A FAST MULTIGRID ALGORITHM FOR ISOTROPIC TRANSPORT PROBLEMS II: WITH ABSORPTION *

T. MANTEUFFEL †, S. MCCORMICK †, J. MOREL ‡, AND G. YANG §

Abstract

A multigrid method for solving the 1-D slab-geometry S_N equations with isotropic scattering and absorption is presented. The case with no absorption was treated in part I of this paper [10]. Relaxation is based on a two-cell inversion, which is very efficient because it takes advantage of the structure of the two-cell problem. For interpolation we use kinked linear elements. The kink is based on the amount of absorption present. The restriction operator is full weighting. Numerical results show this algorithm to be faster than DSA in all regimes. This scheme is also well-suited for massively parallel computer architectures.

§1. Introduction

In this paper we describe a fast method for solving the equations used to model the transport of neutral particles with isotropic scattering in slab geometry. These problems are important in many applications such as nuclear reactor design, radiation therapy in medical science, radiation effects on global weather, and they are a fundamental part of many algorithms used to model more complicated applications such as coupled photon/electron scattering used to model satellite electronics shielding.

We focus on the solution of the steady-state, monoenergetic, linear Boltzmann equation in slab geometry. The need to solve such equations arises from the time-dependent Boltzmann equations used in the applications mentioned above. The physical domain is assumed to be a semi-infinite slab of width $b-a$ in the x dimension. Although three dimensional, we assume that the flux of particles is independent of the y and z coordinates. Let $\psi(x,\ \mu,\ e,\ t)$ represent the flux of particles at a position x, traveling at an angle μ to the x-axis, with energy e, at time t. The Boltzmann equation takes the form

$$\frac{\partial\psi}{\partial \mathrm{t}} = -\mu\frac{\partial\psi}{\partial \mathrm{x}} - \sigma_{\mathrm{t}}(\mathrm{x},\,\mathrm{e})\psi + \int\!\!\int \mathrm{K}(\mu'\to\mu,\,\mathrm{e}'\to\mathrm{e})\psi(\mathrm{x},\,\mu',\,\mathrm{e}',\,\mathrm{t})d\mu' d\mathrm{e}' + \mathrm{q}(\mathrm{x},\,\mu,\,\mathrm{e},\,\mathrm{t}) \tag{1.1}$$

Here, σ_t is the collision cross section and $\int_{x_0}^{x_1}\sigma_t dx$ represents the expected number of collisions that a particle will experience in the interval (x_0, x_1). In the same manner, the scattering kernel $\int\int K(\mu'\to\mu,\ e'\to e)$ represents the expected number of particles that will scatter from direction μ' and energy e' to direction μ and energy e. Finally, $q(x,\ \mu,\ e,\ t)$ represents particle sources.

---

* This work was supported by AFOSR under grant AFOSR 91-0156, the NSF under grant DMS-8704169,the DOE under grant DE-FG03-93ER25165 and Los Alamos National Laboratory.

† Program in Applied Mathematics, University of Colorado at Boulder, Boulder, CO

‡ Computer Research Group (C-3), Los Alamos National Laboratory, Los Alamos, NM,

§ Center for Computational Mathematics, University of Colorado of Denver, Denver, CO

The above equation is usually integrated with implicit time stepping methods and the energy variable is treated with a piecewise-constant finite element method to yield a coupled system of equations at each time step. Let $\psi^k(x, \mu)$ be the flux of particles with energy e_k. Then

$$\mu \frac{\partial \psi^k}{\partial x} + \sigma_t^k(x)\psi^k = \sum_j \int K_{k,j}(\mu' \to \mu)\psi^j(x, \mu')d\mu' + q^k(x, \mu), \tag{1.2}$$

where the time step information has been incorporated into σ_t^k and q^k, and $\int K_{k,j}(\mu' \to \mu)$ represents the expected number of collisions that will rescatter a particle from energy e_j into energy e_k.

Since particles generally lose energy, the system of equations is solved by a block Gauss/Seidel type iteration. Starting with the highest energy level, $\psi^k(x, \mu)$ is updated assuming that $\psi^j(x, \mu)$ for $j \neq k$ are known. This yields the equation

$$\mu \frac{\partial \psi^k}{\partial x} + \sigma_t^k(x)\psi^k = \int K_{k,k}(\mu' \to \mu)\psi^k(x, \mu')d\mu' + \hat{q}^k(x, \mu), \tag{1.3}$$

where now $\hat{q}^K$ contains all the inscatter from other energy levels.

The kernel $K_{k,k}$ is often mildly anisotropic, becoming more isotropic as energy decreases. In this paper we focus on the isotropic form of (1.3):

$$\mu \frac{\partial \psi}{\partial x} + \sigma_t \psi = \frac{\sigma_s}{2} \int_{-1}^{1} \psi(x, \mu')d\mu' + q(x, \mu). \tag{1.4}$$

Here, σ_s is now the scattering cross section and represents the expected number of collisions that result in a rescatter, while $\sigma_a = \sigma_t - \sigma_s$ represents the expected number of collisions that result in an absorption. When $\sigma_t = \sigma_s$, there is no absorption. The equation is well defined if the flux of particles entering the slab is given as boundary conditions:

$$\psi(a, \mu) = g_a(\mu), \quad \psi(b, -\mu) = g_b(\mu) \quad \mu\epsilon(0, 1). \tag{1.5}$$

In general, numerical methods that are effective for isotropic equations also work well for mildly anisotropic equations. We believe that this will be especially true for the multigrid algorithm described below because of the special form of relaxation used (see Section 3).

Problem (1.4) will inherit the discretization scheme from problem (1.1). The standard approach is to expand the angular dependence in terms of the first N Legendre polynomials and close the system with a Galerkin condition. This is attractive because the Legendre polynomials are the eigenvectors of the scattering operators $K_{k,j}$ in slab geometry. This results in the P_{N-1} discretization. In slab geometry, this is equivalent to collocating at Gauss quadrature points which is known as the discrete ordinates or S_N discretization (c.f. [7]).

Spatial discretization must take into account the important optically dense or thick limit, in which the problem becomes ill-conditioned. Physically, this corresponds to a mean-free-path between collisions that is small compared to the width of the slab and materials that allow very little absorption. Mathematically, we have

$$\sigma_t \to \infty, \quad \frac{\sigma_s}{\sigma_t} \to 1. \tag{1.6}$$

Dividing (1.1) by σ_t and taking the limits in (1.4) yields

$$\psi(x,\mu) = \frac{1}{2}\int_{-1}^{1} \psi(x,\mu')d\mu', \tag{1.7}$$

which admits any $\psi(x,\mu)$ that is independent of μ. Thus, in this limit, (1.4) is singularly perturbed with a large near null space. For discrete equations it is the product $\sigma_t h_i$, where h_i is a mesh parameter, that parameterizes each equation. There are two quantities that determine the character of the discrete problem: the overall thickness, $\sigma_t(b-a)$, and the thickness of individual cells, $\sigma_t h_i$.

Some spatial discretization schemes, for example up-wind differences, have discretization error $O(\sigma_t h_i)$. Thus, for $\sigma_t \gg 1$ these schