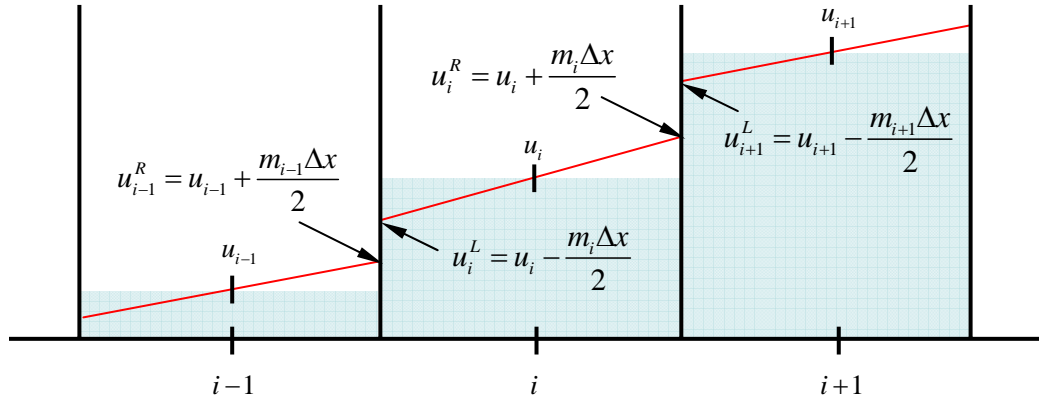


Figure 3.1 Reconstruction scheme with harmonic mean limiter

In this dissertation, we will reconstruct the slope with the harmonic mean limiter proposed by van Leer [9], which is reconstruction of the solution by expressing the slope with nonlinear averaging as

$$m_i = \frac{2m_i^+ m_i^-}{m_i^+ + m_i^-} \quad \text{where} \quad m_i^- = \frac{u_i - u_{i-1}}{\Delta x}, \quad m_i^+ = \frac{u_{i+1} - u_i}{\Delta x} \quad (3.27)$$

where m_i^- and m_i^+ are the slopes to the left and the right of cell i . This slope is constructed component by component in the solution vector u . Having a different sign for the slopes m_i^- and m_i^+ implies the cell i contains a local extreme point, and so m_i is set to 0 in such cases. With this, four boundary vectors can be reconstructed as

$$u_i^R = u_i + \frac{m_i \Delta x}{2} \quad (3.28)$$

$$u_{i+1}^L = u_{i+1} - \frac{m_{i+1} \Delta x}{2} \quad (3.29)$$

$$u_{i-1}^R = u_{i-1} + \frac{m_{i-1} \Delta x}{2} \quad (3.30)$$

$$u_i^L = u_i - \frac{m_i \Delta x}{2} \quad (3.31)$$

Using Eq. (3.28)-(3.31), the fluxes at each face of cell in Eq. (3.22)-(3.25) are rewritten for the high resolution scheme as

$$F_{i+1/2,j} = \frac{1}{2} [F(u_{i+1,j}^L) + F(u_{i,j}^R)] - \frac{1}{2} |R(u_{i+1,j}^L, u_{i,j}^R)| (u_{i+1,j}^L - u_{i,j}^R) \quad (3.32)$$

$$F_{i-1/2,j} = \frac{1}{2} [F(u_{i,j}^L) + F(u_{i-1,j}^R)] - \frac{1}{2} |R(u_{i,j}^L, u_{i-1,j}^R)| (u_{i,j}^L - u_{i-1,j}^R) \quad (3.33)$$

$$F_{i,j+1/2} = \frac{1}{2} [F(u_{i,j+1}^L) + F(u_{i,j}^R)] - \frac{1}{2} |R(u_{i,j+1}^L, u_{i,j}^R)| (u_{i,j+1}^L - u_{i,j}^R) \quad (3.34)$$

$$F_{i,j-1/2} = \frac{1}{2} [F(u_{i,j}^L) + F(u_{i,j-1}^R)] - \frac{1}{2} |R(u_{i,j}^L, u_{i,j-1}^R)| (u_{i,j}^L - u_{i,j-1}^R) \quad (3.35)$$

and then Eq. (3.11) is reconstructed as

$$\begin{aligned}
\frac{\partial u_{i,j}}{\partial t} + \frac{1}{2\Delta x} \left\{ \left(F(u_{i+1,j}^L) + F(u_{i,j}^R) \right) - \left(F(u_{i,j}^L) + F(u_{i-1,j}^R) \right) \right\} \\
- \frac{1}{2\Delta x} \left\{ \left| R(u_{i+1,j}^L, u_{i,j}^R) \right| (u_{i+1,j}^L - u_{i,j}^R) - \left| R(u_{i,j}^L, u_{i-1,j}^R) \right| (u_{i,j}^L - u_{i-1,j}^R) \right\} \\
+ \frac{1}{2\Delta z} \left\{ \left(F(u_{i,j+1}^L) + F(u_{i,j}^R) \right) - \left(F(u_{i,j}^L) + F(u_{i,j-1}^R) \right) \right\} \\
- \frac{1}{2\Delta z} \left\{ \left| R(u_{i,j+1}^L, u_{i,j}^R) \right| (u_{i,j+1}^L - u_{i,j}^R) - \left| R(u_{i,j}^L, u_{i,j-1}^R) \right| (u_{i,j}^L - u_{i,j-1}^R) \right\} = 0.
\end{aligned} \tag{3.36}$$

Eq. (3.36) is a general form for the 2D High Resolution Roe type Riemann solver which will be intensively used in Chapters 4 and 5.

3.6 Time discretization

In this dissertation, we use an explicit Euler method for time differencing. This method only uses information from the previous time level to evaluate the system for the next time level. ‘‘Explicit’’ means only one unknown u_i^{n+1} for the next time level appears in each equation.

The explicit forward Euler time discretization for Eq. (3.36) is described

$$\begin{aligned}
\frac{u_i^{n+1} - u_i^n}{\Delta t} + \frac{1}{2\Delta x} \left\{ \left(F(u_{i+1,j}^L) + F(u_{i,j}^R) \right) - \left(F(u_{i,j}^L) + F(u_{i-1,j}^R) \right) \right\} \\
- \frac{1}{2\Delta x} \left\{ \left| R(u_{i+1,j}^L, u_{i,j}^R) \right| (u_{i+1,j}^L - u_{i,j}^R) - \left| R(u_{i,j}^L, u_{i-1,j}^R) \right| (u_{i,j}^L - u_{i-1,j}^R) \right\} \\
+ \frac{1}{2\Delta z} \left\{ \left(F(u_{i,j+1}^L) + F(u_{i,j}^R) \right) - \left(F(u_{i,j}^L) + F(u_{i,j-1}^R) \right) \right\} \\
- \frac{1}{2\Delta z} \left\{ \left| R(u_{i,j+1}^L, u_{i,j}^R) \right| (u_{i,j+1}^L - u_{i,j}^R) - \left| R(u_{i,j}^L, u_{i,j-1}^R) \right| (u_{i,j}^L - u_{i,j-1}^R) \right\} = 0.
\end{aligned} \tag{3.37}$$

where u_i^n is the cell averaged value in the i^{th} cell at the n^{th} time step and Δt indicates the time step size. Generally the time step size is determined by the CFL condition (see Appendix A) for stability of the solution.

3.7 Central Difference Scheme for the Second Order Derivatives

The P_3QS method introduced in Chapter 4 includes second and mixed order derivatives, so we need to review how to discretize these derivatives. The most frequently used first-derivative approximations on a grid are [4]

$$\text{Forward difference scheme } \left. \frac{\partial u}{\partial x} \right)_{i,j} = \frac{u_{i+1,j} - u_{i,j}}{\Delta x} + O(\Delta x) \quad (3.38)$$

$$\text{Backward difference scheme } \left. \frac{\partial u}{\partial x} \right)_{i,j} = \frac{u_{i,j} - u_{i-1,j}}{\Delta x} + O(\Delta x) \quad (3.39)$$

$$\text{Central difference scheme } \left. \frac{\partial u}{\partial x} \right)_{i,j} = \frac{u_{i+1/2,j} - u_{i-1/2,j}}{2\Delta x} + O(\Delta x^2) \quad (3.40)$$

These are using only two neighboring grid points. In this dissertation, we will use the central difference scheme for second and mixed order derivatives.

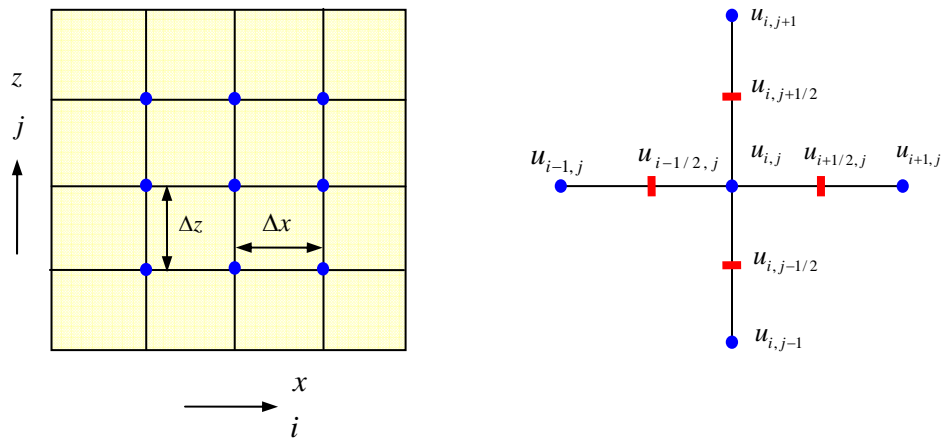


Figure 3.2 A typical finite difference grid

By Eq. (3.40), the central difference representation of the second order derivative is expressed as

$$\left. \frac{\partial^2 u}{\partial x^2} \right)_{i,j} = \frac{1}{\Delta x} \left[\left. \frac{\partial u}{\partial x} \right)_{i+1/2,j} - \left. \frac{\partial u}{\partial x} \right)_{i-1/2,j} \right] = \frac{1}{\Delta x} \left[\frac{u_{i+1,j} - u_{i,j}}{\Delta x} - \frac{u_{i,j} - u_{i-1,j}}{\Delta x} \right] = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta x^2} \quad (3.41)$$

and the mixed order derivative is evaluated as

$$\left. \frac{\partial^2 u}{\partial x \partial z} \right)_{i,j} = \frac{\partial}{\partial z} \left[\frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} \right] = \frac{1}{2\Delta x} \left[\left. \frac{\partial u}{\partial z} \right)_{i+1,j} - \left. \frac{\partial u}{\partial z} \right)_{i-1,j} \right] = \frac{1}{2\Delta x} \left[\frac{u_{i+1,j+1} - u_{i+1,j-1}}{2\Delta z} - \frac{u_{i-1,j+1} - u_{i-1,j-1}}{2\Delta z} \right]. \quad (3.42)$$

3.8 Boundary Conditions

We use vacuum boundary conditions for our problems in this dissertation. This boundary condition can be considered as the Mark boundary condition [10, 11] which means zero incoming current outside a physical system implying

$$\psi_l^m(\mathbf{r}, t) = \int_{n \cdot \mathbf{\Omega} < 0} \psi(\mathbf{r}, \mathbf{\Omega}, t) \bar{Y}_l^m(\mathbf{\Omega}) d\mathbf{\Omega} = 0$$

$$\mathbf{r} \in \partial V, \quad t > 0 \quad (3.43)$$

where ∂V is the outer boundary of the system V .

This boundary condition can be implemented with the ghost cell in a computational domain. The ghost cell method is expanding the computational domain to include a few additional cells outside the system whose values are set at the beginning of each time step based on the boundary conditions and the interior solution. By setting $\psi_l^m|_{ghost} = 0$ we can easily obtain the vacuum boundary conditions and this allows us to solve a Riemann problem at the boundary of the original domain.

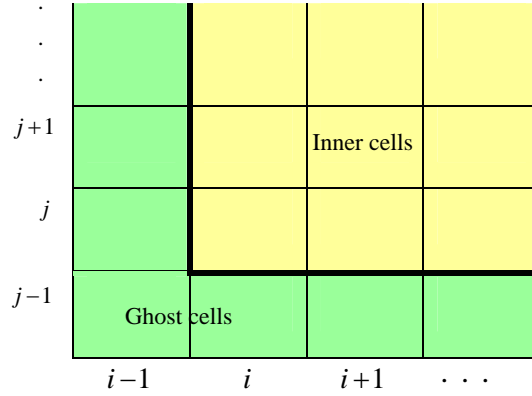


Figure 3.3 The left corner of a typical computational domain

For example, in a cell at the left edge of the system we set the value of $u_{i-1,j}^{\text{ghost}} = 0$, where $u_{i-1,j}^{\text{ghost}}$ is the value in the ghost cell. Then the flux across the boundary is (c.f. Eq. 3.23)

$$F_{i-1/2,j}^{\text{boundary}} = \frac{1}{2} \left[F(u_{i,j}^{\text{inner}}) + F(u_{i-1,j}^{\text{ghost}}) \right] - \frac{1}{2} \left| R(u_{i,j}^{\text{inner}}, u_{i-1,j}^{\text{ghost}}) \right| (u_{i,j}^{\text{inner}} - u_{i-1,j}^{\text{ghost}}) \quad (3.44)$$

Vacuum boundaries for other locations are similarly treated. Also how many rows of ghost cells are needed depends on stencil of algorithms. Generally high resolution schemes require two rows of ghost cells [7].

3.9 References

- [1] R. J. LeVeque, *Numerical Methods for Conservation Laws*, Birkhauser, pp. 58, 1992.
- [2] G. Green, *Essay on the Mathematical Theory of Electricity and Magnetism*, Self published, Nottingham, 1828.
- [3] C. A. J. Fletcher, *Computational Techniques for Fluid Dynamics I*, Springer, pp. 105, 1988.
- [4] J. C. Tannehill, *Computational Fluid Mechanics and Heat transfer*, Taylor and Francis, pp. 50, 1997.
- [5] P. L. Roe, "Approximate Riemann solvers, parameter vectors, and difference schemes," *Journal of Computational Physics*, vol. 43, pp. 357-372, 1981.
- [6] E. F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics*, Springer, pp. 341-369, 1999.
- [7] R. J. LeVeque, *Finite Volume Methods for Hyperbolic Problems*, Cambridge, pp. 153, 2002.
- [8] C. B. Laney, *Computational Gas dynamics*, Cambridge University Press, pp. 75-81, 1998.
- [9] B. van Leer, "Towards the Ultimate Conservative Difference Scheme V. A second order sequel to Godunov's method," *Journal of Computational Physics*, pp. 101-136, 1979.
- [10] G. I. Bell and S. Glasstone, "*Nuclear Reactor theory*," Kieger, 1970.
- [11] J. J. Duderstadt and L.J. Hamilton, *Nuclear Reactor Analysis*, Wiley & Sons, pp. 103, 1976.

CHAPTER 4

P_3 with Quasi-Static Model

4.1 Introduction

In this chapter, we propose the P_3 Quasi-Static (P_3QS) Model and it will be implemented for the pulsed source problem. In this model, we surrender hyperbolicity and produce a model with higher order derivatives generated by a quasi-static approximation. We begin this chapter with a description of the P_3 equations, the 3rd order spherical harmonics approximation for the time-dependent transport equation. In Section 4.3, we consider a quasi-static approximation inspired by diffusion theory, which is considered to be the key idea for this model. This will change the P_N equation from hyperbolic to non-hyperbolic. Based on this approximation, 2D and 1D P_3QS equations are described. We present in Section 4.4 and 4.5 numerical results for P_3QS in one and two dimensions respectively. The results show that P_3QS can still produce negative scalar fluxes in rapid transients in 2D, but is closer to the transport solutions at late times than the conventional P_N results.

4.2 P_3 equations for time-dependent transport

The P_3 method for the transport equation writes the angular dependence of Eq. (1.1) as a series expansion in spherical harmonics Y_l^m ($l = 0, 1, 2, 3$) and in 2D this results [1] in the system of equations

$$\frac{\partial}{\partial t} \psi_0^0 - \sqrt{\frac{2}{3}} \frac{\partial}{\partial x} \psi_1^1 + \sqrt{\frac{1}{3}} \frac{\partial}{\partial z} \psi_1^0 + \Sigma_t \psi_0^0 = \Sigma_s \psi_0^0 \quad (4.1)$$

$$\frac{\partial}{\partial t} \psi_1^0 - \sqrt{\frac{2}{5}} \frac{\partial}{\partial x} \psi_2^1 + \sqrt{\frac{1}{3}} \frac{\partial}{\partial z} \psi_0^0 + \sqrt{\frac{4}{15}} \frac{\partial}{\partial z} \psi_2^0 + \Sigma_t \psi_1^0 = 0 \quad (4.2)$$

$$\frac{\partial}{\partial t} \psi_2^0 + \sqrt{\frac{2}{15}} \frac{\partial}{\partial x} \psi_1^1 - \sqrt{\frac{12}{35}} \frac{\partial}{\partial x} \psi_3^1 + \sqrt{\frac{4}{15}} \frac{\partial}{\partial z} \psi_1^0 + \sqrt{\frac{9}{35}} \frac{\partial}{\partial z} \psi_3^0 + \Sigma_t \psi_2^0 = 0 \quad (4.3)$$

$$\frac{\partial}{\partial t} \psi_3^0 + \sqrt{\frac{6}{35}} \frac{\partial}{\partial x} \psi_2^1 + \sqrt{\frac{9}{35}} \frac{\partial}{\partial z} \psi_2^0 + \Sigma_t \psi_3^0 = 0 \quad (4.4)$$

$$\frac{\partial}{\partial t} \psi_1^1 - \sqrt{\frac{1}{6}} \frac{\partial}{\partial x} \psi_0^0 + \sqrt{\frac{1}{30}} \frac{\partial}{\partial x} \psi_2^0 - \sqrt{\frac{1}{5}} \frac{\partial}{\partial x} \psi_2^2 + \sqrt{\frac{1}{5}} \frac{\partial}{\partial z} \psi_2^1 + \Sigma_t \psi_1^1 = 0 \quad (4.5)$$

$$\frac{\partial}{\partial t} \psi_2^1 - \sqrt{\frac{1}{10}} \frac{\partial}{\partial x} \psi_1^0 + \sqrt{\frac{3}{70}} \frac{\partial}{\partial x} \psi_3^0 - \sqrt{\frac{1}{7}} \frac{\partial}{\partial x} \psi_3^2 + \sqrt{\frac{1}{5}} \frac{\partial}{\partial z} \psi_1^1 + \sqrt{\frac{8}{35}} \frac{\partial}{\partial z} \psi_3^1 + \Sigma_t \psi_2^1 = 0 \quad (4.6)$$

$$\frac{\partial}{\partial t} \psi_3^1 - \sqrt{\frac{3}{35}} \frac{\partial}{\partial x} \psi_2^0 + \sqrt{\frac{1}{70}} \frac{\partial}{\partial x} \psi_2^2 + \sqrt{\frac{8}{35}} \frac{\partial}{\partial z} \psi_2^1 + \Sigma_t \psi_3^1 = 0 \quad (4.7)$$

$$\frac{\partial}{\partial t} \psi_2^2 - \sqrt{\frac{1}{5}} \frac{\partial}{\partial x} \psi_1^1 + \sqrt{\frac{1}{70}} \frac{\partial}{\partial x} \psi_3^1 - \sqrt{\frac{3}{14}} \frac{\partial}{\partial x} \psi_3^3 + \sqrt{\frac{1}{7}} \frac{\partial}{\partial z} \psi_3^2 + \Sigma_t \psi_2^2 = 0 \quad (4.8)$$

$$\frac{\partial}{\partial t} \psi_3^2 - \sqrt{\frac{1}{7}} \frac{\partial}{\partial x} \psi_1^1 + \sqrt{\frac{1}{7}} \frac{\partial}{\partial z} \psi_2^2 + \Sigma_t \psi_3^2 = 0 \quad (4.9)$$

$$\frac{\partial}{\partial t} \psi_3^3 - \sqrt{\frac{3}{14}} \frac{\partial}{\partial x} \psi_2^2 + \Sigma_t \psi_3^3 = 0. \quad (4.10)$$

4.3 Quasi-Static Closure

In diffusion theory the time-rate-of-change of the current (first order expansion coefficients) is set to zero; in a P_3 approximation we can instead make the quasi-static approximation for the 3rd order coefficients. The assumption is that these higher order moments ψ_3^m can quickly change so that they follow the evolution of the lower moments in a quasi-static way. Therefore, setting $\partial \psi_3^m / \partial t \approx 0$ in Eqs. (4.4), (4.7), (4.9) and (4.10) allows us to eliminate the 3rd order coefficients as

$$\psi_3^0 = -\frac{1}{\Sigma_t} \left(\sqrt{\frac{6}{35}} \frac{\partial}{\partial x} \psi_2^1 + \sqrt{\frac{9}{35}} \frac{\partial}{\partial z} \psi_2^0 \right) \quad (4.11)$$

$$\psi_3^1 = -\frac{1}{\Sigma_t} \left(-\sqrt{\frac{3}{35}} \frac{\partial}{\partial x} \psi_2^0 + \sqrt{\frac{1}{70}} \frac{\partial}{\partial x} \psi_2^2 + \sqrt{\frac{8}{35}} \frac{\partial}{\partial z} \psi_2^1 \right) \quad (4.12)$$

$$\psi_3^2 = -\frac{1}{\Sigma_t} \left(-\sqrt{\frac{1}{7}} \frac{\partial}{\partial x} \psi_2^1 + \sqrt{\frac{1}{7}} \frac{\partial}{\partial z} \psi_2^2 \right) \quad (4.13)$$

$$\psi_3^3 = -\frac{1}{\Sigma_t} \left(-\sqrt{\frac{3}{14}} \frac{\partial}{\partial x} \psi_2^2 \right). \quad (4.14)$$

These are analogous to Fick's law in diffusion theory. Substituting Eq. (4.11) - (4.14) into the remaining 6 equations Eq. (4.1) - (4.10) gives a new model, which we call P_3 with quasi-static closure (P_3QS) with governing equations

$$\frac{\partial}{\partial t} \psi_0^0 - \sqrt{\frac{2}{3}} \frac{\partial}{\partial x} \psi_1^1 + \sqrt{\frac{1}{3}} \frac{\partial}{\partial z} \psi_1^0 + \Sigma_t \psi_0^0 = \Sigma_s \psi_0^0 \quad (4.15)$$

$$\frac{\partial}{\partial t} \psi_1^0 - \sqrt{\frac{2}{5}} \frac{\partial}{\partial x} \psi_2^2 + \sqrt{\frac{1}{3}} \frac{\partial}{\partial z} \psi_0^0 + \sqrt{\frac{4}{15}} \frac{\partial}{\partial z} \psi_2^0 + \Sigma_t \psi_1^0 = 0 \quad (4.16)$$

$$\frac{\partial}{\partial t} \psi_2^0 + \sqrt{\frac{2}{15}} \frac{\partial}{\partial x} \psi_1^1 + \sqrt{\frac{4}{15}} \frac{\partial}{\partial z} \psi_1^0 - \frac{1}{\Sigma_t} \frac{6}{35} \frac{\partial^2}{\partial x^2} \psi_2^0 + \frac{1}{\Sigma_t} \frac{2\sqrt{6}}{70} \frac{\partial^2}{\partial x^2} \psi_2^2 - \frac{1}{\Sigma_t} \frac{9}{35} \frac{\partial^2}{\partial z^2} \psi_2^0 + \frac{1}{\Sigma_t} \frac{\sqrt{6}}{35} \frac{\partial^2}{\partial x \partial z} \psi_2^1 + \Sigma_t \psi_2^0 = 0 \quad (4.17)$$

$$\frac{\partial}{\partial t} \psi_1^1 - \sqrt{\frac{1}{6}} \frac{\partial}{\partial x} \psi_0^0 + \sqrt{\frac{1}{30}} \frac{\partial}{\partial x} \psi_2^0 - \sqrt{\frac{1}{5}} \frac{\partial}{\partial x} \psi_2^2 + \sqrt{\frac{1}{5}} \frac{\partial}{\partial z} \psi_2^1 + \Sigma_t \psi_1^1 = 0 \quad (4.18)$$

$$\frac{\partial}{\partial t} \psi_2^1 - \sqrt{\frac{1}{10}} \frac{\partial}{\partial x} \psi_1^0 + \sqrt{\frac{1}{5}} \frac{\partial}{\partial z} \psi_1^1 - \frac{1}{\Sigma_t} \frac{8}{35} \frac{\partial^2}{\partial x^2} \psi_2^1 - \frac{1}{\Sigma_t} \frac{8}{35} \frac{\partial^2}{\partial z^2} \psi_2^1 + \frac{1}{\Sigma_t} \frac{\sqrt{6}}{70} \frac{\partial^2}{\partial x \partial z} \psi_2^0 + \frac{1}{\Sigma_t} \frac{3}{35} \frac{\partial^2}{\partial x \partial z} \psi_2^2 + \Sigma_t \psi_2^1 = 0 \quad (4.19)$$

$$\frac{\partial}{\partial t} \psi_2^2 - \sqrt{\frac{1}{5}} \frac{\partial}{\partial x} \psi_1^1 + \frac{1}{\Sigma_t} \frac{\sqrt{6}}{70} \frac{\partial^2}{\partial x^2} \psi_2^0 - \frac{1}{\Sigma_t} \frac{8}{35} \frac{\partial^2}{\partial x^2} \psi_2^2 - \frac{1}{\Sigma_t} \frac{1}{7} \frac{\partial^2}{\partial z^2} \psi_2^2 + \frac{1}{\Sigma_t} \frac{3}{35} \frac{\partial^2}{\partial x \partial z} \psi_2^1 + \Sigma_t \psi_2^2 = 0. \quad (4.20)$$

Note that these equations now contain second order derivative terms, like time-dependent diffusion theory. Because of these higher order derivative terms, we find this system can not be written as a quasi-linear coupled system of advection equations and the hyperbolicity (see chapter 3.1) is lost. This loss of hyperbolicity is the main difference between regular P_3 and P_3QS .

In addition, the 1D form of P_3 with a quasi-static closure can be obtained from Eq. (4.15) - (4.17) by dropping x-dependent terms as,

$$\frac{\partial}{\partial t} \psi_0^0 + \sqrt{\frac{1}{3}} \frac{\partial}{\partial z} \psi_1^0 + \Sigma_t \psi_0^0 = \Sigma_s \psi_0^0 \quad (4.21)$$

$$\frac{\partial}{\partial t} \psi_1^0 + \sqrt{\frac{1}{3}} \frac{\partial}{\partial z} \psi_0^0 + \sqrt{\frac{4}{15}} \frac{\partial}{\partial z} \psi_2^0 + \Sigma_t \psi_1^0 = 0 \quad (4.22)$$

$$\frac{\partial}{\partial t} \psi_2^0 + \sqrt{\frac{4}{15}} \frac{\partial}{\partial z} \psi_1^0 - \frac{1}{\Sigma_t} \frac{9}{35} \frac{\partial^2}{\partial z^2} \psi_2^0 + \Sigma_t \psi_2^0 = 0 \quad (4.23)$$

4.4 Numerical Results in 1D

The one-dimensional time-dependent behavior of P_3QS is tested by solving the pulsed source problem for the scalar flux in a slab. The slab size is 20 cm and consists of a purely scattering medium ($\Sigma_t = \Sigma_s = 1 \text{ cm}^{-1}$). A delta function initially isotropic flux is imposed to provide the Green's function solution. The spatial grid and time steps initially are chosen as $\Delta x = 0.002$, $\Delta t = 0.001$ respectively.

Figure 4.1 shows comparison of P_1 , transport, and the P_3QS solution at one second after the pulse. The transport result is from the Ganapol's analytic solution [10, 11]. The collided flux shown between the two wave fronts in P_3QS agrees better with the transport solution than does P_1 . Note the non-zero flux in front of the two wave fronts of P_3QS ; this is a reflection of the non-hyperbolicity of the method. In P_3QS , particles can travel with infinite speed, just as they can in diffusion theory.

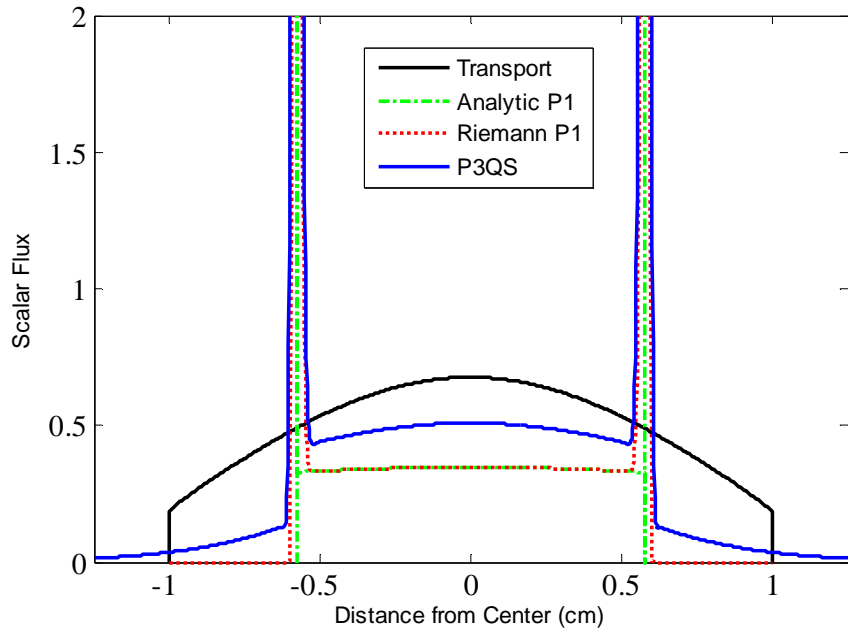


Figure 4.1 Comparison of P_3QS solution with P_1 and the transport solution at 1 second after the pulse in 1D

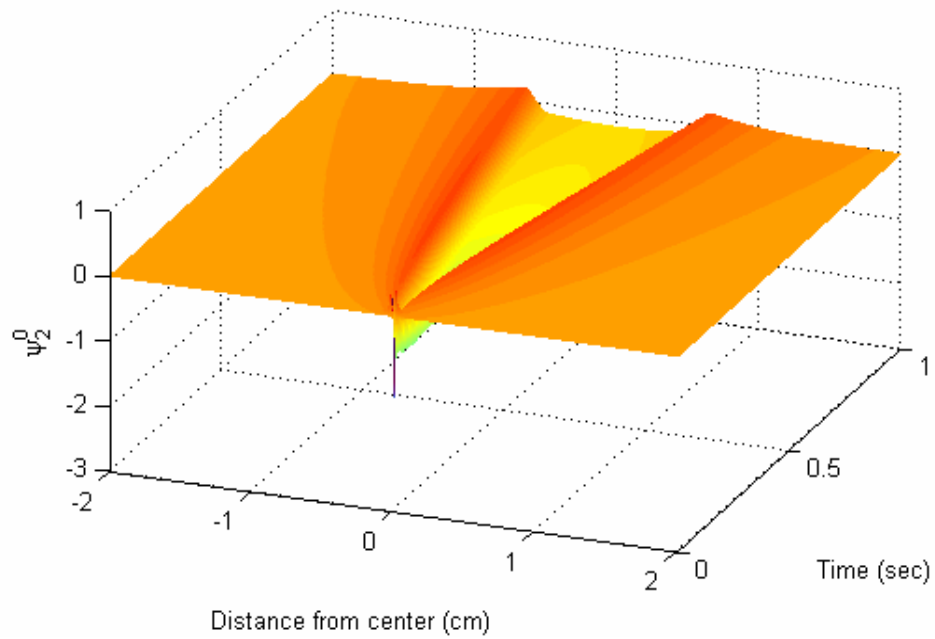


Figure 4.2 ψ_2^0 of P_3QS solution as a function of time and space

Ignoring the second order terms, characteristic speeds of P_3QS can be obtained from the Jacobian of Eq. (4.21) and (4.23) and are $\pm\sqrt{3/5}$. These speeds are identical to the non-zero characteristic speeds of P_2 theory. They are different from the characteristic speeds ($\pm\sqrt{1/3}$) of P_1 . But in Figure 4.1 we see two peaks in the P_3QS solution moving at the characteristic P_1 speeds. To understand this, note that the wave speed of P_3QS looks like that of P_1 when the derivative of ψ_2^0 becomes 0. At a point where $\partial \psi_2^0 / \partial t \approx 0$ we have 1D P_3QS equation as

$$\frac{\partial}{\partial t} \psi_0^0 + \sqrt{\frac{1}{3}} \frac{\partial}{\partial z} \psi_1^0 + \Sigma_t \psi_0^0 = \Sigma_s \psi_0^0 \quad (4.24)$$

$$\frac{\partial}{\partial t} \psi_1^0 + \sqrt{\frac{1}{3}} \frac{\partial}{\partial z} \psi_0^0 + \Sigma_t \psi_1^0 \approx 0 \quad (4.25)$$

$$\frac{\partial}{\partial t} \psi_2^0 + \sqrt{\frac{4}{15}} \frac{\partial}{\partial z} \psi_1^0 - \frac{1}{\Sigma_t} \frac{9}{35} \frac{\partial^2}{\partial z^2} \psi_2^0 + \Sigma_t \psi_2^0 = 0 \quad (4.26)$$

These equations have eigenvalues 0 and $\pm\sqrt{1/3}$ where the non-zero eigenvalues are the same with the characteristic speeds of P_1 . Figure 4.2 shows ψ_2^0 of P_3QS solution as a function of time and space. ψ_2^0 has a maximum at the location of the uncollided flux peaks, so $\partial \psi_2^0 / \partial t \approx 0$ at these peaks. We can see that this maximum coincides with the peaks moving at speeds $\pm\sqrt{1/3}$. Further, ψ_2^0 becomes everywhere smaller as time goes on, as shown in Figure 4.2, and so the P_3QS solution behaves increasingly like the P_1 solution at later times.

Figure 4.3 shows a comparison of P_3QS , P_1 , P_5 , P_{15} and the transport solution at 5 seconds after the pulse. P_3QS is closer to the transport solution than either P_1 or P_5 and it looks close to the P_{15} results. But P_3QS is much more efficient than P_{15} because it has fewer unknowns and requires fewer computational resources. P_{15} has 16 unknowns per cell, but P_3QS has only 3 unknowns (See Table 4.1 on page 39).

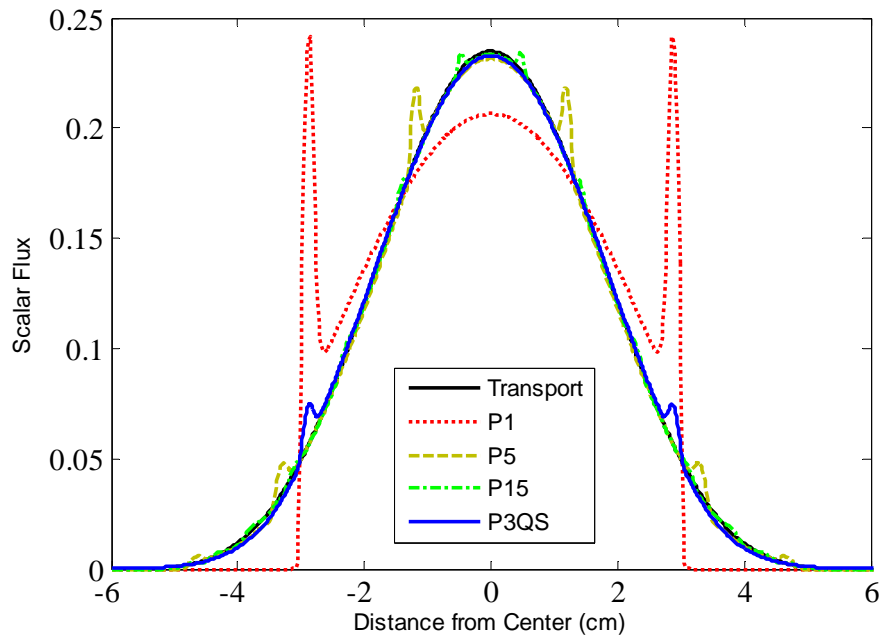


Figure 4.3 Comparison of P_3QS solution with P_1 , P_5 , P_{15} and transport solution at 5 seconds after the pulse in 1D

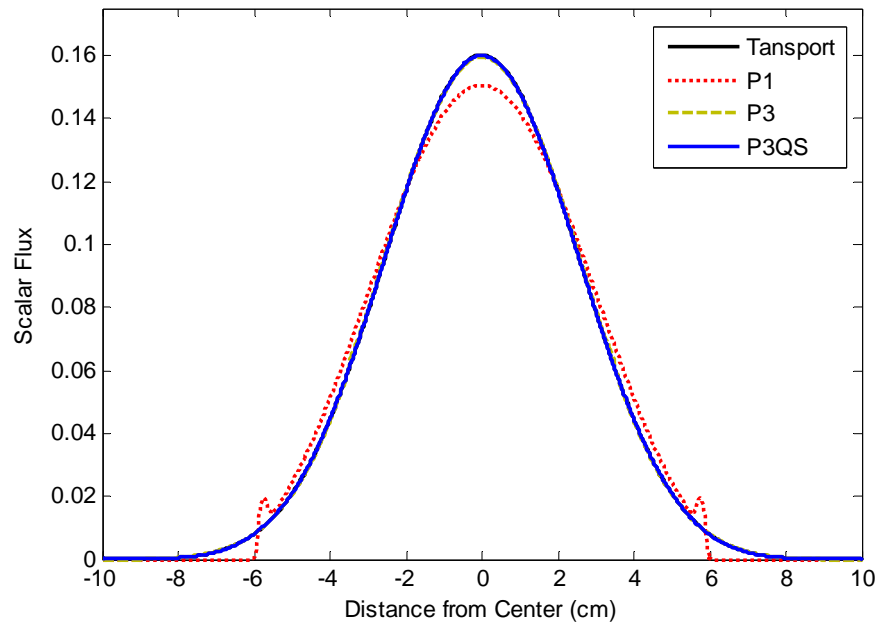


Figure 4.4 Comparison of P_3QS solution with P_1 , P_3 and transport solution at 10 seconds after the pulse in 1D

Figure 4.4 shows a comparison of P_3QS , P_1 , P_3 and the transport solution at 10 seconds after the pulse. P_3QS does exceptionally well at matching the shape of the transport solution at later times when most of the particle have undergone multiple collisions. Note the P_1 does less well, even after 10 scattering times. It is difficult for P_1 to overcome the error generated early in the transient.

4.5 Numerical results in 2D

Our two-dimensional test problem is a 20 cm square of purely scattering medium ($\Sigma_t = \Sigma_s = 1 \text{ cm}^{-1}$). A pulsed line source is imposed as an initial condition at the center of the system and vacuum boundary conditions are used.

Figure 4.5 shows a comparison of the P_3QS solution with P_1 , P_3 , P_9 and the transport solution at one second after the pulse. From this figure we see that at short times none of the P_N results is a good approximation to the transport solution because the scalar fluxes become negative. Also, the higher order P_N methods are increasingly oscillatory and increasingly negative at $r=0$. But we can see that P_3QS is less negative and less oscillatory than any of the P_N results. Also it shows a positive flux near the center of the domain, although it still has a negative dip behind the front of uncollided particles.

Figure 4.6 shows a comparison of the solutions at 5 seconds after the pulse. At this time, all of the P_N solutions have become positive and they still have oscillations. However, the higher order solutions are less oscillatory and closer to the transport solution. Moreover the P_3QS solution is less oscillatory and closer to the transport solution than any other P_N result. P_9 looks close to P_3QS but it has far more unknowns per cell than P_3QS (55 vs. 6).

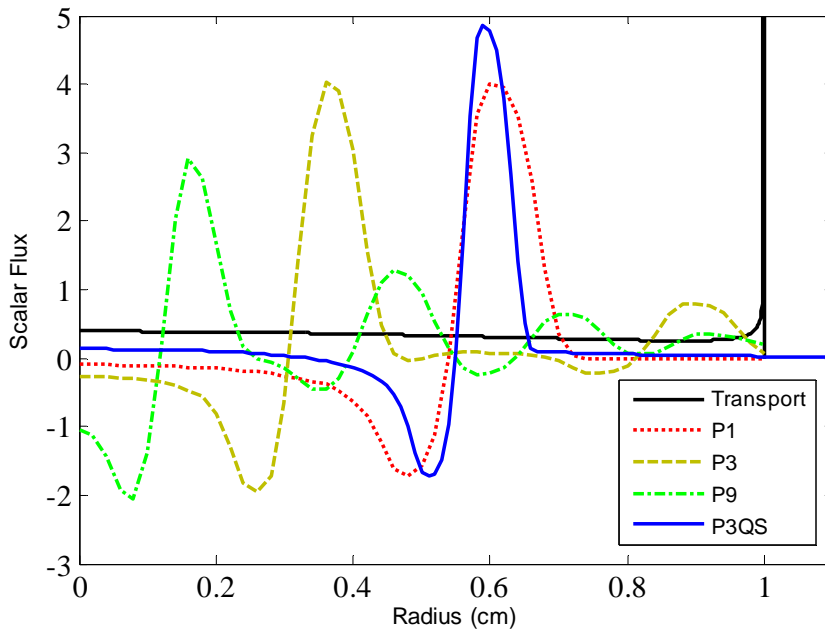


Figure 4.5 Comparison of P_3QS with P_1 , P_3 , P_9 and the transport solution at 1 second after the pulse in 2D

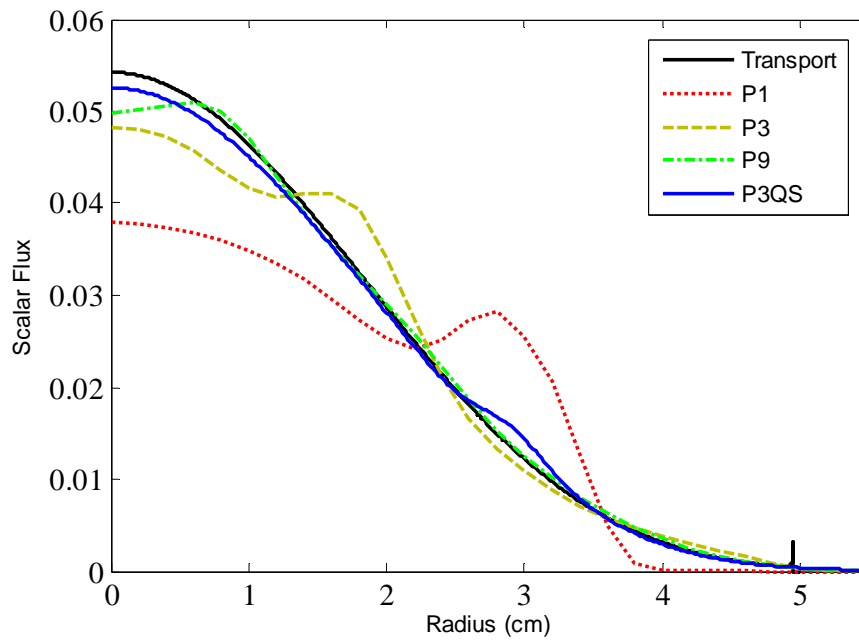


Figure 4.6 Comparison of P_3QS solution with P_1 , P_3 , P_9 and transport solution at 5 seconds after the pulse in 2D

Table 4.1 Comparison of the number of unknowns for P_N vs. P_3QS

Dimensions	Expansion Order						P_3QS
	P_1	P_3	P_5	P_9	P_{15}	P_N	
One	2	4	6	10	16	$N + 1$	3
Two	3	10	21	55	136	$1/2N^2 + 3/2N + 1$	6

Figure 4.7 shows a comparison of the solutions at 10 seconds after the pulse. At ten seconds higher order P_N solutions match transport result fairly well and the P_3QS solution is nearly indistinguishable from the exact transport solution on the scale of this graph. Note that the P_3QS solution is better than the P_3 solution at $t = 10$. While none of the methods was accurate at short times, the P_3QS method can move through the short time transient more successfully.

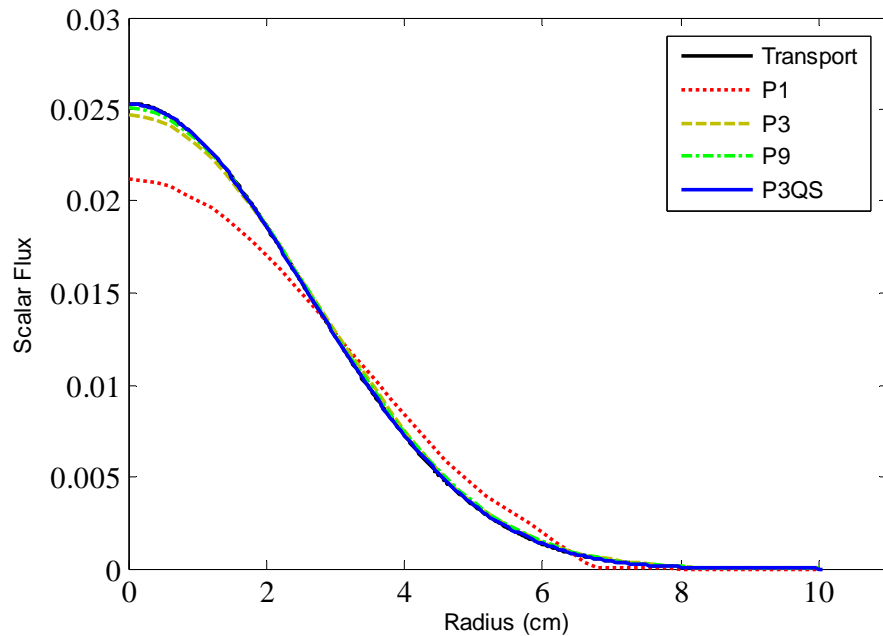


Figure 4.7 Comparison of P_3QS solution with P_1, P_3, P_9 and transport solution at 10 seconds after the pulse in 2D

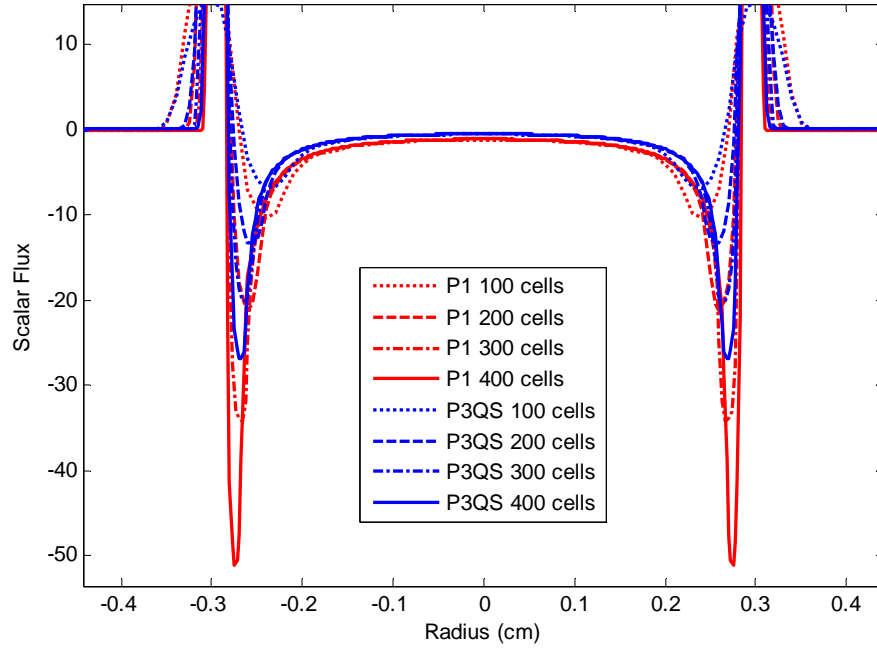


Figure 4.8 Negative fluxes for P_3QS and P_1 solutions at 0.5 second after the pulse in 2D

Figure 4.8 shows the negative fluxes for P_3QS and P_1 solutions computed using a high-resolution Riemann solver from 100 to 400 cells per cm at 0.5 second after the pulse. This shows that P_3QS is less negative than the P_1 solution near the center of the domain, but it is difficult to fully resolve the negative spike behind the wave front.

In order to try to see the convergence of the negative flux, the volume integral of the negative scalar flux is defined as,

$$\mathbf{VI} = \int_{-2}^2 \int_{-2}^2 (|\psi_0^0| - \psi_0^0) dz dx \quad (4.27)$$

where ψ_0^0 is a scalar flux. Figure 4.9 shows a comparison of this volume integral for P_3QS and P_1 solutions as a function of the number of cells at 0.5 second after the pulse.

As the number of cells increases, **VI** for both solutions increases and there is no evidence of convergence. Note however that **VI** for the P_3QS method is always smaller than that for the P_1 method.

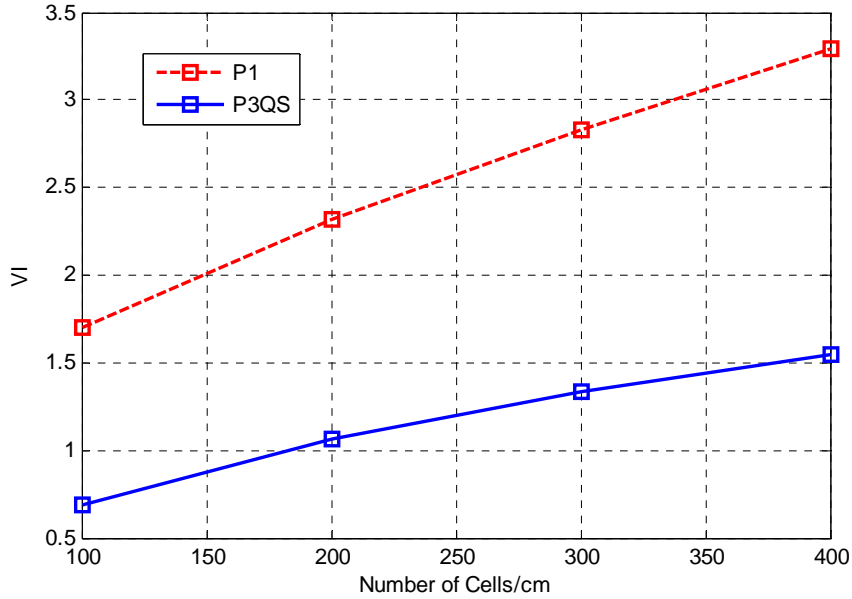


Figure 4.9 Comparison of the volume integrals of negative fluxes for P_3QS and P_1

So far all our results have been for a pure scatterer. Here we briefly present some results for various scattering ratios. Figure 4.10 and Figure 4.11 show the solutions for various scattering ratios ($c = \Sigma_s / \Sigma_t$) at 1 second in 1D and 2D respectively. Here, the plot of $c = 0$ implies the solution in pure absorber and $c = 1$, pure scatterer. From these figures we can see that, as expected, the collided flux behind the wave front increases as the scattering ratio increases.

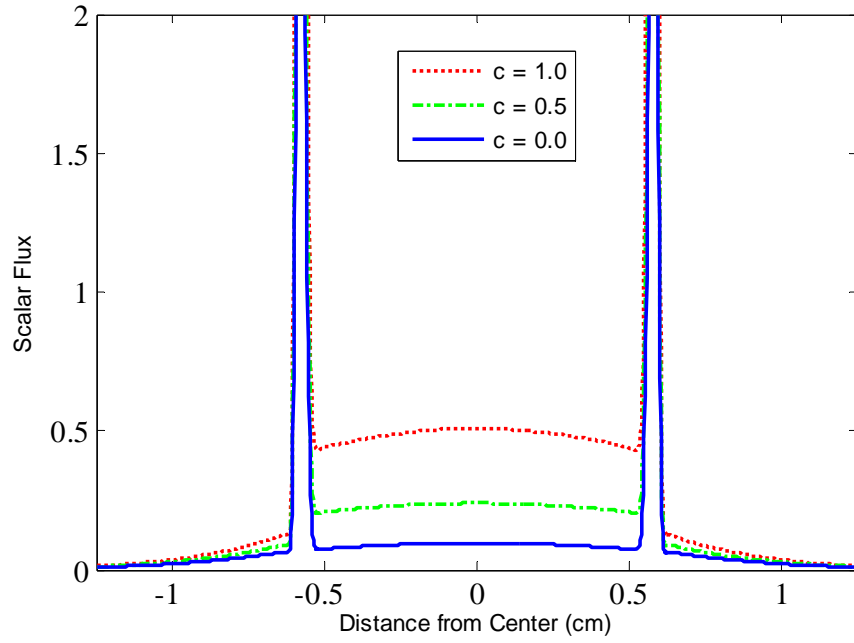


Figure 4.10 P_3QS solution for various scattering ratios at 1 second in 1D

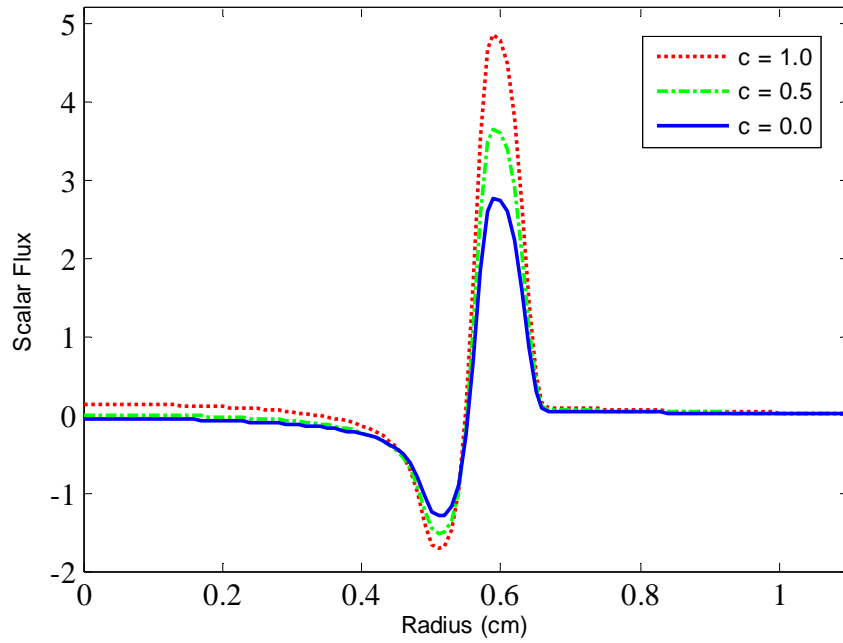


Figure 4.11 P_3QS solution for various scattering ratios at 1 second in 2D

4.8 References

- [1] K. S. Oh, J. P. Holloway, "A quasi-static closure for 3rd order spherical harmonics time-dependent radiation transport in 2-D," *International Conference on Mathematics, Computational Methods & Reactor Physics*, American Nuclear Society, 2009.
- [2] R. G. McClarren, *Spherical Harmonics Methods for Thermal Radiation Transport*, Ph.D. Thesis, University of Michigan, Ann Arbor, Michigan, 2007.
- [3] R. G. McClarren, J. P. Holloway, T. A. Brunner, T. A. Mehlhorn, "A quasi-linear implicit Riemann solver for the time-dependent P_N equations," *Nuclear Science and Engineering*, vol. 155, pp. 290-299, 2007.
- [4] R. G. McClarren, J. P. Holloway, T. A. Brunner, "On Solutions to the P_N equations for thermal radiative transfer," *Journal of Computational Physics*, vol. 227, 2008.
- [5] G. N. Minerbo, "Maximum Entropy Eddington Factors" *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol.20, pp. 541, 1978.
- [6] J. E. Morel, T. A. Wareing, R. B. Lowrie, D. K. Parsons, "Analysis of ray-effect mitigation techniques," *Nuclear Science and Engineering*, vol. 144, pp. 1-22, 2003.
- [7] M. Frank, E. W. Larsen, "Time-dependent simplified P_N approximation to the equations of radiative transfer" *Journal of Computational Physics*, vol. 226, pp. 386-399, 2005.
- [8] E. E. Lewis and W. F. Miller, *Computational Methods of Neutron Transport*, John Wiley and Sons, pp. 194, 1984.
- [9] B. van Leer, "Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method," *Journal of Computational Physics*, vol. 135, pp. 229-248, 1997.
- [10] B. D. Ganapol, "Solution of the one-group time-dependent neutron transport equation in and infinite medium by polynomial reconstruction," *Nuclear Science and Engineering*, vol. 92, pp. 272-279, 1986.
- [11] B. D. Ganapol, "The Generation of Time-Dependent Neutron Transport Solution in Infinite Media," *Nuclear Science and Engineering*, vol. 64, pp. 317-331, 1977.

CHAPTER 5

Nonlinear P_1 model

5.1 Introduction

In this chapter, we propose a new nonlinear Eddington factor model in which the property of linearity of regular P_1 has been given up. We begin this chapter with a derivation of the general form of the Eddington tensor and the Eddington factor. In Section 5.3, we consider properties of various Eddington factor models in 1D by investigating discontinuity structure, relaxation length and eigenstructure. Based on the eigenstructure of Maximum Entropy closure, our Nonlinear P_1 Eddington factor is constructed in Section 5.4 and compared with other Eddington factor models in Section 5.5. Riemann solver construction for Nonlinear P_1 closure is described in Section 5.6. Finally, we present in Section 5.7 and 5.8 the numerical results for 1D and 2D. The results will demonstrate that this model can keep the scalar flux positive for rapid transients in 2D.

5.2 Eddington Tensor in 3D and Eddington factor

We suppose the angular flux written as $\psi = \psi(\boldsymbol{\eta} \cdot \boldsymbol{\Omega})$ where $\boldsymbol{\eta} = \mathbf{J} / \phi$ and ϕ , \mathbf{J} and \mathbf{P} are, respectively, the scalar flux, current and pressure tensor:

$$\phi = \int \psi(\boldsymbol{\eta} \cdot \boldsymbol{\Omega}) d\boldsymbol{\Omega} \quad (5.1)$$

$$\mathbf{J} = \int \boldsymbol{\Omega} \psi(\boldsymbol{\eta} \cdot \boldsymbol{\Omega}) d\boldsymbol{\Omega} \quad (5.2)$$

$$\mathbf{P} = \int \boldsymbol{\Omega} \otimes \boldsymbol{\Omega} \psi(\boldsymbol{\eta} \cdot \boldsymbol{\Omega}) d\boldsymbol{\Omega}, \quad (5.3)$$

The Eddington tensor χ is defined as the normalized second moment

$$\chi = \frac{\mathbf{P}}{\phi}. \quad (5.4)$$

With $\Omega = [\sqrt{1-\mu^2} \cos \gamma, \sqrt{1-\mu^2} \sin \gamma, \mu]^T$ expressed using polar coordinates, the outer product in Eq. (5.3) expands to

$$\Omega \otimes \Omega = \begin{bmatrix} (1-\mu^2) \cos^2 \gamma & (1-\mu^2) \cos \gamma \sin \gamma & \mu \sqrt{1-\mu^2} \cos \gamma \\ (1-\mu^2) \cos \gamma \sin \gamma & (1-\mu^2) \sin^2 \gamma & \mu \sqrt{1-\mu^2} \sin \gamma \\ \mu \sqrt{1-\mu^2} \cos \gamma & \mu \sqrt{1-\mu^2} \sin \gamma & \mu^2 \end{bmatrix} \quad (5.5)$$

and Eq. (5.3) is thus written as

$$\mathbf{P} = \int_{2\pi} \int_{-1}^1 \begin{bmatrix} (1-\mu^2) \cos^2 \gamma & (1-\mu^2) \cos \gamma \sin \gamma & \mu \sqrt{1-\mu^2} \cos \gamma \\ (1-\mu^2) \cos \gamma \sin \gamma & (1-\mu^2) \sin^2 \gamma & \mu \sqrt{1-\mu^2} \sin \gamma \\ \mu \sqrt{1-\mu^2} \cos \gamma & \mu \sqrt{1-\mu^2} \sin \gamma & \mu^2 \end{bmatrix} \psi(\boldsymbol{\eta} \cdot \boldsymbol{\Omega}) d\mu d\gamma. \quad (5.6)$$

If we choose the pole of Ω to be the same direction as $\boldsymbol{\eta}$, the only dependence on γ in the integrand is contained in the tensor in Eq. (5.6), and all the off-diagonal terms integrate to zero.

$$\mathbf{P} = \int_{-1}^1 \begin{bmatrix} (1-\mu^2)\pi & 0 & 0 \\ 0 & (1-\mu^2)\pi & 0 \\ 0 & 0 & 2\pi\mu^2 \end{bmatrix} \psi(\eta\mu) d\mu. \quad (5.7)$$

Also if we define the n^{th} angular cosine moment as

$$m_n(\eta) = 2\pi \int_{-1}^1 \mu^n \psi(\eta\mu) d\mu \quad n = 0, 1, 2, \dots, \quad (5.8)$$

Eq. (5.7) becomes

$$\mathbf{P} = \begin{bmatrix} \frac{1}{2}(m_0(\eta) - m_2(\eta)) & 0 & 0 \\ 0 & \frac{1}{2}(m_0(\eta) - m_2(\eta)) & 0 \\ 0 & 0 & 2m_2(\eta) \end{bmatrix} \quad (5.9)$$

We can separate Eq. (5.9) into two parts,

$$\mathbf{P} = \frac{1}{2}(m_0(\eta) - m_2(\eta)) \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \frac{1}{2}(3m_2(\eta) - m_0(\eta)) \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.10)$$

These tensors are in the coordinate system where the pole of $\mathbf{\Omega}$ is in the direction of $\boldsymbol{\eta} = \mathbf{J}/\phi$, so Eq. (5.10) can be written generally as

$$\mathbf{P} = \frac{1}{2}(m_0 - m_2)\mathbf{I} + \frac{1}{2}(3m_2 - m_0)\frac{\mathbf{J} \otimes \mathbf{J}}{J^2} \quad (5.11)$$

Factoring out m_0 , Eq. (5.11) becomes,

$$\mathbf{P} = \frac{m_0(\eta)}{2} \left(1 - \frac{m_2(\eta)}{m_0(\eta)} \right) \mathbf{I} + \frac{m_0(\eta)}{2} \left(3 \frac{m_2(\eta)}{m_0(\eta)} - 1 \right) \frac{\mathbf{J} \otimes \mathbf{J}}{J^2} \quad (5.12)$$

Now define the Eddington factor $\chi(\eta)$

$$\chi(\eta) = \frac{m_2(\eta)}{m_0(\eta)} = \frac{\int_{-1}^1 \mu^2 \psi(\eta\mu) d\mu}{\int_{-1}^1 \psi(\eta\mu) d\mu} \quad (5.13)$$

and recall that $m_0 = \phi$, and so Eq. (5.12) can be written as

$$\mathbf{P} = \frac{\phi}{2}(1-\chi)\mathbf{I} + \frac{\phi}{2}(3\chi-1)\frac{\mathbf{J} \otimes \mathbf{J}}{J^2} \quad (5.14)$$

From Eq. (5.14) and the definition Eq. (5.4), we find the Eddington tensor χ has the following generalized form

$$\chi = \frac{1}{2}(1-\chi)\mathbf{I} + \frac{1}{2}(3\chi-1)\frac{\mathbf{J} \otimes \mathbf{J}}{J^2}. \quad (5.15)$$

Note that the Eddington tensor χ in Eq. (5.15) consists of two terms, an isotropic term and a beam-like term. $\chi(\eta)$ in Eq. (5.15) depends on ϕ and J through their ratio, and is called the variable Eddington factor [1-4].

5.3 Properties of Eddington Factor Models

The full allowed range for η , a function of ϕ and J through their ratio is $-1 \leq \eta \leq 1$. An isotropic flux and a mono-directional beam represent two extremes for the angular dependence of the angular flux. The limit $\eta = 0$ implies $J = 0$ and it corresponds to the isotropic in angle, while $\eta = 1$ means $\phi = J$, which is reached when all particles flow in the same direction.

For an Eddington factor to give a linear model it must be constant, and it is well known that P_1 theory has $\chi = 1/3$ as its Eddington factor over the full range $-1 \leq \eta \leq 1$. For a non-constant Eddington factor a nonlinear transport model will result. In the isotropic case, the pressure tensor \mathbf{P} in Eq. (5.14) must also be isotropic, and $\chi(0) = 1/3$ as is evident from Eq. (5.14). In the opposite stream, when particles flow in one direction, χ must obey $\chi(1) = 1$ as it is again evident from Eq. (5.14). So we have two properties for an Eddington factor that captures both extremes

P1.
$$\chi(\eta) \xrightarrow{\eta \rightarrow 0} \frac{1}{3} \quad (5.16)$$

P2.
$$\chi(\eta) \xrightarrow{\eta \rightarrow \pm 1} 1. \quad (5.17)$$

As a nonlinear Eddington factor, Minerbo suggested a Maximum Entropy Eddington factor based on the Entropy principle [3] and defined implicitly by

$$\chi(\eta) = 1 - 2\frac{\eta}{b}, \quad \eta = \coth b - \frac{1}{b} \quad (5.18)$$

This Eddington factor has been applied for time-dependent transport by Brunner [5, 6] and it can be a good approximation to the transport solution in 1D.

Discontinuity structure

Now we inspect the one-dimensional equation of variable Eddington closures to get some insight into the discontinuities of the solutions. The two moment closure equations in steady state can be written as

$$\frac{\partial}{\partial x} \begin{bmatrix} J \\ \chi\phi \end{bmatrix} = \begin{bmatrix} \Sigma_a \phi \\ \Sigma_t J \end{bmatrix} + \begin{bmatrix} S_0 \\ 0 \end{bmatrix} \quad (5.19)$$

We then integrate Eq. (5.19) over a very small spatial region,

$$\int_{x-\varepsilon}^{x+\varepsilon} \left(\frac{\partial}{\partial x} \begin{bmatrix} J \\ \chi\phi \end{bmatrix} = \begin{bmatrix} \Sigma_a \phi \\ \Sigma_t J \end{bmatrix} + \begin{bmatrix} S_0 \\ S_1 \end{bmatrix} \right) dx \quad (5.20)$$

As long as ϕ , J and sources are finite, we find that as $\varepsilon \rightarrow 0$

$$\begin{bmatrix} J(x+\varepsilon) \\ \chi\phi(x+\varepsilon) \end{bmatrix} - \begin{bmatrix} J(x-\varepsilon) \\ \chi\phi(x-\varepsilon) \end{bmatrix} = 0 \quad (5.21)$$

This implies that J is continuous, and the product $\chi\phi$ is continuous. However, the scalar flux ϕ or the Eddington factor χ need not be independently continuous. If we divide $\chi\phi$ by J , which is continuous, we get χ/η , and this must therefore also be continuous.

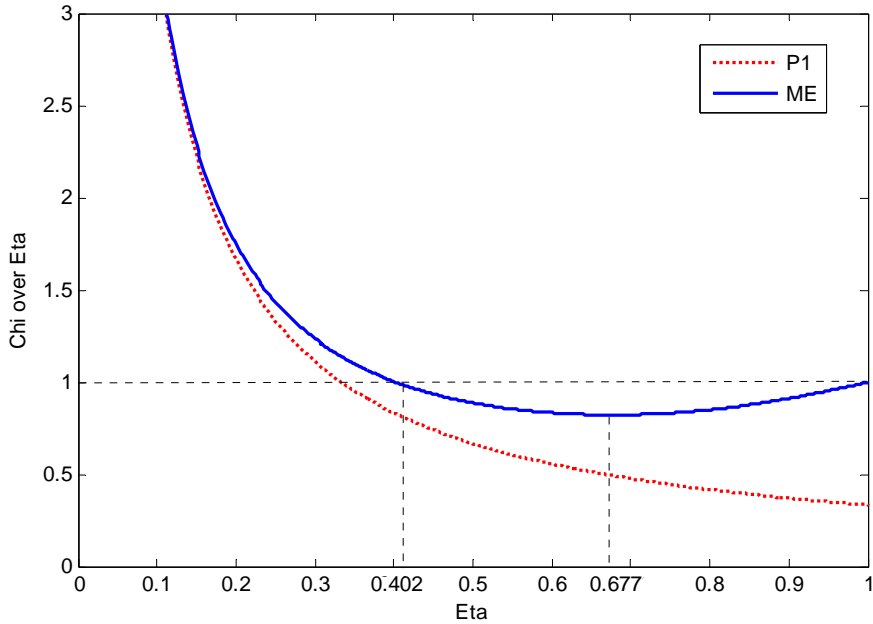


Figure 5.1 Plot of χ/η as a function of η

Figure 5.1 shows the χ/η as a function of η for the P_1 and the Maximum Entropy (ME) closures. For the ME solution, there are two different values η for each value of χ/η for $\eta > 0.402$. This implies that discontinuous solutions are allowed in this regime and that in this discontinuity η jumps from one side of the minimum of χ/η , at $\eta \approx 0.677$, to the other. At such jumps, χ/η does not change, while η and hence ϕ , does change discontinuously. In the regime where $\eta < 0.402$, there is only one value of χ/η for each η discontinuous solution can only occur when $\eta > 0.402$. In contrast there is no such minimum for the P_1 closure, implying it has no discontinuity.

Relaxation length

A comparison of the steady state relaxation length versus scattering ratio for various models will also give us some insights their qualities. The relaxation length is the distance in which the flux decreases by the factor of e (base of the natural log).

Consider a solution of the form $\psi_i = A_i e^{x/L}$ in the two moment closure equations in steady state, yielding

$$\frac{A_1}{L} + \Sigma_t A_0 = \Sigma_s A_0 \quad (5.22)$$

$$\chi(\eta) \frac{A_0}{L} + \Sigma_t A_1 = 0. \quad (5.23)$$

Introducing the scattering ratio, $c = \Sigma_s / \Sigma_t$ and the dimensionless relaxation length, $\nu = L \Sigma_t$ to Eq. (5.22) we can get

$$(c-1)\nu = \frac{A_1}{A_0} \quad (5.24)$$

Substitute Eq. (5.24) into (5.23) and use the relaxation length, $\nu = L \Sigma_t$ yielding,

$$\chi(\eta) = \nu^2 (1-c) \quad (5.25)$$

where χ is a function of η , and η can be defined as a function of c and ν by

$$\eta = \frac{\psi_1}{\psi_0} = \frac{A_1 e^{x/L}}{A_0 e^{x/L}} = \frac{A_1}{A_0} = \nu(c-1). \quad (5.26)$$

Figure 5.2 shows the plot of relaxation length for transport, P_1 and ME closures. The relaxation length of the ME is everywhere closer to that of transport than is P_1 . We can also see the relaxation length of ME has a minimum.

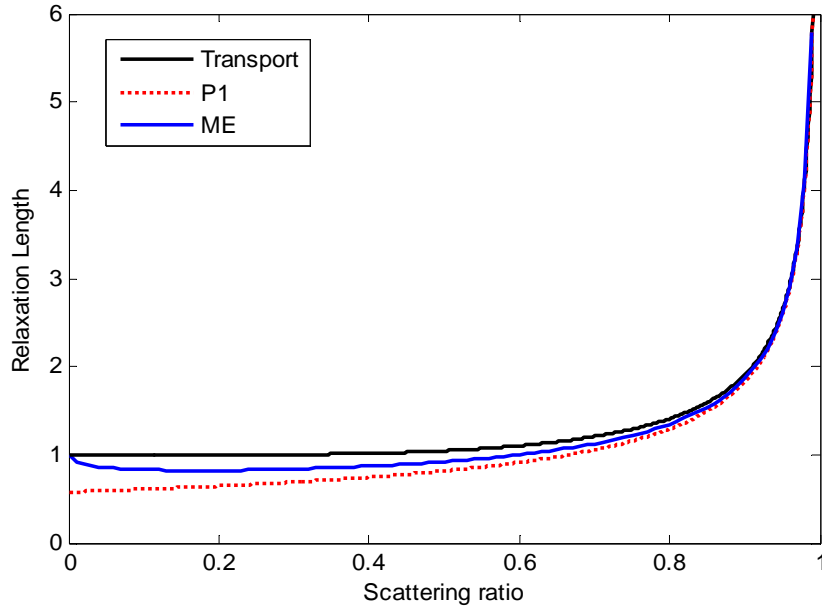


Figure 5.2 Plot of relaxation length for transport, P_1 and ME closures

The existence of a minimum in the relaxation length implies discontinuous solutions. To see this use Eq. (5.25) and (5.26) to get

$$\frac{\chi(\eta)}{\eta} = -v \quad (5.27)$$

Differentiate both sides of Eq. (5.27) and apply the chain rule, yielding

$$-\frac{dv}{dc} = \frac{d}{dc} \left[\frac{\chi(\eta)}{\eta} \right] = \frac{d}{d\eta} \left[\frac{\chi(\eta)}{\eta} \right] \frac{d\eta}{dc} \quad (5.28)$$

By Eq. (5.26), Eq (5.28) can be rewritten as

$$-\frac{dv}{dc} = \frac{d}{d\eta} \left[\frac{\chi(\eta)}{\eta} \right] \left[\frac{dv}{dc} (c-1) + v \right] \quad (5.29)$$

From Eq. (5.29) we can find if $\frac{dv}{dc} = 0$, then $\frac{d}{d\eta} \left[\frac{\chi(\eta)}{\eta} \right] = 0$. This indicates that the existence of a minimum in the relaxation length means the existence of a minimum in $\chi(\eta)/\eta$. This latter minimum implies that a discontinuity is possible in the scalar flux of the variable Eddington factor model. We can see an example of this property by comparing the plot of ME shown in Figure 5.1 and 5.2.

Also, note that the solution of ME becomes exact as $c \rightarrow 0$ in Figure 5.2; this is because the closure approximation becomes a mono-directional beam as $c \rightarrow 0$ (no scattering) and can exactly reproduce a mono-directional solution of the transport equation.

Eigenstructures

Further insights into Eddington factors are gained by considering the eigenvalues of the Jacobian. The two moment closure equations in the absence of sources and interactions can be written as,

$$\frac{\partial u}{\partial t} + \frac{\partial F(u)}{\partial x} = 0 \quad (5.30)$$

where $u = [\phi \quad J]^T$, $F(u) = [J \quad \chi\phi]^T$. Eq. (5.30) can be cast into a quasi-linear form

$$\frac{\partial u}{\partial t} + A(u) \frac{\partial u}{\partial x} = 0 \quad (5.31)$$

where $A(u) = \partial F / \partial u$ is the Jacobian of $F(u)$ and it is obtained with

$$A = \begin{bmatrix} 0 & 1 \\ \chi - \eta \frac{d\chi}{d\eta} & \frac{d\chi}{d\eta} \end{bmatrix} \quad (5.32)$$

Then the eigenvalues of this are obtained by

$$\lambda_{\pm} = \frac{1}{2} \frac{d\chi}{d\eta} \pm \frac{1}{2} \sqrt{\left(\frac{d\chi}{d\eta}\right)^2 + 4\left(\chi - \eta \frac{d\chi}{d\eta}\right)} \quad (5.33)$$

These eigenvalues give the speed at which information propagates through the system.

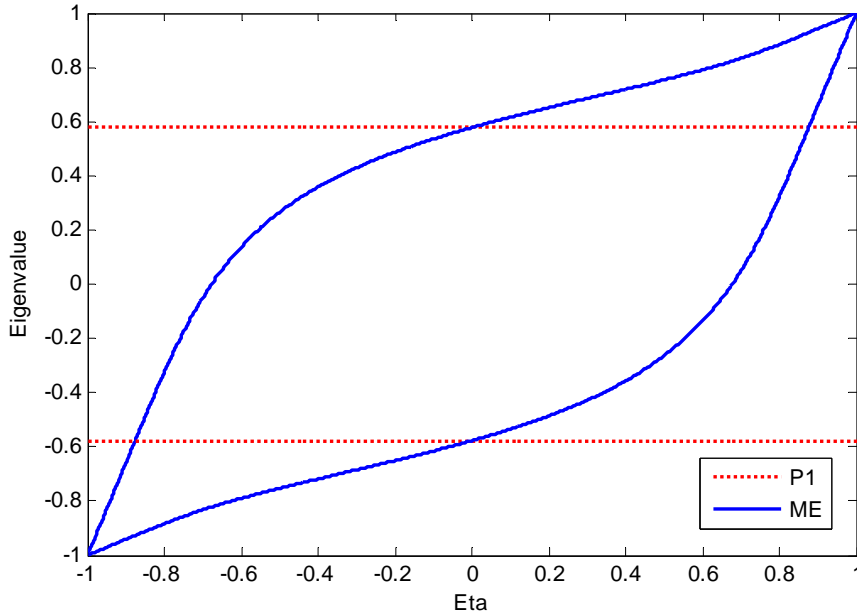


Figure 5.3 Plot of the eigenvalues of the Jacobian

Figure 5.3 shows the plot of the eigenvalues of the Jacobian for the P_1 and ME closures. Since P_1 is linear and has an Eddington factor of $1/3$ the eigenvalues of the Jacobian for P_1 are constant and equal to $\pm 1/\sqrt{3}$. But the ME Eddington factor is nonlinear and the eigenvalues vary with respect to η . We can see for isotropic flow ($\eta \rightarrow 0$), the eigenvalues of the ME are the same as those of P_1 . Also, for beam flow ($\eta \rightarrow \pm 1$), the two eigenvalues become ± 1 which implies the flow is mono-directional beam.

5.4 Construction of Nonlinear P_1 Eddington factor

We would like to develop a nonlinear Eddington factor to give up the linearity in the P_1 equation and hopefully get a positive scalar flux in 2D. The ME Eddington factor has been applied for time-dependent transport by Brunner in 1D. But it is known to be too expensive to obtain a Jacobian and eigenvalues with the ME Eddington tensor in 2D because of the variable b in Eq. (5.18).

In the previous section, we have found that a nonlinear Eddington factor should have two properties

$$\chi(\eta) \xrightarrow{\eta \rightarrow 0} \frac{1}{3} \quad (5.34)$$

$$\chi(\eta) \xrightarrow{\eta \rightarrow \pm 1} 1 \quad (5.35)$$

and the ME Eddington factor has an eigenstructure that models both the isotropic and the beam behavior

$$\lambda(\eta) \xrightarrow{\eta \rightarrow 0} \pm \frac{1}{\sqrt{3}} \quad (5.36)$$

$$\lambda(\eta) \xrightarrow{\eta \rightarrow 1} 1 \quad (5.37)$$

We therefore construct a new Eddington factor from the eigenstructure of ME. Assume $\chi(\eta)$ can be expressed in terms of an unknown function $G(\eta)$,

$$\chi(\eta) = \frac{1}{3} + G(\eta) \quad (5.38)$$

with
$$G(\eta) = G(-\eta) \quad (5.39)$$

where Eq. (5.39) is chosen to satisfy that the Eddington factor should consist of only even orders of η .

From the eigenvalues for the two moment closures shown in Eq. (5.33), we have

$$\lambda(\eta) = \frac{1}{2} \frac{dG}{d\eta} \pm \frac{1}{2} \sqrt{\left(\frac{dG}{d\eta}\right)^2 + 4\left(\frac{1}{3} + G(\eta) - \eta \frac{dG}{d\eta}\right)} \quad (5.40)$$

In order to satisfy Eq. (5.36) and (5.37)

$$\lambda(0) = \frac{1}{2} \frac{dG(0)}{d\eta} \pm \frac{1}{2} \sqrt{\left(\frac{dG(0)}{d\eta}\right)^2 + 4\left(\frac{1}{3} + G(0) - \eta \frac{dG(0)}{d\eta}\right)} = \pm \frac{1}{\sqrt{3}} \quad (5.41)$$

$$\lambda(1) = \frac{1}{2} \frac{dG(1)}{d\eta} \pm \frac{1}{2} \sqrt{\left(\frac{dG(1)}{d\eta}\right)^2 + 4\left(\frac{1}{3} + G(1) - \eta \frac{dG(1)}{d\eta}\right)} = 1 \quad (5.42)$$

From Eq. (5.41) - (5.42) we get three conditions

$$\frac{dG(0)}{d\eta} = 0 \quad (5.44)$$

$$\frac{dG(1)}{d\eta} = 2, \quad G(1) = \frac{2}{3} \quad (5.45)$$

Define $G(\eta)$ as the simplest explicit polynomial of even orders of η

$$G(\eta) = a + b\eta^2 + c\eta^4 \quad (5.46)$$

Substituting Eq. (5.45)-(5.45) into Eq. (5.46) yields

$$a = 0, \quad b = \frac{1}{3}, \quad c = \frac{1}{3} \quad (5.47)$$

Hence we get $G(\eta)$ as

$$G(\eta) = \frac{1}{3}\eta^2 + \frac{1}{3}\eta^4 \quad (5.48)$$

Finally, we obtain a new nonlinear Eddington factor by substituting Eq. (5.48) into Eq. (5.38) as

$$\chi(\eta) = \frac{1}{3} + \frac{1}{3}\eta^2 + \frac{1}{3}\eta^4 \quad (5.49)$$

We call Eq. (5.49) the Nonlinear P_1 (NLP_1) Eddington factor. Also we will be able to obtain a 3D NLP_1 Eddington tensor by substituting Eq. (5.49) into Eq. (5.15).

5.5 Comparisons of various Eddington factors

So far we have considered the P_1 , ME and NLP_1 Eddington factors. Here we introduce several closures for comparison of the various discontinuity structures, relaxation lengths and the eigenstructures in 1D.

Kershaw suggested a simple closure [5] and it has a very similar form with our NLP_1 Eddington factor.

$$\chi(\eta) = \frac{1}{3}(1 + 2\eta^2) \quad (5.50)$$

Besides the ME Eddington factor, Minerbo proposed also [3]

$$\chi(\eta) = \begin{cases} \frac{1}{3} & \text{if } 0 \leq \eta \leq \frac{1}{3} \\ \frac{1}{2}(1-\eta)^2 + \eta^2 & \text{if } \frac{1}{3} \leq \eta \leq 1 \end{cases} \quad (5.51)$$

Levermore derived a closure [6] with the hypothesis that there exists a reference frame in which the photon distribution function is isotropic.

$$\chi(\eta) = \frac{5}{3} - \frac{2}{3}\sqrt{4 - 3\eta^2} \quad (5.52)$$

Figure 5.4 shows a comparison of the discontinuity structures for the P_1 , ME, Kershaw Minerbo, Levermore and NLP_1 closures. All of the nonlinear closures have a minimum and so can have discontinuous solutions. But such a minimum does not occur for the linear P_1 closure. The minimum of Kershaw is closest to 1 and shallowest ($\chi/\eta = 0.942$). The minimum of NLP_1 is closer to 1 than the one in the Minerbo closure, implying that the NLP_1 closure has a smaller range of discontinuity than the Minerbo closure. Also, NLP_1 is very close to the ME solution.

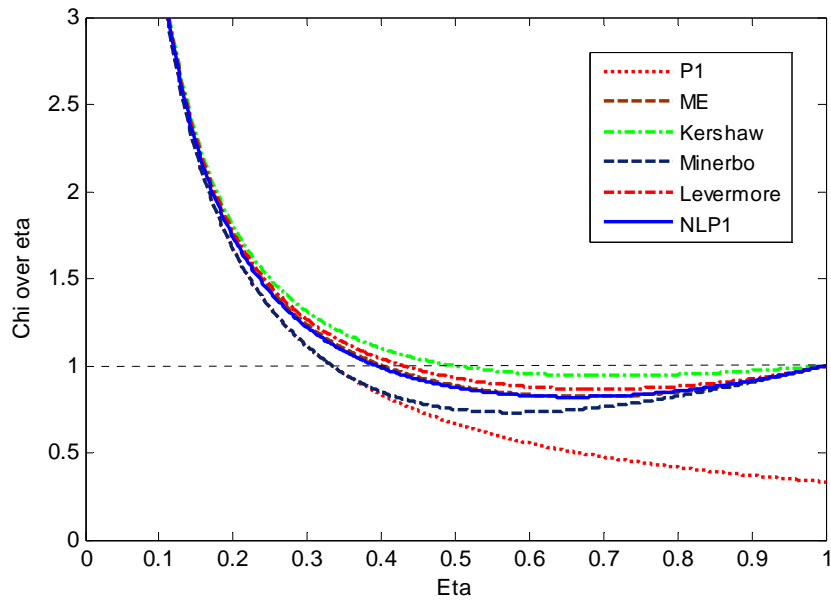


Figure 5.4 Comparison of χ/η as a function of η for P_1 , ME, Kershaw, Minerbo, Levermore and NLP_1

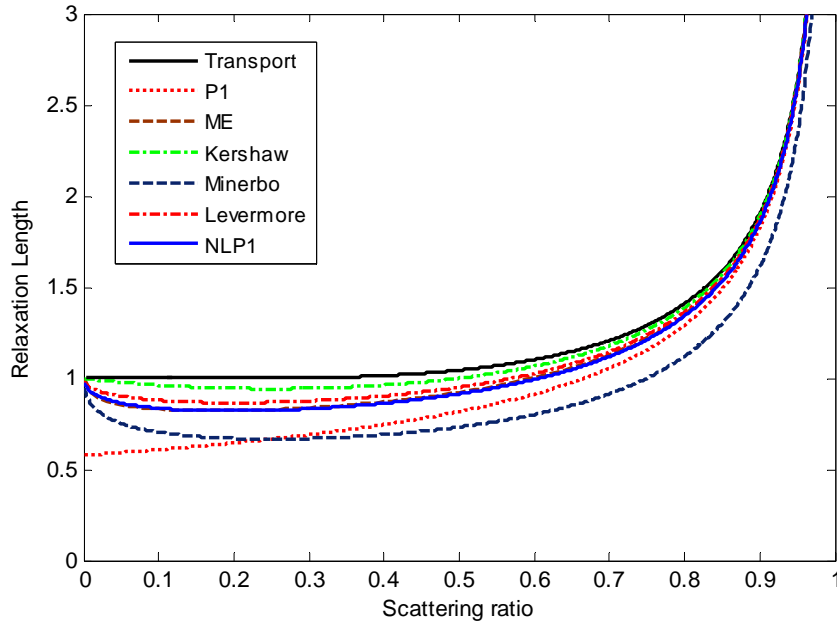


Figure 5.5 Comparison of relaxation length as a function of scattering ratio for transport, P_1 , ME, Kershaw, Levermore and NLP_1

Figure 5.5 shows a comparison of relaxation length as a function of scattering ratio for transport, P_1 , ME, Kershaw, Minerbo, Levermore and NLP_1 . The relaxation length of the NLP_1 is closer to the transport than the Minerbo and the linear P_1 but it is less close than the Kershaw or Levermore. The NLP_1 solution is again very close to the ME solution. We can see that the nonlinear closures, which all show a minimum in χ/η in Figure 5.4, also have a minimum in the relaxation length in Figure 5.5.

Figure 5.6 shows a comparison of the eigenvalues of the Jacobian for P_1 , ME, Kershaw, Levermore and NLP_1 . We can see that all the nonlinear closures have critical values of η at which the eigenvalues transition from having opposite sign to having the same sign, implying the flow changes from bi-directional to mono directional. Note that the Kershaw solution does not become ± 1 at $\eta = \pm 1$ and is unique among the nonlinear closures in this respect. Also note that the eigenvalue varies linearly in the Minerbo

closure. The NLP_1 solution is again very close to the ME solution. The linear P_1 has constant eigenvalues for all η .

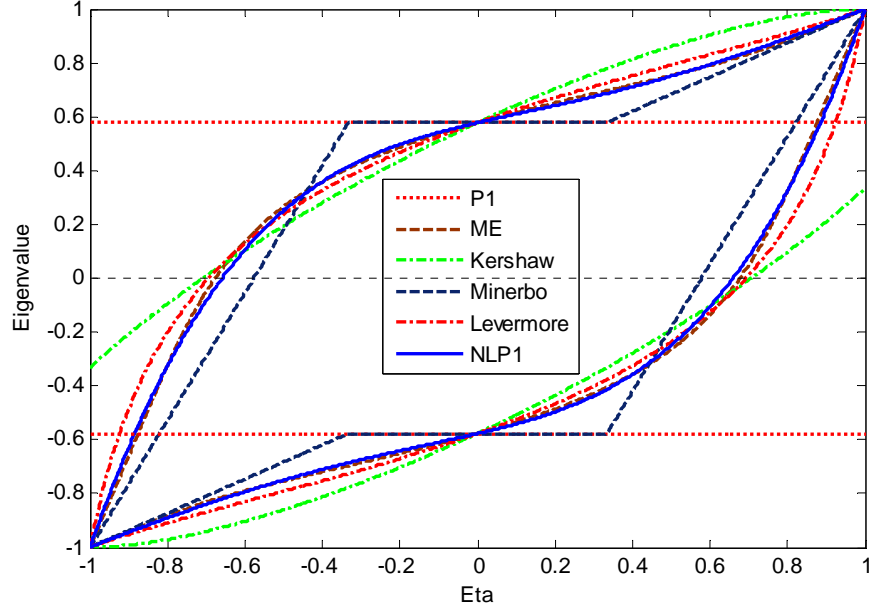


Figure 5.6 Comparison of the eigenvalues of the Jacobian for P_1 , ME, Kershaw, Minerbo, Levermore and NLP_1

5.6 Riemann Solver Construction for the NLP_1 Closure

From Eq. (5.15) and (5.49), NLP_1 in 3D can be written as

$$\frac{\partial}{\partial t} \begin{bmatrix} \phi \\ \mathbf{J} \end{bmatrix} + \nabla \cdot \begin{bmatrix} \mathbf{J} \\ \chi \phi \end{bmatrix} = - \begin{bmatrix} \Sigma_a \phi \\ \Sigma_t \mathbf{J} \end{bmatrix} + \begin{bmatrix} Q_0 \\ \mathbf{Q}_1 \end{bmatrix} \quad (5.53)$$

with

$$\chi = \frac{1}{2}(1 - \chi) \mathbf{I} + \frac{1}{2}(3\chi - 1) \frac{\mathbf{J} \otimes \mathbf{J}}{J^2} \quad (5.54)$$

where

$$\chi(\eta) = \frac{1}{3} + \frac{1}{3}\eta^2 + \frac{1}{3}\eta^4 \quad (5.55)$$

The streaming term is simplified as

$$\nabla \cdot \boldsymbol{\chi} \phi = \nabla \cdot \frac{(1-\chi)\phi}{2} + \nabla \cdot \frac{\mathbf{J} \otimes \mathbf{J}}{J^2} \frac{(3\chi-1)\phi}{2} \quad (5.56)$$

Expanding

$$\frac{\mathbf{J} \otimes \mathbf{J}}{J^2} = \frac{1}{J_x^2 + J_y^2 + J_z^2} \begin{bmatrix} J_x J_x & J_x J_y & J_x J_z \\ J_y J_x & J_y J_y & J_y J_z \\ J_z J_x & J_z J_y & J_z J_z \end{bmatrix} \quad (5.57)$$

and using Eq. (5.57) we expand Eq. (5.53) as,

$$\frac{\partial}{\partial t} \begin{bmatrix} \phi \\ J_x \\ J_y \\ J_z \end{bmatrix} + \frac{\partial}{\partial x} \begin{bmatrix} J_x \\ a + (J_x^2/J^2)c \\ (J_y J_x/J^2)c \\ (J_x J_z/J^2)c \end{bmatrix} + \frac{\partial}{\partial y} \begin{bmatrix} J_y \\ (J_x J_y/J^2)c \\ a + (J_y^2/J^2)c \\ (J_y J_z/J^2)c \end{bmatrix} + \frac{\partial}{\partial z} \begin{bmatrix} J_z \\ (J_x J_z/J^2)c \\ (J_y J_z/J^2)c \\ a + (J_z^2/J^2)c \end{bmatrix} = - \begin{bmatrix} \sum_a \phi \\ \sum_t J_x \\ \sum_t J_y \\ \sum_t J_z \end{bmatrix} + \begin{bmatrix} Q_0 \\ Q_x \\ Q_y \\ Q_z \end{bmatrix} \quad (5.58)$$

where
$$a = \frac{(1-\chi)\phi}{2}, \quad c = \frac{(3\chi-1)\phi}{2} \quad (5.59)$$

By dropping the z dependent term in Eq. (5.58) we get the 2D form

$$\frac{\partial}{\partial t} \begin{bmatrix} \phi \\ J_x \\ J_y \end{bmatrix} + \frac{\partial}{\partial x} \begin{bmatrix} J_x \\ a + (J_x^2/J^2)c \\ (J_y J_x/J^2)c \end{bmatrix} + \frac{\partial}{\partial y} \begin{bmatrix} J_y \\ (J_x J_y/J^2)c \\ a + (J_y^2/J^2)c \end{bmatrix} = - \begin{bmatrix} \sum_a \phi \\ \sum_t J_x \\ \sum_t J_y \end{bmatrix} + \begin{bmatrix} Q_0 \\ Q_x \\ Q_y \end{bmatrix} \quad (5.60)$$

Eq. (5.60) can be written as,

$$\frac{\partial u}{\partial t} + \frac{\partial F_x(u)}{\partial x} + \frac{\partial F_y(u)}{\partial y} = S(u) \quad (5.61)$$

where

$$u = \begin{bmatrix} \phi \\ J_x \\ J_y \end{bmatrix}, \quad F_x(u) = \begin{bmatrix} J_x \\ a + (J_x^2 / J^2)c \\ (J_y J_x / J^2)c \end{bmatrix}, \quad F_y(u) = \begin{bmatrix} J_y \\ (J_x J_y / J^2)c \\ a + (J_y^2 / J^2)c \end{bmatrix}, \quad S(u) = - \begin{bmatrix} \sum_a \phi \\ \sum_t J_x \\ \sum_t J_y \end{bmatrix} + \begin{bmatrix} Q_0 \\ Q_x \\ Q_y \end{bmatrix} \quad (5.62)$$

Also Eq. (5.61) can be cast into a quasilinear form

$$\frac{\partial u}{\partial t} + A_x(u) \frac{\partial u}{\partial x} + A_y(u) \frac{\partial u}{\partial y} = S(u) \quad (5.63)$$

where $A_x(u) = \partial F_x / \partial u$ and $A_y(u) = \partial F_y / \partial u$ are the Jacobians of F_x and F_y . They are

$$A_x = \begin{bmatrix} 0 & 1 & 0 \\ \frac{1}{3} - \eta_x^4 + \frac{1}{2} \eta_y^4 - \frac{1}{2} \eta_x^2 \eta_y^2 - \frac{1}{3} \eta_x^2 + \frac{1}{6} \eta_y^2 & \frac{4}{3} \eta_x^3 + \frac{1}{3} \eta_x \eta_y^2 + \frac{2}{3} \eta_x & \frac{1}{3} \eta_x^2 \eta_y - \frac{2}{3} \eta_x^3 - \frac{1}{3} \eta_y \\ -\frac{3}{2} \eta_x^3 \eta_y - \frac{3}{2} \eta_x \eta_y^3 - \frac{1}{2} \eta_x \eta_y & \frac{3}{2} \eta_x^2 \eta_y + \frac{1}{2} \eta_y^3 + \frac{1}{2} \eta_y & \frac{1}{2} \eta_x^3 + \frac{3}{2} \eta_x \eta_y^2 + \frac{1}{2} \eta_x \end{bmatrix} \quad (5.64)$$

$$A_y = \begin{bmatrix} 0 & 0 & 1 \\ -\frac{3}{2} \eta_x^3 \eta_y - \frac{3}{2} \eta_x \eta_y^3 - \frac{1}{2} \eta_x \eta_y & \frac{3}{2} \eta_x^2 \eta_y + \frac{1}{2} \eta_y^3 + \frac{1}{2} \eta_y & \frac{1}{2} \eta_x^3 + \frac{3}{2} \eta_x \eta_y^2 + \frac{1}{2} \eta_x \\ \frac{1}{3} - \eta_y^4 + \frac{1}{2} \eta_x^4 - \frac{1}{2} \eta_x^2 \eta_y^2 - \frac{1}{3} \eta_y^2 + \frac{1}{6} \eta_x^2 & \frac{1}{3} \eta_x \eta_y^2 - \frac{2}{3} \eta_x^3 - \frac{1}{3} \eta_x & \frac{4}{3} \eta_y^3 + \frac{1}{3} \eta_x^2 \eta_y + \frac{2}{3} \eta_y \end{bmatrix} \quad (5.65)$$

By Eq. (3.22) - (3.25) in chapter 3, the right and the left cell interface fluxes in 2D can be written as

$$F_{x,i+1/2,j} = \frac{1}{2} [F_x(u_{i+1,j}) + F_x(u_{i,j})] - \frac{1}{2} |R_x(u_{i+1,j}, u_{i,j})| (u_{i+1,j} - u_{i,j}) \quad (5.66)$$

$$F_{x,i-1/2,j} = \frac{1}{2} [F_x(u_{i,j}) + F_x(u_{i-1,j})] - \frac{1}{2} |R_x(u_{i,j}, u_{i-1,j})| (u_{i,j} - u_{i-1,j}) \quad (5.67)$$

$$F_{y,i,j+1/2} = \frac{1}{2} [F_y(u_{i,j+1}) + F_y(u_{i,j})] - \frac{1}{2} |R_y(u_{i,j+1}, u_{i,j})| (u_{i,j+1} - u_{i,j}) \quad (5.68)$$

$$F_{y,i,j-1/2} = \frac{1}{2} [F_y(u_{i,j}) + F_y(u_{i,j-1})] - \frac{1}{2} |R_y(u_{i,j}, u_{i,j-1})| (u_{i,j} - u_{i,j-1}) \quad (5.69)$$

where, since the Roe matrix becomes the Jacobian as u_L and u_R approach each other (see the Roe property **R2** in Section 3.4), so we approximate Roe matrix in Eq. (5.66) - (5.69) as

$$R_x(u_{i+1/2,j}, u_{i,j}) \approx A_{x\ i+1/2,j} \left(\eta_x = \frac{(J_{x\ i+1,j} + J_{x\ i,j})/2}{(\phi_{i+1,j} + \phi_{i,j})/2}, \eta_y = \frac{(J_{y\ i+1,j} + J_{y\ i,j})/2}{(\phi_{i+1,j} + \phi_{i,j})/2} \right) \quad (5.70)$$

$$R_x(u_{i,j}, u_{i-1/2,j}) \approx A_{x\ i-1/2,j} \left(\eta_x = \frac{(J_{x\ i,j} + J_{x\ i-1,j})/2}{(\phi_{i,j} + \phi_{i-1,j})/2}, \eta_y = \frac{(J_{y\ i,j} + J_{y\ i-1,j})/2}{(\phi_{i,j} + \phi_{i-1,j})/2} \right) \quad (5.71)$$

$$R_y(u_{i,j+1/2}, u_{i,j}) \approx A_{y\ i,j+1/2} \left(\eta_x = \frac{(J_{x\ i,j+1} + J_{x\ i,j})/2}{(\phi_{i,j+1} + \phi_{i,j})/2}, \eta_y = \frac{(J_{y\ i,j+1} + J_{y\ i,j})/2}{(\phi_{i,j+1} + \phi_{i,j})/2} \right) \quad (5.72)$$

$$R_y(u_{i,j}, u_{i,j-1/2}) \approx A_{y\ i-1/2,j} \left(\eta_x = \frac{(J_{x\ i,j} + J_{x\ i,j-1})/2}{(\phi_{i,j} + \phi_{i,j-1})/2}, \eta_y = \frac{(J_{y\ i,j} + J_{y\ i,j-1})/2}{(\phi_{i,j} + \phi_{i,j-1})/2} \right) \quad (5.73)$$

To get a cell average flux, operate $\frac{1}{\Delta x \Delta y} \int_{x_{i,j-1/2}}^{x_{i,j+1/2}} \int_{y_{i-1/2,j}}^{y_{i+1/2,j}} (\cdot) dy dx$ on Eq. (5.61) yielding

$$\frac{\partial u_{i,j}}{\partial t} + \frac{1}{\Delta x} (F_{x,i+1/2,j} - F_{x,i-1/2,j}) + \frac{1}{\Delta y} (F_{y,i,j+1/2} - F_{y,i,j-1/2}) = S(u_{i,j}) \quad (5.74)$$

where

$$u_{i,j} = \frac{1}{\Delta x \Delta y} \int_{x_{i,j-1/2}}^{x_{i,j+1/2}} \int_{y_{i-1/2,j}}^{y_{i+1/2,j}} u(x,t) dy dx \quad (5.75)$$

Applying Eq. (5.70)-(5.73) into Eq. (5.74) yields

$$\begin{aligned} \frac{\partial u_{i,j}}{\partial t} + \frac{1}{2\Delta x} [F_x(u_{i+1,j}) - F_x(u_{i-1,j})] - \frac{1}{2\Delta x} |R_x(u_{i+1,j}, u_{i,j})| (u_{i+1,j} - u_{i,j}) + \frac{1}{2\Delta x} |R_x(u_{i,j}, u_{i-1,j})| (u_{i,j} - u_{i-1,j}) \\ + \frac{1}{2\Delta y} [F_y(u_{i,j+1}) - F_y(u_{i,j-1})] - \frac{1}{2\Delta y} |R_y(u_{i,j+1}, u_{i,j})| (u_{i,j+1} - u_{i,j}) + \frac{1}{2\Delta y} |R_y(u_{i,j}, u_{i,j-1})| (u_{i,j} - u_{i,j-1}) = S(u_{i,j}) \end{aligned} \quad (5.76)$$

Finally, forward Euler time discretizing on Eq. (5.76) yields,

$$\begin{aligned}
u_{i,j}^{n+1} = u_{i,j}^n - \frac{\Delta t}{2\Delta x} [F_x(u_{i+1,j}^n) - F_x(u_{i-1,j}^n)] + \frac{\Delta t}{2\Delta x} |R_x(u_{i+1,j}^n, u_{i,j}^n)| (u_{i+1,j}^n - u_{i,j}^n) - \frac{\Delta t}{2\Delta x} |R_x(u_{i,j}^n, u_{i-1,j}^n)| (u_{i,j}^n - u_{i-1,j}^n) \\
- \frac{\Delta t}{2\Delta y} [F_y(u_{i,j+1}^n) - F_y(u_{i,j-1}^n)] + \frac{\Delta t}{2\Delta y} |R_y(u_{i,j+1}^n, u_{i,j}^n)| (u_{i,j+1}^n - u_{i,j}^n) - \frac{\Delta t}{2\Delta y} |R_y(u_{i,j}^n, u_{i,j-1}^n)| (u_{i,j}^n - u_{i,j-1}^n) + S(u_{i,j}^n)
\end{aligned} \tag{5.77}$$

where u_i^n is the cell averaged value in the i^{th} cell at the n^{th} time step and Δt indicates the time step size.

The 1D form of NLP_1 closure can be obtained from Eq. (5.77) by dropping x or y - dependent terms as,

$$u_i^{n+1} = u_i^n - \frac{\Delta t}{2\Delta x} [F(u_{i+1}^n) - F(u_{i-1}^n)] + \frac{\Delta t}{2\Delta x} [|R(u_{i+1}^n, u_i^n)| (u_{i+1}^n - u_i^n) - |R(u_i^n, u_{i-1}^n)| (u_i^n - u_{i-1}^n)] + S(u_i^n) \tag{5.78}$$

5.7 Numerical results in 1D

The one-dimensional time-dependent behavior of NLP_1 is tested by solving for the scalar flux in a slab. The slab size is 20 cm and consists of a purely scattering medium ($\Sigma_t = \Sigma_s = 1 \text{ cm}^{-1}$). A delta function initially isotropic flux is imposed at the center. The spatial grid and time steps initially are chosen as $\Delta x = 0.002$, $\Delta t = 0.001$ respectively.

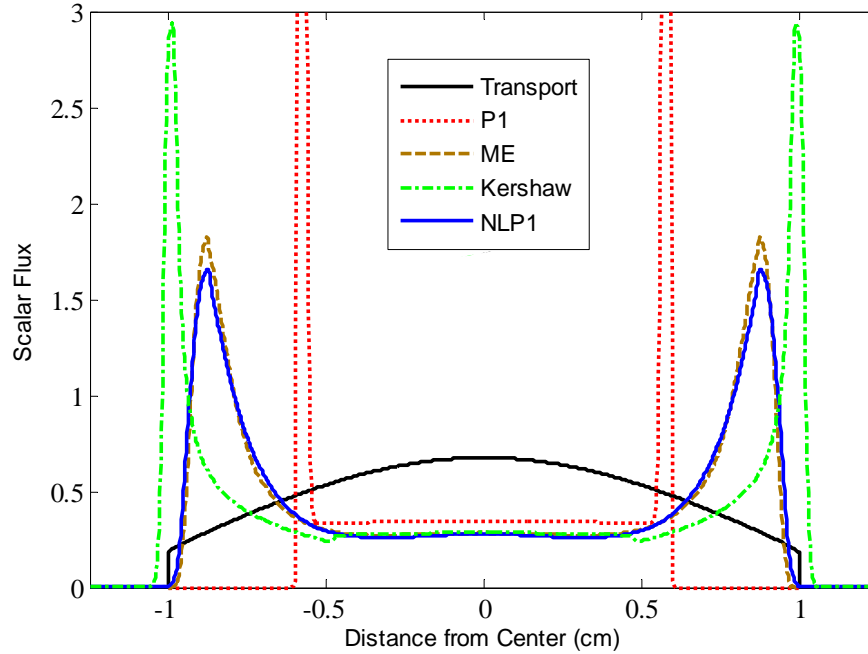


Figure 5.7 Comparison of transport, P_1 , ME, Kershaw and NLP_1 at 1 sec in 1D

Figure 5.7 shows comparison of transport, P_1 , ME, Kershaw and NLP_1 solution at one second after the pulse in 1D. The transport result is from Ganapol's analytic solution [7, 8]. The collided fluxes shown between the two wave fronts in ME, Kershaw and NLP_1 are very close to each other but underestimate the true collided flux. Note the characteristic speed of the Kershaw is faster than the NLP_1 and even faster than transport. Also the Kershaw closure produces discontinuities near $x = \pm 0.5 \text{ cm}$. However the NLP_1

and the ME have no discontinuity. Generally the NLP_1 result looks very close to the ME solution.

Figure 5.8 and 5.9 show solutions at 5 and 10 seconds for the same problem. The solutions have become closer to the transport solution as time goes on. The NLP_1 is closer to the transport solution than Kershaw near the center. Kershaw moves particles out towards the wave front too quickly, so it is low in the center compared with NLP_1 . We find NLP_1 solutions become very close to the ME results at later times.

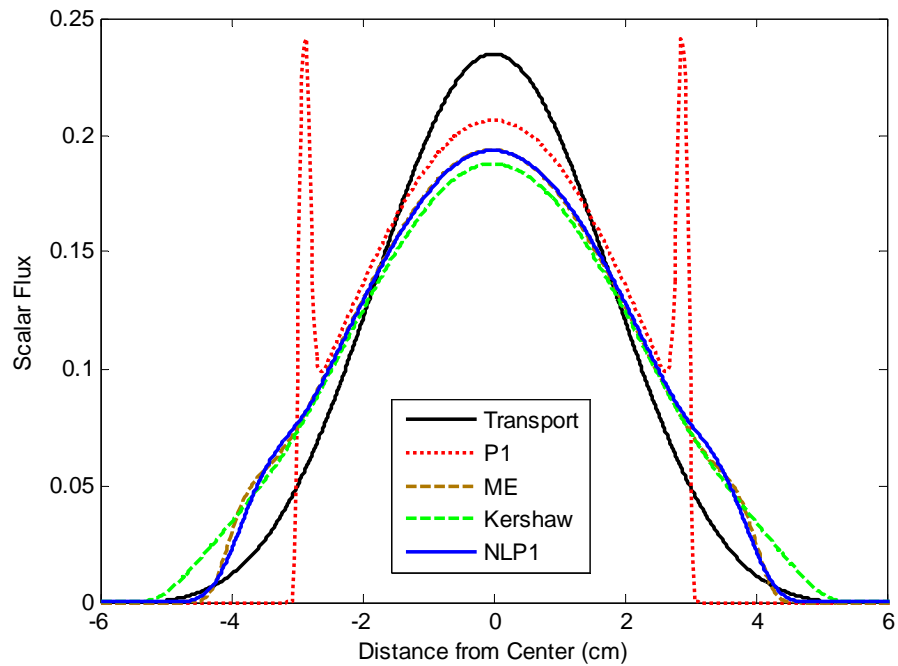


Figure 5.8 Comparison of Transport, P_1 , ME, Kershaw and NLP_1 at 5 sec in 1D

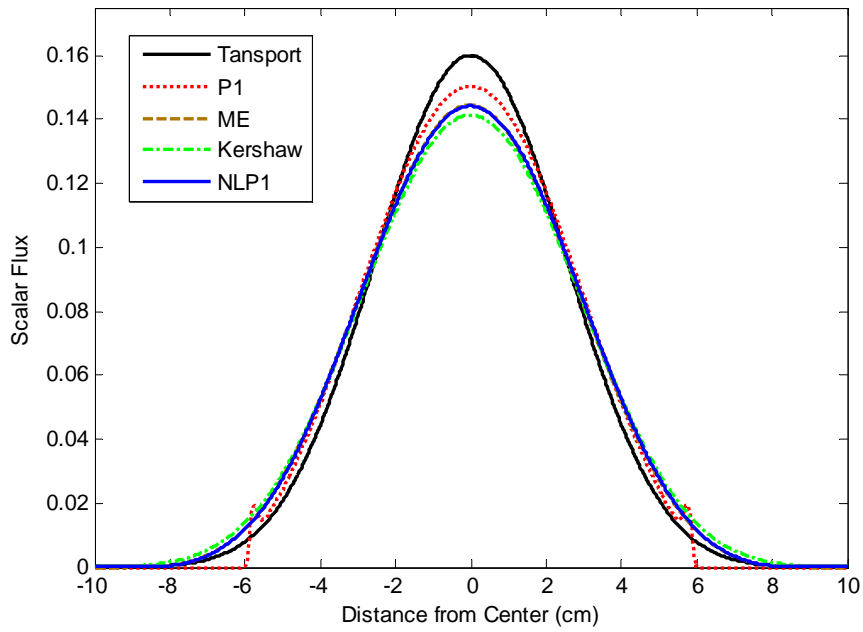


Figure 5.9 Comparison of Transport, P_1 , ME, Kershaw and NLP_1 at 10 sec in 1D

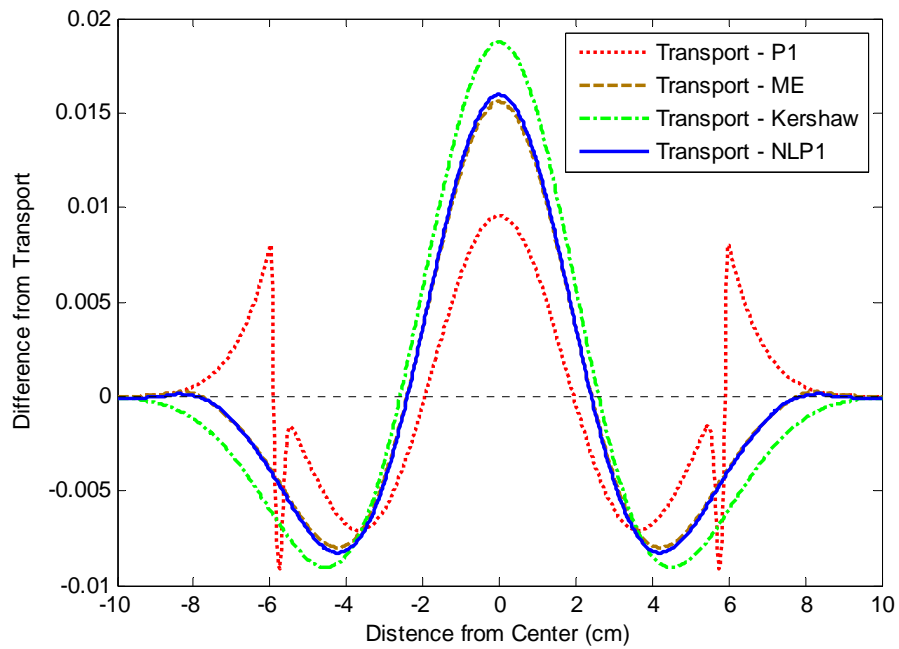


Figure 5.10 Plot of difference from Transport at 10 sec in 1D

Figure 5.10 shows a plot of the difference of the scalar flux from the transport solution at 10 seconds and Table 5.1 gives the results of the volume integral of the difference defined as

$$\mathbf{DI} = \int_{-10}^{10} |\phi_{tras.} - \phi| dx \quad (5.79)$$

where $\phi_{tras.}$ is the transport and ϕ is the solution computed by each method. From these results we can see that the NLP_1 has a smaller difference from the transport solution than Kershaw and is very close to ME. The P_1 error is the smallest but it gives a solution that is qualitatively incorrect.

Table 5.1 Volume integral of the difference of the scalar flux from transport

Time	P_1	ME	Kershaw	NLP_1
10 sec	0.074842	0.094152	0.127720	0.096582
5sec	0.132898	0.192696	0.223464	0.188402

5.8 Numerical results in 2D

Our two-dimensional test problem is implemented in a 20 cm square of purely scattering medium ($\Sigma_t = \Sigma_s = 1 \text{ cm}^{-1}$). A pulsed line source is imposed as an initial condition at the center of the system and vacuum boundary conditions are used.

Figure 5.11 shows a comparison of P_1 , Kershaw and NLP_1 at 0.2 sec after the pulse. The scalar flux of P_1 becomes negative, as does any P_N solution at early times. But we can see that Kershaw and NLP_1 do not become negative. Note the collided flux shown between the two wave fronts in NLP_1 approaches closer to 0 at the center than Kershaw does.

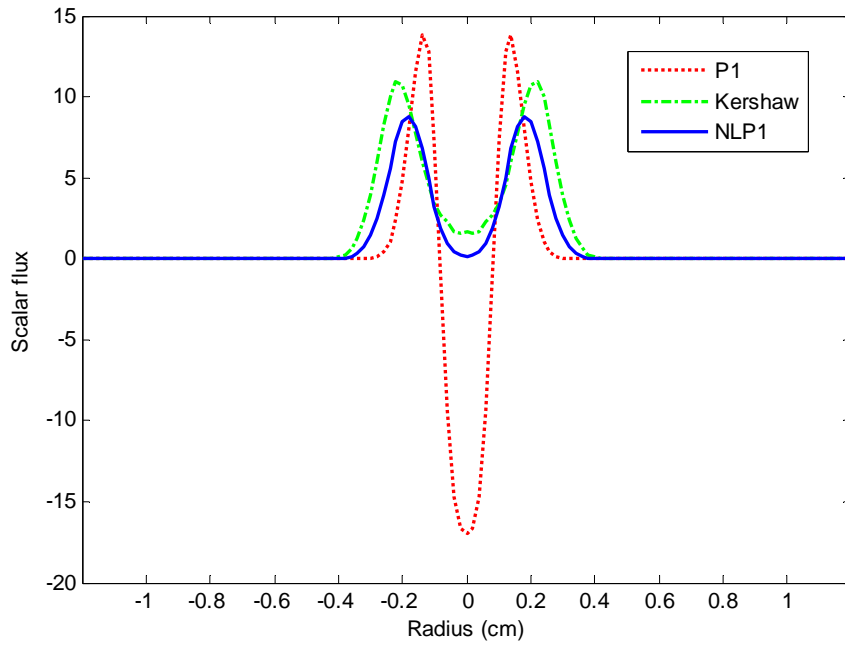


Figure 5.11 Comparison of P_1 , Kershaw and NLP_1 at 0.2 sec after the pulse in 2D

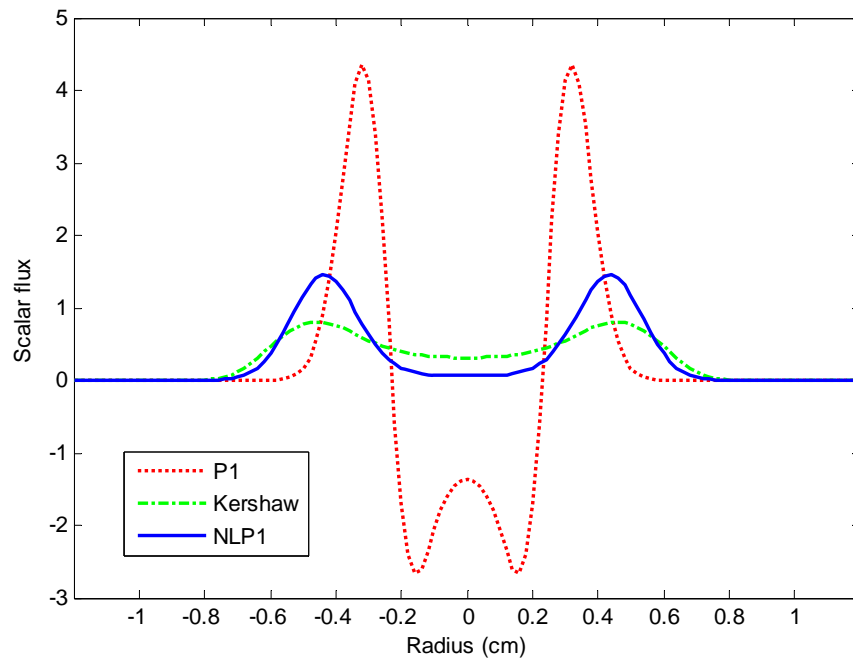


Figure 5.12 Comparison of P_1 , Kershaw and NLP_1 at 0.5 sec after the pulse in 2D

Figure 5.12 shows a comparison of solutions at 0.5 sec after the pulse under the same condition. The P_1 solution still shows negative scalar flux but the NLP_1 remains positive and propagates particles to the outside from the origin. Kershaw also remains positive.

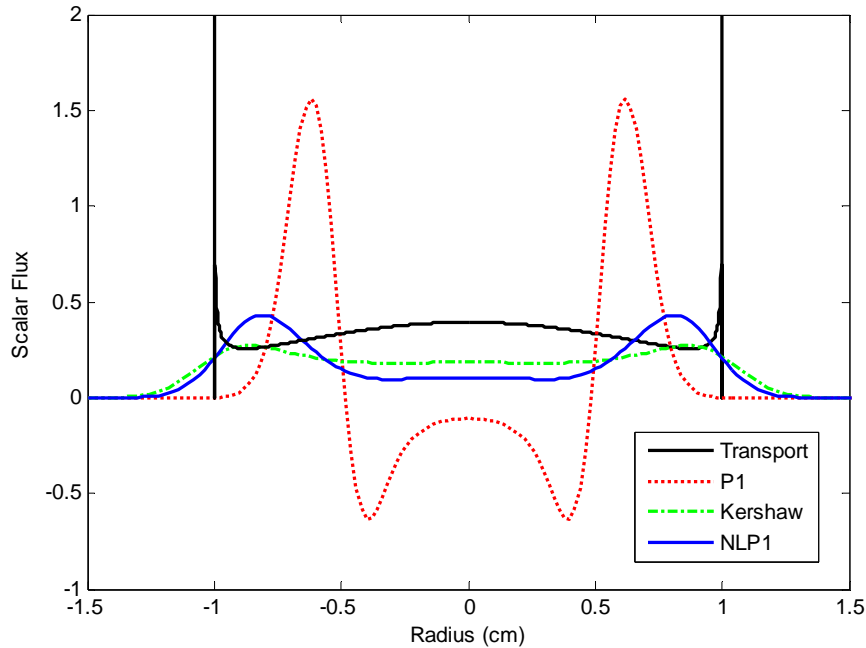
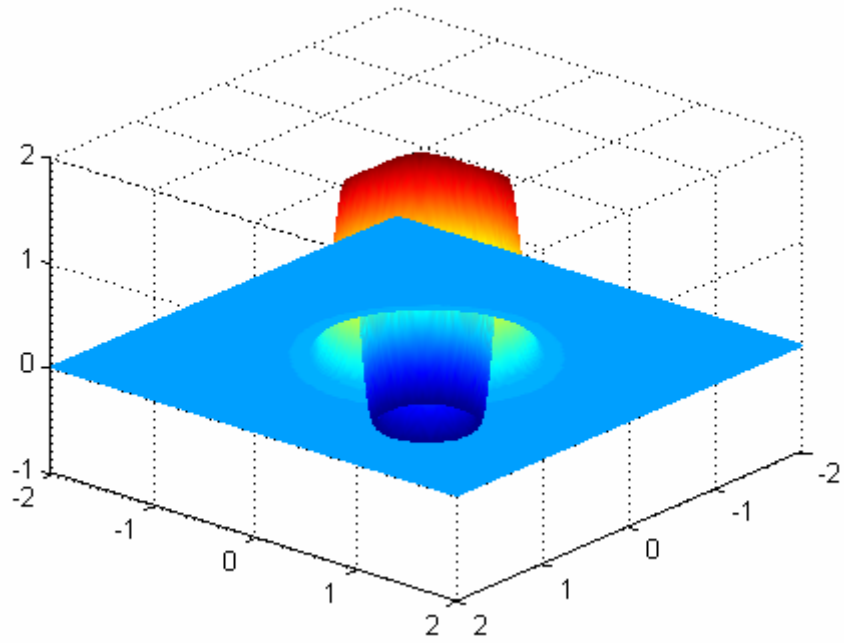


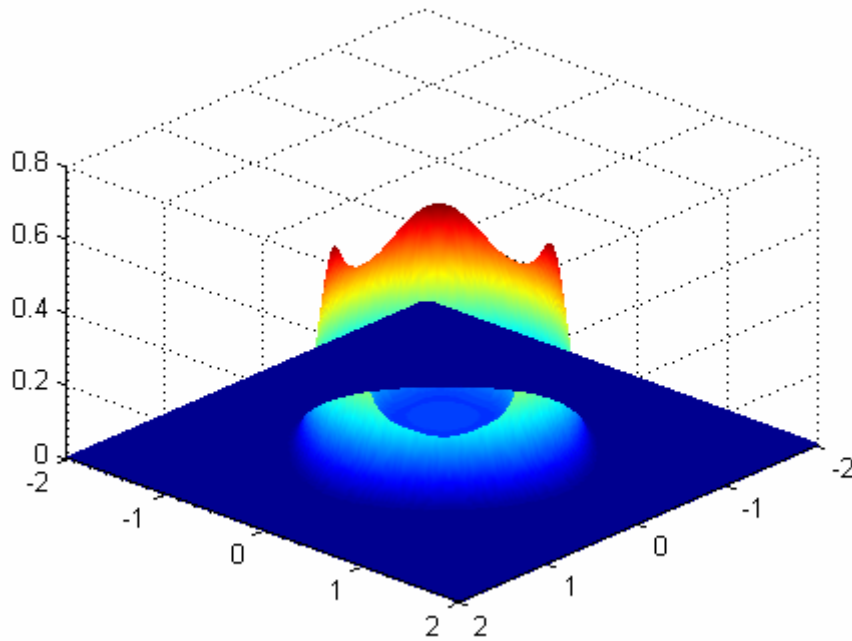
Figure 5.13 Comparison of P_1 , Kershaw and NLP_1 at 1 sec after the pulse in 2D

Figure 5.13 shows a comparison of transport, P_1 , Kershaw and NLP_1 at 1 sec after the pulse. P_1 remains negative but NLP_1 never becomes negative even at 1 sec and it does not have any negative dip behind the front of uncollided particles. Figure 5.14 is another view of the fluxes computed using P_1 and NLP_1 at 1 sec after the pulse.

Figure 5.15 and 5.16 show a comparison of solutions at 3 and 5 sec for the same problem respectively. At this time all solutions become positive. The P_1 solution still shows an oscillation. Near the center of the system Kershaw and NLP_1 solutions (nonlinear solution) are closer to the transport solution than P_1 solution.



(a) P_1



(b) NLP_1

Figure 5.14 Comparison of P_1 and NLP_1 at 1 sec after the pulse in 2D

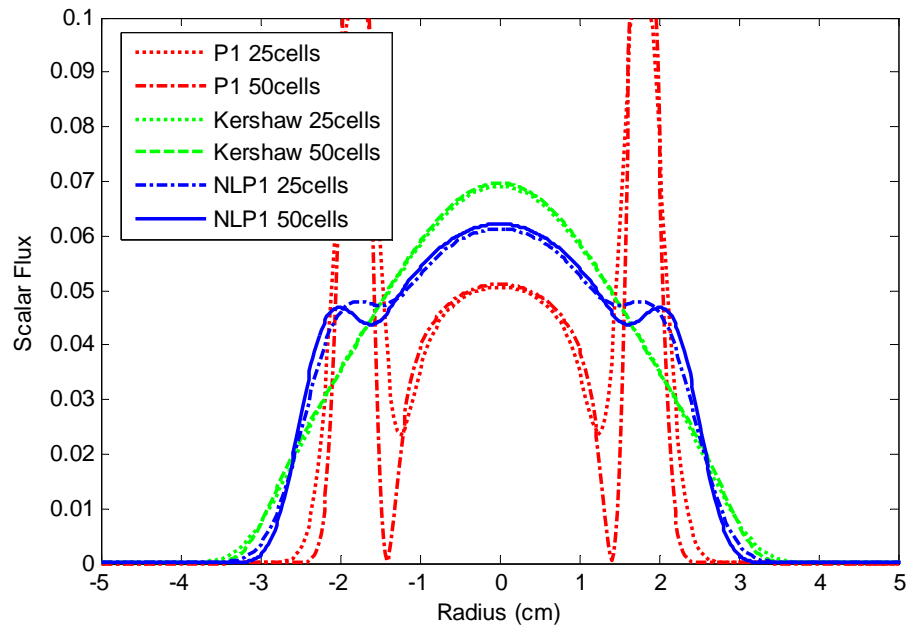


Figure 5.15 Comparison of P_1 , Kershaw and NLP_1 at 3 sec after the pulse in 2D

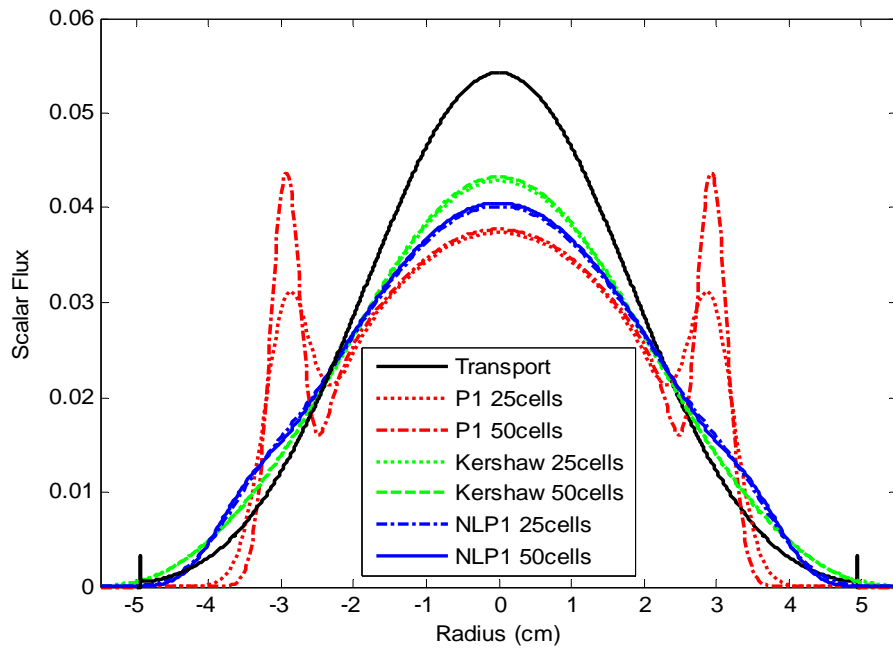


Figure 5.16 Comparison of transport, P_1 , Kershaw and NLP_1 at 5 sec after the pulse in 2D

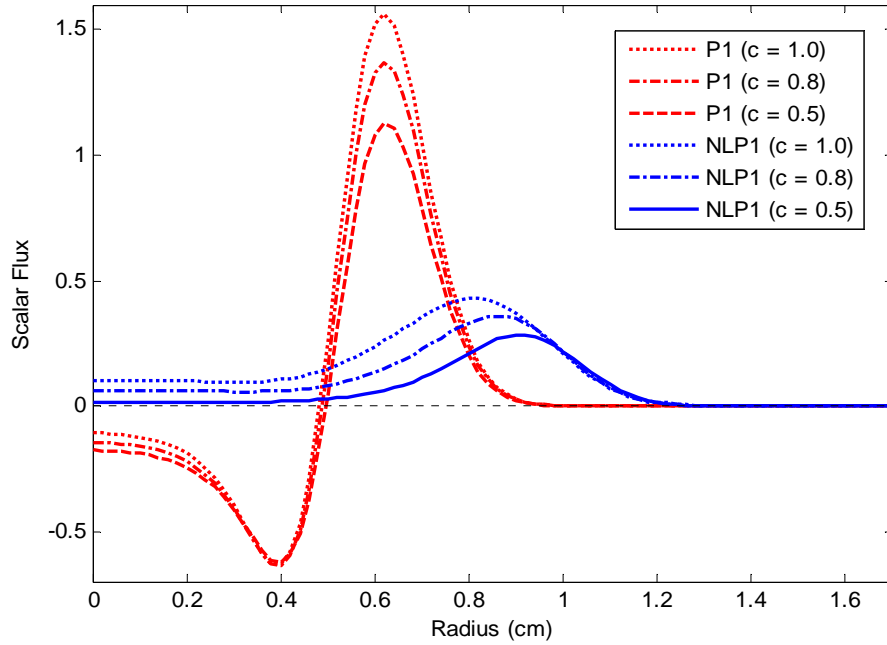


Figure 5.17 NLP_1 solutions for various scattering ratios at 1 second in 2D

Figure 5.17 shows the comparison of NLP_1 and P_1 for various scattering ratios ($c = \Sigma_s / \Sigma_t$) at 1 second in 2D. From this figure we can see that as the scattering ratio decreases the wave speed of NLP_1 approaches to 1. Early in the transient particles have not had much chance to scatter because they have traveled only one mean-free-path of distance. This means that as scattering ratio reduces, beam-like quality is retained further into the system and J_x approaches ϕ implying η_x goes to 1. As η_x goes to 1, the eigenvalues of NLP_1 approach 1 as shown in Figure 5.6. So we can understand that the wave speed of NLP_1 in Figure 5.17 approaches 1 as the scattering ratio decreases. In contrast, we can see the wave speeds of P_1 do not vary with the scattering ratio. This is because P_1 has constant eigenvalues as shown in Figure 5.3. Also we can see that the collided flux behind the wave front for NLP_1 decreases as the scattering ratio decreases due to increased absorption.

Test problem

So far all our results have been for a pulsed source problems. Here we present another test problem. That is propagation of radiation across a system. A schematic of this problem is shown in Figure 5.18. The system size is 6 by 6 cm. There is a 2 by 2 cm material block in the center and a source block is located at the bottom left corner of the system. The material block and source block have cross sections with $\Sigma_t = 1\text{cm}^{-1}$. The source block has scattering ratio $c = 1$, and the material block will have a variable c . The vacuum boundaries are assumed outside the system. The problem is implemented for different media; vacuum, absorber and scatterer. This problem is a good indicator of how our NLP_1 model in general will work on realistic problems with a material across various mediums compared to conventional P_1 approximation. All results are compared at $t = 10\text{sec}$.

Figure 5.19 shows the results for NLP_1 and P_1 across vacuum. P_1 does not give a good solution because it does not capture a shadow behind the material block. For NLP_1 it is unusual that the radiation comes out in two beams - one going along the x-axis and the other along the y-axis. One possible explanation for this is that in the vacuum J_x becomes ϕ , implying $\eta_x = 1$, because of no collisions and this results in the beam like behavior (similarly in the y-direction). Moreover we can see NLP_1 can capture the shadow behind the material block better than P_1 . Figure 5.20 is a line-out ($z = 0$) for NLP_1 and P_1 in Figure 5.19. Note the rapid flux changes for NLP_1 on left-hand side in the figure and for the flux for P_1 which looks like convex surface on the right-hand side. Either one is considered to be undesirable result.

Figure 5.21 to 5.23 show the results across medium from pure absorber to pure scatterer. We can see that as the scattering ratio increases, the propagation for NLP_1 becomes

isotropic like that for P_1 . This can be explained that in the medium, as collision increases, J_x approaches 0 implying $\eta_x = 0$ and then $\eta_x = 0$ indicates the flow is isotropic (similarly in the y -direction). For this reason, the pure scatterer for NLP_1 displays isotropic propagation like P_1 .

Figure 5.24 shows the results for NLP_1 and P_1 with a block of strong absorber ($\Sigma_t = 20\text{cm}^{-1}$, $c = 0$) in the center. The source and medium (purely scattering) are the same with those for Figure 5.23. By comparison Figure 5.23 and 5.24 we can see that a block of strong absorber in Figure 5.24 makes a strong shadow in the block material region. We also can see that NLP_1 displays very similar propagation behavior with that for P_1 in the purely scattering medium.

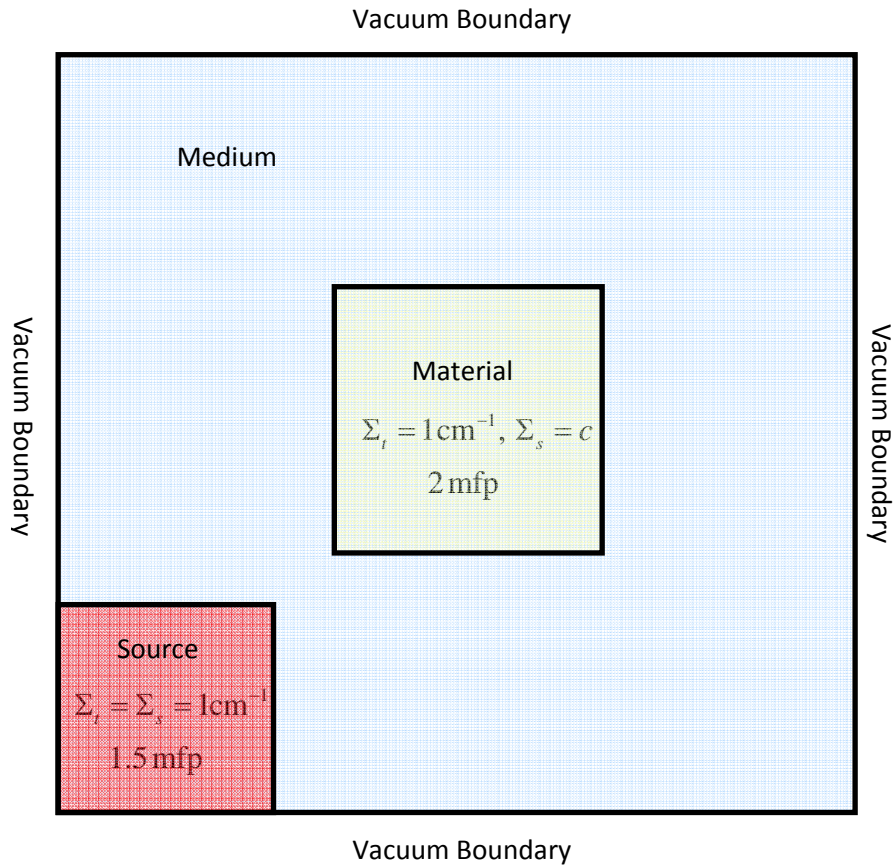


Figure 5.18 Problem of radiation propagation across a medium

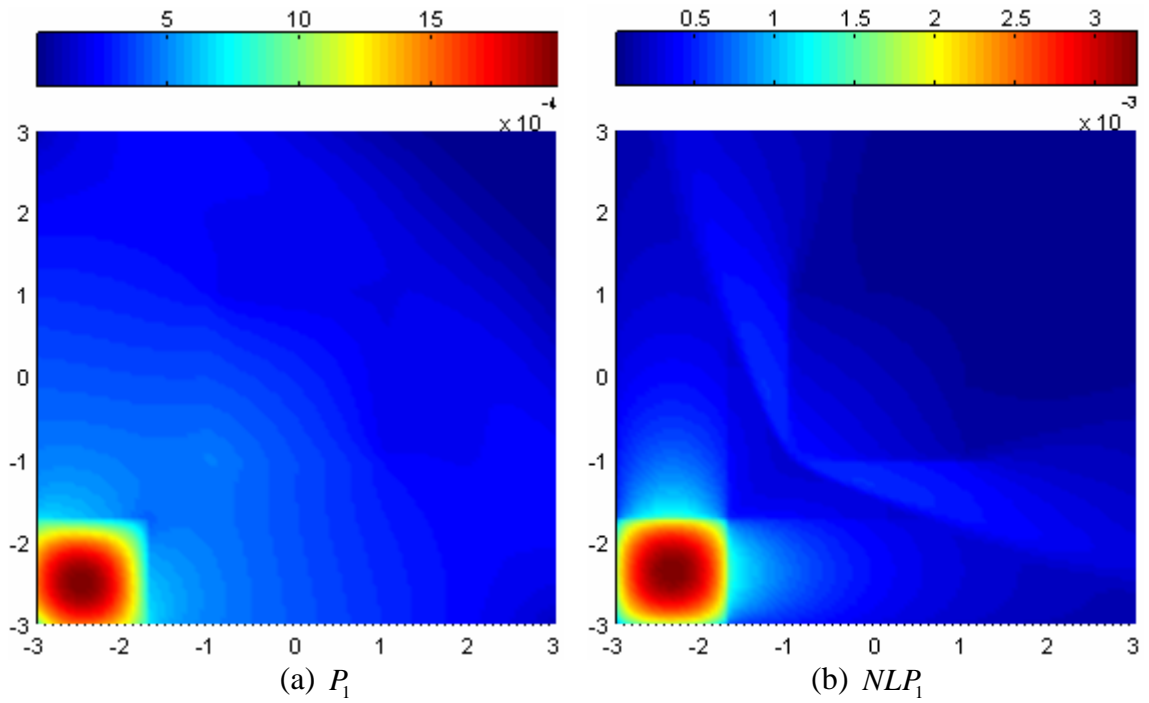


Figure 5.19 Radiation propagation for NLP_1 and P_1 across vacuum

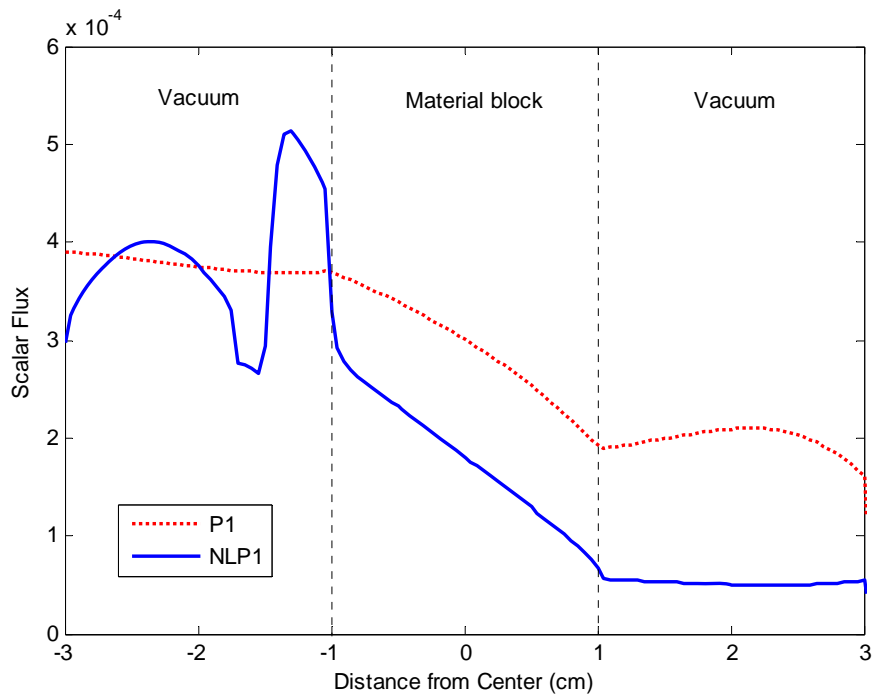


Figure 5.20 Line-out ($z = 0$) for NLP_1 and P_1 across vacuum

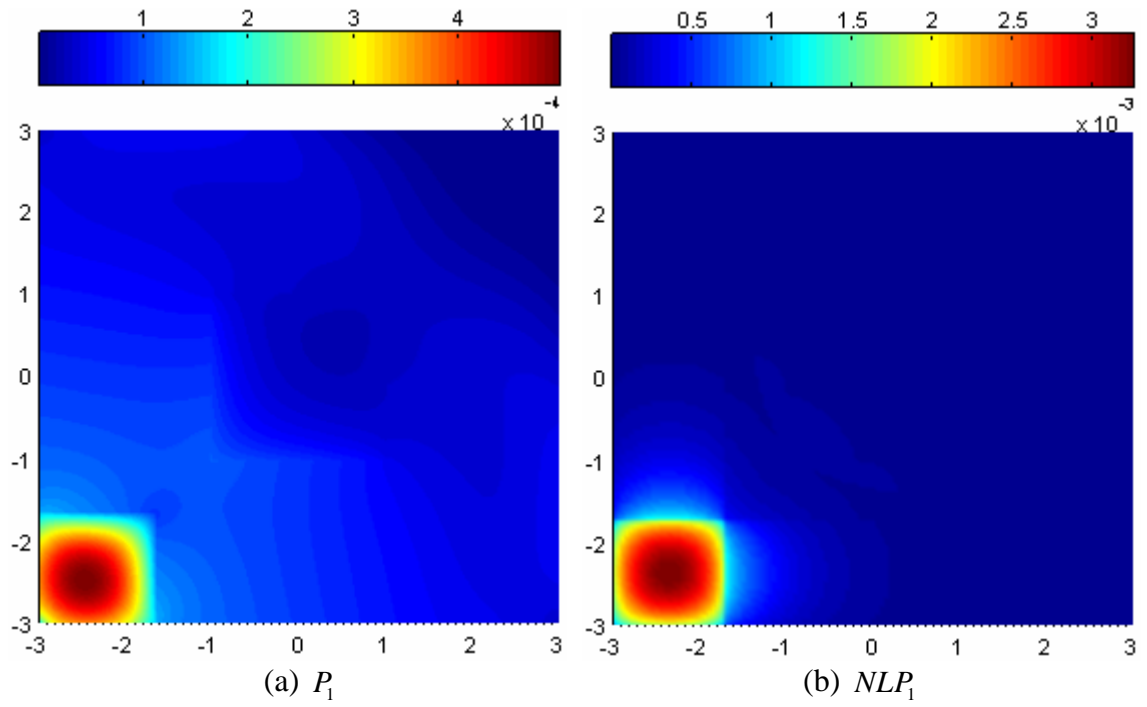


Figure 5.21 Radiation propagation for NLP_1 and P_1 across pure absorber ($c = 0$)

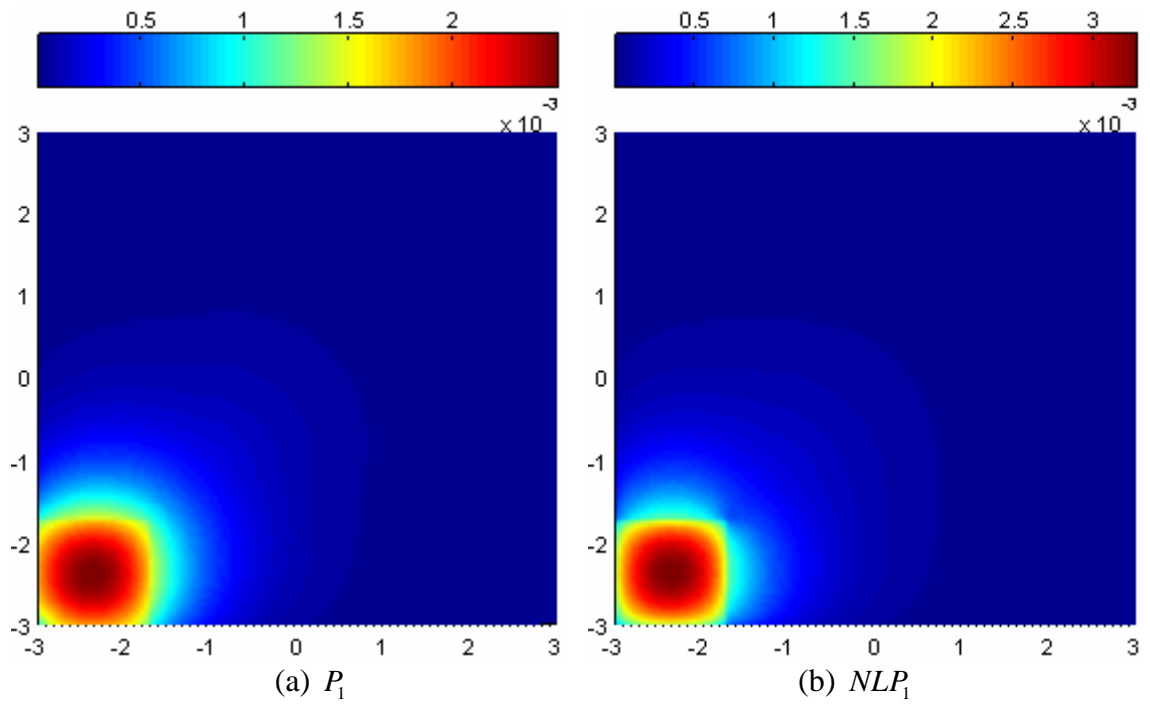


Figure 5.22 Radiation propagation for NLP_1 and P_1 across scatterer ($c = 0.5$)

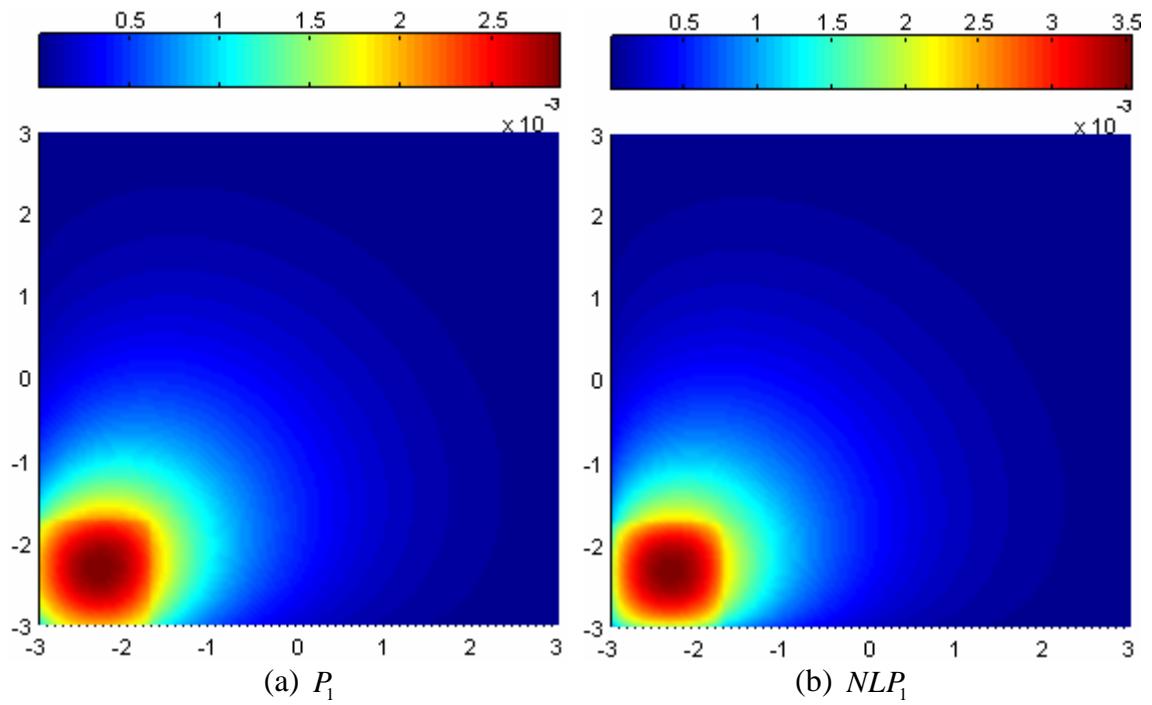


Figure 5.23 Radiation propagation for NLP_1 and P_1 across pure scatterer ($c = 1$)

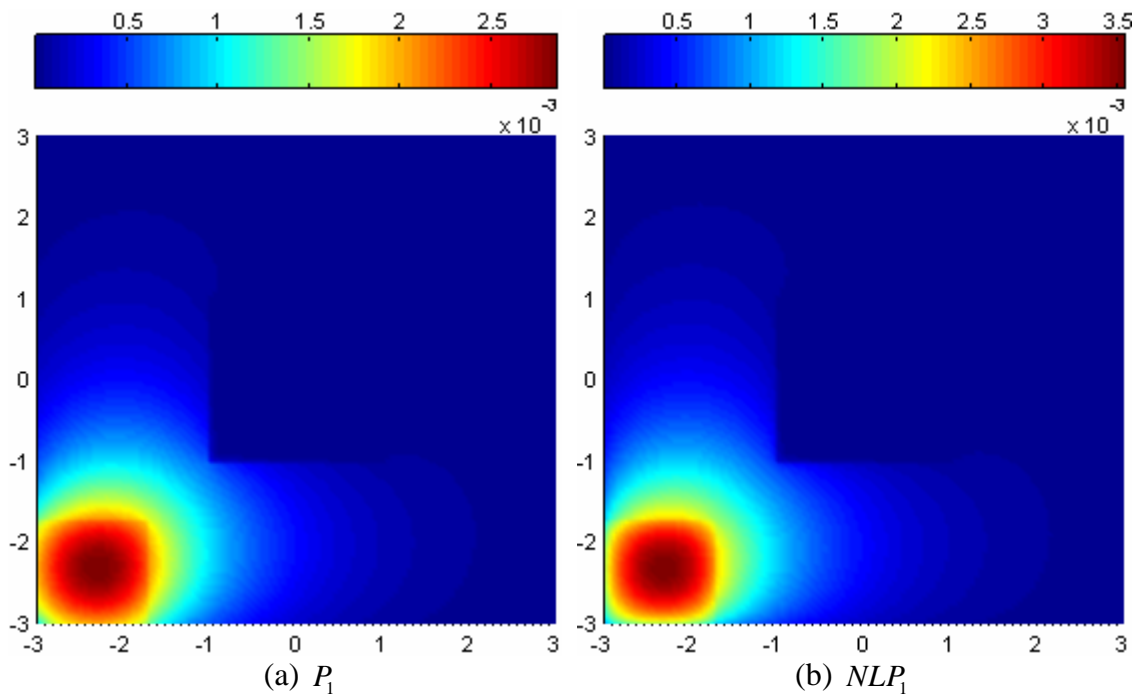


Figure 5.24 Radiation propagation for NLP_1 and P_1 with strong absorber ($\Sigma_t = 20\text{cm}^{-1}$, $c = 0$) in the center

5.9 References

- [1] D. Mihalas and M. B. Weibel, *Foundations of Radiation Hydrodynamics*, Oxford University Press, New York, 1984.
- [2] C. D. Levermore, *Bounds of Eddington Factors*, Lawrence Livermore National Laboratory, Livermore, CA, 1982.
- [3] G. N. Minerbo, "Maximum Entropy Eddington Factors" *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol.20, pp. 541, 1978.
- [4] T. A. Brunner, *Riemann solvers for time-dependent transport based on the maximum entropy and spherical harmonics closures*, University of Michigan, Ph.D. thesis, pp. 30, 2000.
- [5] D. S. Kershaw, "Flux limiting nature's own way," *Lawrence Livermore National Laboratory*, UCRL-78378, 1976.
- [6] C. D. Levermore, *Journal of Quantitative Spectroscopy & Radiative Transfer*, vol. 31, pp. 149, 1984.
- [7] B. D. Ganapol, "Solution of the one-group time-dependent neutron transport equation in and infinite medium by polynomial reconstruction," *Nuclear Science and Engineering*, vol. 92, pp. 272-279, 1986.
- [8] B. D. Ganapol, "The Generation of Time-Dependent Neutron Transport Solution in Infinite Media," *Nuclear Science and Engineering*, vol. 64, pp. 317-331, 1977.

CHAPTER 6

Conclusions

6.1 Our Goal

The ultimate goal of this study was to obtain a positive scalar flux in short time transients for time-dependent transport based on the spherical harmonics method in 2D. In this dissertation, we have studied two major approaches to achieve this goal. One is P_3 with a quasi-static closure (P_3QS), which amounts to relaxing the hyperbolicity of the spherical harmonics equation by using a quasi-static approximation. The other is a Nonlinear P_1 closure (NLP_1), which gives up the linearity of spherical harmonics equation by using a new nonlinear Eddington factor. This Nonlinear P_1 closure does yield a positive scalar flux for rapid transients in 2D.

6.2 Summary

In the P_3QS approach, we have used a quasi-static approximation to a low order spherical harmonic closure, specifically the P_3 equations. The heart of this approach is to treat the evolution of the highest order moments in a quasi-static way by eliminating their time derivatives from the high order moment equations. This is analogous to the derivation of time-dependent diffusion theory.

From the numerical results we have found that P_3QS is somewhat superior to other P_N closures because it has fewer unknowns for a given angular order, and produces a less negative scalar flux at the early times. Somewhat surprisingly, P_3QS is usually more accurate than P_3 and more accurate than even higher order P_N methods for the pulsed

source problem. However, it does still produce negative scalar fluxes during a rapid transient (less than about one collision time). Further, like diffusion theory, P_3QS is not useful in a vacuum because the equations become singular as Σ_t goes to 0.

In the second approach, we have proposed the Nonlinear P_1 closure in which the property of linearity of regular P_1 has been given up by using a new nonlinear Eddington factor. We have developed a derivation of the general form of the Eddington tensor which consists of two terms, an isotropic term and a beam-like term. We have constructed the NLP_1 Eddington factor as the simplest closure to satisfy the eigenstructure of both isotropic flow and also beam-like flow. We have found the NLP_1 Eddington factor is very similar to Maximum Entropy with respect to flux discontinuity, relaxation length and eigenstructure. But it is a simple explicit polynomial of current-to-flux ratio, while Maximum Entropy is defined only implicitly. This has enabled us to explore 2D problem numerically.

From the numerical results we have found that the NLP_1 closure produces results very similar to those of the ME closure. It is also more accurate than the Kershaw closure for the pulsed source problem in 1D. Best of all, we have found that the NLP_1 closure can produce a positive scalar flux and remains positive for rapid transients in 2D. We have found the Kershaw also remains positive for rapid transients in 2D; ours appears to be the first 2D solutions for the Kershaw closure.

6.3 Future work

1. Development of higher order nonlinear P_N closures such as NLP_3

We have found our NLP_1 can retain positive scalar flux for rapid and long time transients. Generally higher order P_N solutions are more accurate than low order solutions. So it will be interesting to see how much the higher order nonlinear P_N

closure such as NLP_3 will be better than NLP_1 and if it can still retain positivity. (Note diffusion theory namely, P_1QS retains positivity but P_3QS does not.)

2. Implementation in different geometry

Throughout this study, the derivative operators in various closures have been analyzed only in Cartesian coordinate. Implementing our methods in a different geometry like cylindrical will give us additional insight into their properties.

APPENDICES

APPENDIX A

CFL Condition

For the numerical calculation, finding a stability condition for a given system is important because it gives the useful information between the time step size and the spatial mesh spacing.

Consider an advection equation

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = 0 \quad (\text{A.1})$$

where c is the wave speed. The explicit forward Euler time discretization and the first order upwind special discretization for Eq. (A.1) is

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + c \frac{u_{i+1}^n - u_i^n}{\Delta x} = 0 \quad (\text{A.2})$$

where u_i^n is the cell averaged value in the i^{th} cell at the n^{th} time step. When the von Neumann stability analysis is applied to Eq. (A.2) we can find its stability condition.

Eq. (A.2) can be written as

$$u_i^{n+1} = u_i^n - c \frac{\Delta t}{\Delta x} (u_{i+1}^n - u_i^n) \quad (\text{A.3})$$

Define

$$\varepsilon_i^n = u_{i \text{ exact}}^n - u_i^n \quad (\text{A.4})$$

Substitute Eq. (A.4) into Eq. (A.3) to obtain

$$\varepsilon_i^{n+1} = \varepsilon_i^n - c \frac{\Delta t}{\Delta x} (\varepsilon_{i+1}^n - \varepsilon_i^n) \quad (\text{A.5})$$

Write the error as a wave form and then the error at node $n + 1, i + 1$ is written as

$$\varepsilon_i^{n+1} = \varepsilon^{n+1} e^{ikx_i} \quad (\text{A.6})$$

$$\varepsilon_{i+1}^n = \varepsilon^n e^{ik(x_i + \Delta x)} \quad (\text{A.7})$$

where k is constant.

Using Eq. (A.6) and (A.7), Eq. (A.5) can be rewritten as

$$\frac{\varepsilon_i^{n+1}}{\varepsilon_i^n} = 1 - c \frac{\Delta t}{\Delta x} e^{ik\Delta x} + c \frac{\Delta t}{\Delta x} = 1 + c \frac{\Delta t}{\Delta x} - c \frac{\Delta t}{\Delta x} \cos k\Delta x - ic \frac{\Delta t}{\Delta x} \sin k\Delta x \quad (\text{A.8})$$

where $\varepsilon_i^{n+1} / \varepsilon_i^n$ is called amplification factor. Stability requires

$$\left| \frac{\varepsilon_i^{n+1}}{\varepsilon_i^n} \right| \leq 1 \quad (\text{A.9})$$

so that the error decays.

After some algebraic manipulation, Eq. (A.8) becomes

$$\left| \frac{\varepsilon_i^{n+1}}{\varepsilon_i^n} \right|^2 = 1 + 2c \frac{\Delta t}{\Delta x} \left(1 + c \frac{\Delta t}{\Delta x} \right) (1 - \cos k\Delta x) \quad (\text{A.10})$$

Since $1 - \cos k\Delta x \geq 0$ for all $-\pi < k\Delta x < \pi$, $\left| \frac{\varepsilon_i^{n+1}}{\varepsilon_i^n} \right| \leq 1$ if and only if

$$2c \frac{\Delta t}{\Delta x} \left(1 + c \frac{\Delta t}{\Delta x} \right) \leq 0 \quad \text{or equivalently} \quad -1 \leq c \frac{\Delta t}{\Delta x} \leq 0 \quad \text{where } c > 0 \quad (\text{A.11})$$

If we assume the wave speed c is negative, Eq. (A.8) becomes

$$-2c \frac{\Delta t}{\Delta x} \left(1 - c \frac{\Delta t}{\Delta x} \right) \leq 0 \quad \text{or equivalently} \quad 0 \leq c \frac{\Delta t}{\Delta x} \leq 1 \quad (\text{A.12})$$

From Eq. (A.11) and (A.12) we can find the stability condition for the explicit forward Euler differencing is

$$\left| c \frac{\Delta t}{\Delta x} \right| \leq 1 \quad (\text{A.13})$$

As we can see Eq. (A.13), a conditional stability requirement must be placed on the time step size and the spatial mesh spacing. This is called the Courant-Friedrichs-Lewy (CFL) condition [10] and this should be kept for the stability of solution.

¹⁰R. Courant, K. Friedrichs and H. Lewy, Über die partiellen Differenzgleichungen der mathematischen Physik, *Mathematische Annalen*, vol. 100, no. 1, pages 32–74, 1928.

APPENDIX B

Roe matrix for a two moment Eddington type closure

A two moment Eddington type closure can have the following general form of Roe matrix

$$R(u_l, u_r) = \begin{bmatrix} 0 & 1 \\ \frac{\chi^l \eta^r - \chi^r \eta^l}{\eta^r - \eta^l} & \frac{\chi^r - \chi^l}{\eta^r - \eta^l} \end{bmatrix} \quad (\text{B.1})$$

where $\chi^{r,l} = \chi(\eta^{r,l})$ and $\eta^{r,l} = u_1^{r,l} / u_0^{r,l}$ respectively for “ r ” and “ l ”. Proving that this matrix satisfies the Roe property **R3** in Section 3.4 is a matter of algebra

$$\begin{aligned} R(u_l, u_r)(u_l - u_r) &= \begin{bmatrix} 0 & 1 \\ \frac{\chi_l \eta_r - \chi_r \eta_l}{\eta_r - \eta_l} & \frac{\chi_r - \chi_l}{\eta_r - \eta_l} \end{bmatrix} \begin{bmatrix} \phi_r - \phi_l \\ J_r - J_l \end{bmatrix} \\ &= \begin{bmatrix} J_r - J_l \\ \left(\frac{\chi_l \eta_r - \chi_r \eta_l}{\eta_r - \eta_l} \right) (\phi_r - \phi_l) + \frac{\chi_r - \chi_l}{\eta_r - \eta_l} (J_r - J_l) \end{bmatrix} \\ &= \begin{bmatrix} J_r - J_l \\ \frac{\chi_l \eta_r \phi_r - \chi_r \eta_l \phi_r - \chi_l \eta_r \phi_l + \chi_r \eta_l \phi_l + \chi_r J_r - \chi_l J_r - \chi_r J_l + \chi_l J_l}{\eta_r - \eta_l} + (\phi_r - \phi_l) \end{bmatrix} \\ &= \begin{bmatrix} J_r - J_l \\ \frac{\chi_r \phi_r (\eta_r - \eta_l) - \chi_l \phi_l (\eta_r - \eta_l)}{\eta_r - \eta_l} + (\phi_r - \phi_l) \end{bmatrix} \\ &= \begin{bmatrix} J_r - J_l \\ \chi_r \phi_r - \chi_l \phi_l \end{bmatrix} \\ &= F(u_l) - F(u_r) \end{aligned} \quad (\text{B.2})$$

Also by letting both η_r and η_l go to η in Eq. (B.1), we can show this matrix satisfied the Roe property **R2** in Section 3.4 as

$$\lim_{\eta_r, \eta_l \rightarrow \eta} R(\eta_l, \eta_r) = \begin{bmatrix} 0 & 1 \\ \chi - \eta \frac{d\chi}{d\eta} & \frac{d\chi}{d\eta} \end{bmatrix} = A \quad (\text{B.3})$$

APPENDIX C

2D P_3QS Riemann Solver (coded by Kyeong)

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>

double Hfunc(double h, double u1_im, double u1_i, double u1_ip, double u2_im, double u2_i,
double u2_ip, double u3_im, double u3_i, double u3_ip);
double Rfunc(double h, double u1_im, double u1_i, double u1_ip, double u3_im, double u3_i,
double u3_ip);
double mfunc(double u_im, double u_i, double u_ip, double h);

main()
{
double tx = 1;
double c = 1;

int ncells_x = 801;           int ncells_z = 801;
int nx = ncells_x + 2;       int nz = ncells_z + 2;
double length_x = 2;         double length_z = 2;
double hx = length_x/(ncells_x-1); double hz = length_z / (ncells_z-1);
int nstep = 100001;
double dt = 0.000005;
double time = 0.0;

double l = 1 - dt*tx*(1-c);
double r = 1 - dt*tx;
double p = dt/(2*hx);
double q = dt/(2*hz);
double s = dt/(hx*hx);
double w = dt/(hz*hz);
double v = dt/(4*hx*hz);

int a, b, d, e, f, z; //double
u0[nx][nz], u1[nx][nz], u2[nx][nz], u3[nx][nz], u4[nx][nz], u5[nx][nz];

int index_mem; // for mem alloc
double **u0 = (double **)malloc(nx*sizeof(double *));
for(index_mem=0; index_mem<nx; index_mem++)
    u0[index_mem] = (double *)malloc(nz*sizeof(double));
double **u1 = (double **)malloc(nx*sizeof(double *));
for(index_mem=0; index_mem<nx; index_mem++)
    u1[index_mem] = (double *)malloc(nz*sizeof(double));
double **u2 = (double **)malloc(nx*sizeof(double *));
for(index_mem=0; index_mem<nx; index_mem++)
    u2[index_mem] = (double *)malloc(nz*sizeof(double));
double **u3 = (double **)malloc(nx*sizeof(double *));
for(index_mem=0; index_mem<nx; index_mem++)
    u3[index_mem] = (double *)malloc(nz*sizeof(double));
double **u4 = (double **)malloc(nx*sizeof(double *));
for(index_mem=0; index_mem<nx; index_mem++)
    u4[index_mem] = (double *)malloc(nz*sizeof(double));
double **u5 = (double **)malloc(nx*sizeof(double *));
for(index_mem=0; index_mem<nx; index_mem++)
    u5[index_mem] = (double *)malloc(nz*sizeof(double));

/*initial conditions begin */

for (b = 0; b <= nz-1; b++)
{
for (a = 0; a <= nx-1; a++)
{
u0[a][b] = 0; u1[a][b] = 0; u2[a][b] = 0;
u3[a][b] = 0; u4[a][b] = 0; u5[a][b] = 0;

```

```

    }
}

u0[(nx-1)/2][(nz-1)/2] = 1/(hx*hz);

/*initial conditions end */

/* boundry Condition (Vaccum) begin */

for (d = 0; d <= nx-1; d++)
{
    u0[d][0] = 0;    u1[d][0] = 0;    u2[d][0] = 0;
    u3[d][0] = 0;    u4[d][0] = 0;    u5[d][0] = 0;
    u0[d][nz-1] = 0; u1[d][nz-1] = 0; u2[d][nz-1] = 0;
    u3[d][nz-1] = 0; u4[d][nz-1] = 0; u5[d][nz-1] = 0;
}

for (e = 1; e <= nz-2; e++)
{
    u0[0][e] = 0; u1[0][e] = 0; u2[0][e] = 0;
    u3[0][e] = 0; u4[0][e] = 0; u5[0][e] = 0;
    u0[nx-1][e] = 0; u1[nx-1][e] = 0; u2[nx-1][e] = 0;
    u3[nx-1][e] = 0; u4[nx-1][e] = 0; u5[nx-1][e] = 0;
}

for (f = 1; f <= nx-2; f++)
{
    u0[f][1] = 0; u1[f][1] = 0; u2[f][1] = 0;
    u3[f][1] = 0; u4[f][1] = 0; u5[f][1] = 0;

    u0[f][nz-2] = 0; u1[f][nz-2] = 0; u2[f][nz-2] = 0;
    u3[f][nz-2] = 0; u4[f][nz-2] = 0; u5[f][nz-2] = 0;
}

for (z = 2; z <= nz-3; z++)
{
    u0[1][z] = 0; u1[1][z] = 0; u2[1][z] = 0;
    u3[1][z] = 0; u4[1][z] = 0; u5[1][z] = 0;
    u0[nx-2][z] = 0; u1[nx-2][z] = 0; u2[nx-2][z] = 0;
    u3[nx-2][z] = 0; u4[nx-2][z] = 0; u5[nx-2][z] = 0;
}

/* boundry Condition (Vaccum) end */

int g,k,m,i,j ; int ii,jj;

double a0,a1,a2,a3,a4,a5,    b0,b1,b2,b3,b4,b5,    c0,c1,c2,c3,c4,c5,
        d0,d1,d2,d3,d4,d5,    e0,e1,e2,e3,e4,e5,    g0,g1,g2,g3,g4,g5,
        j0,j1,j2,j3,j4,j5,    k0,k1,k2,k3,k4,k5;

double H0x, H1x, H2x, H3x, H4x, H5x, H0z, H1z, H2z, H3z, H4z, H5z,
        R0x, R1x, R2x, R3x, R4x, R5x, R0z, R1z, R2z, R3z, R4z, R5z;

double **y0 = (double **)malloc(nx*sizeof(double *));

for(yindex_mem=0; yindex_mem<nx; yindex_mem++)
    y0[yindex_mem] = (double *)malloc(nz*sizeof(double));

double **y1 = (double **)malloc(nx*sizeof(double *));
for(yindex_mem=0; yindex_mem<nx; yindex_mem++)
    y1[yindex_mem] = (double *)malloc(nz*sizeof(double));
double **y2 = (double **)malloc(nx*sizeof(double *));
for(yindex_mem=0; yindex_mem<nx; yindex_mem++)
    y2[yindex_mem] = (double *)malloc(nz*sizeof(double));
double **y3 = (double **)malloc(nx*sizeof(double *));
for(yindex_mem=0; yindex_mem<nx; yindex_mem++)
    y3[yindex_mem] = (double *)malloc(nz*sizeof(double));
double **y4 = (double **)malloc(nx*sizeof(double *));
for(yindex_mem=0; yindex_mem<nx; yindex_mem++)
    y4[yindex_mem] = (double *)malloc(nz*sizeof(double));
double **y5 = (double **)malloc(nx*sizeof(double *));

```

```

for(yindex_mem=0; yindex_mem<nx; yindex_mem++)
  y5[yindex_mem] = (double *)malloc(nz*sizeof(double));

FILE *file;

file = fopen("HR400t05", "w");

for (m = 1; m <= nstep; m++)
  {
    printf("%f\n", time);

    for (g = 0; g <= nz-1; g++)
      {
        for (k = 0; k <= nx-1; k++)
          {
            y0[k][g] = u0[k][g]; y1[k][g] = u1[k][g]; y2[k][g] = u2[k][g];
            y3[k][g] = u3[k][g]; y4[k][g] = u4[k][g]; y5[k][g] = u5[k][g];
          }
        }

    for (j = 2; j <= nz-3; j++)
      {
        for (i = 2; i <= nx-3; i++)
          {
H0x = Hfunc(hx,y0[i-2][j],y0[i-1][j],y0[i][j],y0[i-1][j],y0[i+1][j],y0[i+2][j]);
H1x = Hfunc(hx,y1[i-2][j],y1[i-1][j],y1[i][j],y1[i-1][j],y1[i+1][j],y1[i+2][j]);
H2x = Hfunc(hx,y2[i-2][j],y2[i-1][j],y2[i][j],y2[i-1][j],y2[i+1][j],y2[i+2][j]);
H3x = Hfunc(hx,y3[i-2][j],y3[i-1][j],y3[i][j],y3[i-1][j],y3[i+1][j],y3[i+2][j]);
H4x = Hfunc(hx,y4[i-2][j],y4[i-1][j],y4[i][j],y4[i-1][j],y4[i+1][j],y4[i+2][j]);
H5x = Hfunc(hx,y5[i-2][j],y5[i-1][j],y5[i][j],y5[i-1][j],y5[i+1][j],y5[i+2][j]);

R0x = Rfunc(hx,y0[i-2][j],y0[i-1][j],y0[i][j],y0[i][j],y0[i+1][j],y0[i+2][j]);
R1x = Rfunc(hx,y1[i-2][j],y1[i-1][j],y1[i][j],y1[i][j],y1[i+1][j],y1[i+2][j]);
R2x = Rfunc(hx,y2[i-2][j],y2[i-1][j],y2[i][j],y2[i][j],y2[i+1][j],y2[i+2][j]);
R3x = Rfunc(hx,y3[i-2][j],y3[i-1][j],y3[i][j],y3[i][j],y3[i+1][j],y3[i+2][j]);
R4x = Rfunc(hx,y4[i-2][j],y4[i-1][j],y4[i][j],y4[i][j],y4[i+1][j],y4[i+2][j]);
R5x = Rfunc(hx,y5[i-2][j],y5[i-1][j],y5[i][j],y5[i][j],y5[i+1][j],y5[i+2][j]);

H0z = Hfunc(hz,y0[i][j-2],y0[i][j-1],y0[i][j],y0[i][j-1],y0[i][j+1],y0[i][j+2]);
H1z = Hfunc(hz,y1[i][j-2],y1[i][j-1],y1[i][j],y1[i][j-1],y1[i][j+1],y1[i][j+2]);
H2z = Hfunc(hz,y2[i][j-2],y2[i][j-1],y2[i][j],y2[i][j-1],y2[i][j+1],y2[i][j+2]);
H3z = Hfunc(hz,y3[i][j-2],y3[i][j-1],y3[i][j],y3[i][j-1],y3[i][j+1],y3[i][j+2]);
H4z = Hfunc(hz,y4[i][j-2],y4[i][j-1],y4[i][j],y4[i][j-1],y4[i][j+1],y4[i][j+2]);
H5z = Hfunc(hz,y5[i][j-2],y5[i][j-1],y5[i][j],y5[i][j-1],y5[i][j+1],y5[i][j+2]);

R0z = Rfunc(hz,y0[i][j-2],y0[i][j-1],y0[i][j],y0[i][j],y0[i][j+1],y0[i][j+2]);
R1z = Rfunc(hz,y1[i][j-2],y1[i][j-1],y1[i][j],y1[i][j],y1[i][j+1],y1[i][j+2]);
R2z = Rfunc(hz,y2[i][j-2],y2[i][j-1],y2[i][j],y2[i][j],y2[i][j+1],y2[i][j+2]);
R3z = Rfunc(hz,y3[i][j-2],y3[i][j-1],y3[i][j],y3[i][j],y3[i][j+1],y3[i][j+2]);
R4z = Rfunc(hz,y4[i][j-2],y4[i][j-1],y4[i][j],y4[i][j],y4[i][j+1],y4[i][j+2]);
R5z = Rfunc(hz,y5[i][j-2],y5[i][j-1],y5[i][j],y5[i][j],y5[i][j+1],y5[i][j+2]);

a0 = y0[i+1][j] - y0[i-1][j] + H0x ; b0 = y0[i-1][j] - 2.0*y0[i][j] + y0[i+1][j] + R0x ;
a1 = y1[i+1][j] - y1[i-1][j] + H1x ; b1 = y1[i-1][j] - 2.0*y1[i][j] + y1[i+1][j] + R1x ;
a2 = y2[i+1][j] - y2[i-1][j] + H2x ; b2 = y2[i-1][j] - 2.0*y2[i][j] + y2[i+1][j] + R2x ;
a3 = y3[i+1][j] - y3[i-1][j] + H3x ; b3 = y3[i-1][j] - 2.0*y3[i][j] + y3[i+1][j] + R3x ;
a4 = y4[i+1][j] - y4[i-1][j] + H4x ; b4 = y4[i-1][j] - 2.0*y4[i][j] + y4[i+1][j] + R4x ;
a5 = y5[i+1][j] - y5[i-1][j] + H5x ; b5 = y5[i-1][j] - 2.0*y5[i][j] + y5[i+1][j] + R5x ;

```

```

c0 = y0[i][j+1] - y0[i][j-1] + H0z ; d0 = y0[i][j-1] - 2.0*y0[i][j] + y0[i][j+1] + R0z ;
c1 = y1[i][j+1] - y1[i][j-1] + H1z ; d1 = y1[i][j-1] - 2.0*y1[i][j] + y1[i][j+1] + R1z ;
c2 = y2[i][j+1] - y2[i][j-1] + H2z ; d2 = y2[i][j-1] - 2.0*y2[i][j] + y2[i][j+1] + R2z ;
c3 = y3[i][j+1] - y3[i][j-1] + H3z ; d3 = y3[i][j-1] - 2.0*y3[i][j] + y3[i][j+1] + R3z ;
c4 = y4[i][j+1] - y4[i][j-1] + H4z ; d4 = y4[i][j-1] - 2.0*y4[i][j] + y4[i][j+1] + R4z ;
c5 = y5[i][j+1] - y5[i][j-1] + H5z ; d5 = y5[i][j-1] - 2.0*y5[i][j] + y5[i][j+1] + R5z ;

e0 = y0[i-1][j] - 2.0*y0[i][j] + y0[i+1][j]; g0 = y0[i][j-1] - 2.0*y0[i][j] + y0[i][j+1];
e1 = y1[i-1][j] - 2.0*y1[i][j] + y1[i+1][j]; g1 = y1[i][j-1] - 2.0*y1[i][j] + y1[i][j+1];
e2 = y2[i-1][j] - 2.0*y2[i][j] + y2[i+1][j]; g2 = y2[i][j-1] - 2.0*y2[i][j] + y2[i][j+1];
e3 = y3[i-1][j] - 2.0*y3[i][j] + y3[i+1][j]; g3 = y3[i][j-1] - 2.0*y3[i][j] + y3[i][j+1];
e4 = y4[i-1][j] - 2.0*y4[i][j] + y4[i+1][j]; g4 = y4[i][j-1] - 2.0*y4[i][j] + y4[i][j+1];
e5 = y5[i-1][j] - 2.0*y5[i][j] + y5[i+1][j]; g5 = y5[i][j-1] - 2.0*y5[i][j] + y5[i][j+1];

j0 = y0[i+1][j+1] - y0[i+1][j-1]; k0 = y0[i-1][j+1] - y0[i-1][j-1];
j1 = y1[i+1][j+1] - y1[i+1][j-1]; k1 = y1[i-1][j+1] - y1[i-1][j-1];
j2 = y2[i+1][j+1] - y2[i+1][j-1]; k2 = y2[i-1][j+1] - y2[i-1][j-1];
j3 = y3[i+1][j+1] - y3[i+1][j-1]; k3 = y3[i-1][j+1] - y3[i-1][j-1];
j4 = y4[i+1][j+1] - y4[i+1][j-1]; k4 = y4[i-1][j+1] - y4[i-1][j-1];
j5 = y5[i+1][j+1] - y5[i+1][j-1]; k5 = y5[i-1][j+1] - y5[i-1][j-1];

/* Governing equation begin */

u0[i][j] = 1*y0[i][j] - p*(- sqrt(2.0/3.0)*a3 - 1.0/3.0*sqrt(5.0/3.0)*b0 +
1.0/3.0*sqrt(1.0/3.0)*b2 - 1.0/3.0*sqrt(2.0)*b5) - q*( sqrt(1.0/3.0) *c1 -
1.0/3.0*sqrt(5.0/3.0)*d0 - 2.0/3.0*sqrt(1.0/3.0)*d2 );

u1[i][j] = r*y1[i][j] - p*(- sqrt(2.0/5.0)*a4 - 1.0/sqrt(5.0)*b1)
- q*( sqrt(1.0/3.0)*c0 + sqrt(4.0/15.0)*c2 - sqrt(3.0/5.0)*d1);

u2[i][j] = r*y2[i][j] - p*( sqrt(2.0/15.0)*a3 + 1.0/3.0*sqrt(1.0/3.0)*b0-
1.0/3.0*sqrt(1.0/15.0)*b2+1.0/3.0*sqrt(2.0/5.0)*b5) - q*( sqrt(4.0/15.0)*c1 -
2.0/3.0*sqrt(1.0/3.0)*d0 - 4.0/3.0*sqrt(1.0/15.0)*d2 ) - (-6.0/35.0 *s*e2 +
sqrt(6.0)/35.0 *s*e5 - 9.0/35.0 *w*g2 + sqrt(6.0)/35.0 * v*(j4 - k4));

u3[i][j] = r*y3[i][j] - p*(- sqrt(1.0/6.0)*a0 + sqrt(1.0/30.0)*a2 - sqrt(1.0/5.0)*a5
- sqrt(3.0/5.0)*b3 ) - q*( sqrt(1.0/5.0)*c4 - sqrt(1.0/5.0)*d3 );

u4[i][j] = r*y4[i][j] - p*(- sqrt(1.0/10.0)*a1 - sqrt(1.0/5.0)*b4)- q*( sqrt(1.0/5.0)*c3
- sqrt(1.0/5.0)*d4 ) - (-8.0/35.0 *s*e4 - 8.0/35.0 *w*g4 + sqrt(6.0)/70.0 *v*(j2 - k2) +
3.0/35.0 *v*(j5 - k5));

u5[i][j] = r*y5[i][j] - p*(- sqrt(1.0/5.0)*a3-
1.0/3.0*sqrt(1.0/2.0)*b0+1.0/3.0*sqrt(1.0/10.0)*b2 - 1.0/3.0*sqrt(3.0/5.0)*b5) -
q*( 0.0 ) - (sqrt(6.0)/70.0 *s*e2 - 8.0/35.0 *s*e5 - 1.0/7.0*w*g5 + 3.0/35.0 *v*(j4 -
k4));

/* Governing equation end */

    }
}

time = time + dt;
}

printf("\n");

for (jj = 0; jj <= nz-1; jj++)
{
for (ii = 0; ii <= nx-1; ii++)
{
fprintf(file, "%f\n", u0[ii][jj]);
}
}

printf("\n");

for(index_mem=0;index_mem<nx;index_mem++) free(u0[index_mem]); free(u0);
for(index_mem=0;index_mem<nx;index_mem++) free(u1[index_mem]); free(u1);

```

```

for(index_mem=0;index_mem<nx;index_mem++) free(u2[index_mem]); free(u2);
for(index_mem=0;index_mem<nx;index_mem++) free(u3[index_mem]); free(u3);
for(index_mem=0;index_mem<nx;index_mem++) free(u4[index_mem]); free(u4);
for(index_mem=0;index_mem<nx;index_mem++) free(u5[index_mem]); free(u5);

}

double Hfunc(double h, double u1_im, double u1_i, double u1_ip, double u2_im, double u2_i,
double u2_ip, double u3_im, double u3_i, double u3_ip)
{
    double m_im, m_i, m_ip;
        m_im = mfunc(u1_im, u1_i, u1_ip, h);
        m_i  = mfunc(u2_im, u2_i, u2_ip, h);
        m_ip = mfunc(u3_im, u3_i, u3_ip, h);
    double Hm;
        Hm = (2.0*m_i - m_im - m_ip) * h/2.0;
    return (Hm);
}

double Rfunc(double h, double u1_im, double u1_i, double u1_ip, double u3_im, double u3_i,
double u3_ip)
{
    double m_im, m_i, m_ip;
        m_im = mfunc(u1_im, u1_i, u1_ip, h);
        m_ip = mfunc(u3_im, u3_i, u3_ip, h);
    double Rm;
        Rm = (m_im - m_ip) * h/2.0;
    return (Rm);
}

double mfunc(double u_im,double u_i,double u_ip,double h)
{
    double mp, mm, test_sign, m ;
        mp = (u_i-u_im)/h;
        mm = (u_ip-u_i)/h;
        test_sign = mp*mm;
        if (test_sign > 0)
            m = 2.0*mp*mm/(mp+mm);
        else
            m = 0;
    return (m);
}

```

APPENDIX D

2D Nonlinear P_1 Riemann Solver (coded by Kyeong)

```

function Nonlinear_P1_2D

tx = 1;           % total cross section
c = 1;           % scattering ratio

MX = 101;        % X-dir. number of real cells
MZ = 101;        % Y-dir. number of real cells

Xlength = 4;     % length of material
Zlength = 4;     % length of material

nx = MX+2;       % X-dir. number of cells ghost cells
nz = MZ+2;       % Y-dir. number of cells ghost cells

hx = Xlength/(MX-1); % cell width
hz = Zlength/(MZ-1); % cell width

x = (-hx-0.5*Xlength):hx:(0.5*Xlength+hx);
z = (-hz-0.5*Zlength):hz:(0.5*Zlength+hz);

nstep = 101;     % number of step
dt = 0.01;       % time step size

p = 1- dt*tx.*(1 - c); % precalculation
q = 1.0 - dt.*tx;     % precalculation
r = dt./(2.0.*hx);    % precalculation
time = 0.0;

%Boundary and Initial Condition begin ---
u0 = ones(nx,nz)* 1e-10; % to avoid by dividing by 0
u1 = zeros(nx,nz);
u2 = zeros(nx,nz);
u0((nx+1)./2, (nz+1)./2) = 1./(hx.*hz); % initial delta funtion at the center
%Boundary and Initial Condition end-----

% Begin time iteration
for m = 1:nstep time

    % store the solution
    for j = 1:nz
        for i = 1:nx
            y0(i,j) = u0(i,j);
            y1(i,j) = u1(i,j);
            y2(i,j) = u2(i,j);
            eta_x(i,j) = (y1(i,j)./y0(i,j) );
            eta_y(i,j) = (y2(i,j)./y0(i,j) );
            G(i,j) = 0.5*((eta_x(i,j).^2 + eta_y(i,j).^2).^2)...
            + 0.5*(eta_x(i,j).^2 + eta_y(i,j).^2);
            a(i,j) = 1/3*(1-G(i,j)).*y0(i,j);

        end
    end

% Begin cell iteration

    for j = 2:nz-1
        for i = 2:nx-1,

            %---- cc and dd-----

            eta_x_r = (y1(i+1,j)+ y1(i,j))./(y0(i+1,j)+ y0(i,j));
            eta_y_r = (y2(i+1,j)+ y2(i,j))./(y0(i+1,j)+ y0(i,j));

```

```

x_r_21 = 1/3 - (eta_x_r^4) + 1/2*(eta_y_r^4) -1/2 *(eta_x_r^2)...
          *(eta_y_r^2) -1/3*(eta_x_r^2) + 1/6*(eta_y_r^2);
x_r_22 = 4/3*(eta_x_r^3) + 1/3*(eta_x_r)*(eta_y_r^2) + 2/3 *(eta_x_r) ;
x_r_23 = 1/3*(eta_x_r^2)*(eta_y_r) - 2/3*(eta_y_r^3) -1/3*(eta_y_r);
x_r_31 = -3/2*(eta_x_r^3)*(eta_y_r) -3/2*(eta_x_r)*(eta_y_r^3)...
          - 1/2*(eta_x_r)*(eta_y_r);
x_r_32 = 3/2*(eta_x_r^2)*(eta_y_r) + 1/2*(eta_y_r^3) + 1/2*(eta_y_r);
x_r_33 = 1/2*(eta_x_r^3) + 3/2*(eta_x_r)*(eta_y_r^2) +1/2*(eta_x_r);

%-----

Rx_r = [          0,          1,          0;
         x_r_21,      x_r_22,      x_r_23;
         x_r_31,      x_r_32,      x_r_33];

cc = get_cc(Rx_r);

%-----

eta_x_l = (y1(i,j)+ y1(i-1,j))./(y0(i,j)+ y0(i-1,j));
eta_y_l = (y2(i,j)+ y2(i-1,j))./(y0(i,j)+ y0(i-1,j));

x_l_21 = 1/3 - (eta_x_l^4) + 1/2*(eta_y_l^4) -1/2 *(eta_x_l^2)...
          *(eta_y_l^2) -1/3*(eta_x_l^2) + 1/6*(eta_y_l^2);
x_l_22 = 4/3*(eta_x_l^3) + 1/3*(eta_x_l)*(eta_y_l^2) + 2/3 *(eta_x_l) ;
x_l_23 = 1/3*(eta_x_l^2)*(eta_y_l) - 2/3*(eta_y_l^3) -1/3*(eta_y_l);
x_l_31 = -3/2*(eta_x_l^3)*(eta_y_l) -3/2*(eta_x_l)*(eta_y_l^3)...
          - 1/2*(eta_x_l)*(eta_y_l);
x_l_32 = 3/2*(eta_x_l^2)*(eta_y_l) + 1/2*(eta_y_l^3) + 1/2*(eta_y_l);
x_l_33 = 1/2*(eta_x_l^3) + 3/2*(eta_x_l)*(eta_y_l^2) +1/2*(eta_x_l);

%-----

Rx_l = [          0,          1,          0;
         x_l_21,      x_l_22,      x_l_23;
         x_l_31,      x_l_32,      x_l_33];

dd = get_cc(Rx_l);

%---- ee and gg-----

eta_x_r = (y1(i,j+1)+ y1(i,j))./(y0(i,j+1)+ y0(i,j));
eta_y_r = (y2(i,j+1)+ y2(i,j))./(y0(i,j+1)+ y0(i,j));

y_r_21 = -3/2*(eta_x_r^3)*(eta_y_r) -3/2*(eta_x_r)*(eta_y_r^3)...
          - 1/2*(eta_x_r)*(eta_y_r);
y_r_22 = 3/2*(eta_x_r^2)*(eta_y_r) + 1/2*(eta_y_r^3) + 1/2*(eta_y_r);
y_r_23 = 1/2*(eta_x_r^3) + 3/2*(eta_x_r)*(eta_y_r^2) +1/2*(eta_x_r);
y_r_31 = 1/3 - (eta_y_r^4) + 1/2*(eta_x_r^4) -1/2 *(eta_x_r^2)...
          *(eta_y_r^2) -1/3*(eta_y_r^2) + 1/6*(eta_x_r^2);
y_r_32 = 1/3*(eta_x_r)*(eta_y_r^2) - 2/3*(eta_x_r^3) -1/3*(eta_x_r);
y_r_33 = 4/3*(eta_y_r^3) + 1/3*(eta_x_r^2)*(eta_y_r) + 2/3 *(eta_y_r) ;

%-----

Ry_r = [          0,          0,          1;
         y_r_21,      y_r_22,      y_r_23;
         y_r_31,      y_r_32,      y_r_33];

gg = get_cc(Ry_r);

%-----

eta_x_l = (y1(i,j)+ y1(i,j-1))./(y0(i,j)+ y0(i,j-1));
eta_y_l = (y2(i,j)+ y2(i,j-1))./(y0(i,j)+ y0(i,j-1));

y_l_21 = -3/2*(eta_x_l^3)*(eta_y_l) -3/2*(eta_x_l)*(eta_y_l^3)...
          - 1/2*(eta_x_l)*(eta_y_l);
y_l_22 = 3/2*(eta_x_l^2)*(eta_y_l) + 1/2*(eta_y_l^3) + 1/2*(eta_y_l);
y_l_23 = 1/2*(eta_x_l^3) + 3/2*(eta_x_l)*(eta_y_l^2) +1/2*(eta_x_l);
y_l_31 = 1/3 - (eta_y_l^4) + 1/2*(eta_x_l^4) -1/2 *(eta_x_l^2)...

```



```

        *(eta_y_1^2) -1/3*(eta_y_1^2) + 1/6*(eta_x_1^2);
y_l_32 = 1/3*(eta_x_1)*(eta_y_1^2) - 2/3*(eta_x_1^3) -1/3*(eta_x_1);
y_l_33 = 4/3*(eta_y_1^3) + 1/3*(eta_x_1^2)*(eta_y_1) + 2/3*(eta_y_1) ;

%-----

Ry_1 = [          0,          0,          1;
        y_l_21,    y_l_22,    y_l_23;
        y_l_31,    y_l_32,    y_l_33];

ff = get_cc(Ry_1);

%-----

b0 = y0(i+1,j)- y0(i,j);   h0 = y0(i,j)- y0(i-1,j) ;
b1 = y1(i+1,j)- y1(i,j);   h1 = y1(i,j)- y1(i-1,j) ;
b2 = y2(i+1,j)- y2(i,j);   h2 = y2(i,j)- y2(i-1,j) ;

d0 = y0(i,j+1)- y0(i,j);   k0 = y0(i,j)- y0(i,j-1) ;
d1 = y1(i,j+1)- y1(i,j);   k1 = y1(i,j)- y1(i,j-1) ;
d2 = y2(i,j+1)- y2(i,j);   k2 = y2(i,j)- y2(i,j-1) ;

%---- Governing equation begin -----

u0(i,j) = p.*y0(i,j) - r*(y1(i+1,j)-y1(i-1,j))...
+ r*(cc(1,1)*b0+cc(1,2)*b1+cc(1,3)*b2)...
- r*(dd(1,1)*h0+dd(1,2)*h1+dd(1,3)*h2)...
- r*(y2(i,j+1)-y2(i,j-1))...
+ r*(gg(1,1)*d0+gg(1,2)*d1+gg(1,3)*d2)...
- r*(ff(1,1)*k0+ff(1,2)*k1+ff(1,3)*k2);

u1(i,j) = q.*y1(i,j) - r*( (a(i+1,j)+ 0.5*(eta_x(i+1,j)^2)...
*((eta_x(i+1,j)^2 + eta_y(i+1,j)^2)+1)...
*y0(i+1,j)) - (a(i-1,j)+ 0.5*(eta_x(i-1,j)^2)...
*((eta_x(i-1,j)^2 + eta_y(i-1,j)^2)+1)*y0(i-1,j)))...
+ r*(cc(2,1)*b0+cc(2,2)*b1+cc(2,3)*b2)...
- r*(dd(2,1)*h0+dd(2,2)*h1+dd(2,3)*h2)...
- r*((0.5*(eta_x(i,j+1)*eta_y(i,j+1))...
*((eta_x(i,j+1)^2 + eta_y(i,j+1)^2)+1)*y0(i,j+1))...
-(0.5*(eta_x(i,j-1)*eta_y(i,j-1))...
*((eta_x(i,j-1)^2 + eta_y(i,j-1)^2)+1)*y0(i,j-1)) )...
+ r*(gg(2,1)*d0+gg(2,2)*d1+gg(2,3)*d2)...
- r*(ff(2,1)*k0+ff(2,2)*k1+ff(2,3)*k2);

u2(i,j) = q.*y2(i,j) - r*((0.5*(eta_x(i+1,j)*eta_y(i+1,j))...
*((eta_x(i+1,j)^2 + eta_y(i+1,j)^2)+1)*y0(i+1,j))...
-(0.5*(eta_x(i-1,j)*eta_y(i-1,j))...
*((eta_x(i-1,j)^2 + eta_y(i-1,j)^2)+1)*y0(i-1,j)))...
+ r*(cc(3,1)*b0+cc(3,2)*b1+cc(3,3)*b2)...
- r*(dd(3,1)*h0+dd(3,2)*h1+dd(3,3)*h2)...
- r*( (a(i,j+1)+ 0.5*(eta_y(i,j+1)^2)...
*((eta_x(i,j+1)^2 + eta_y(i,j+1)^2)+1)*y0(i,j+1))...
-(a(i,j-1)+ 0.5*(eta_y(i,j-1)^2)...
*((eta_x(i,j-1)^2 + eta_y(i,j-1)^2)+1)*y0(i,j-1)))...
+ r*(gg(3,1)*d0+gg(3,2)*d1+gg(3,3)*d2)...
- r*(ff(3,1)*k0+ff(3,2)*k1+ff(3,3)*k2);

%---- Governing equation end -----

end
end

% End cell iteration

subplot(2,1,1),surf(x,z,u0); shading interp;
subplot(2,1,2),plot(x,u0((nz+1)./2,:), 'linewidth',1);

pause(dt);
time = time + dt ; % time update

```

```
end

% End time iteration

% Begin Riemann dissipation terms from |Roe|
function k = get_cc(A)
[R,DR] = eig(A); %disp([Ax, D,R]);
k = (R * abs(DR)) / R;
% End Riemann dissipation terms from |Roe|
```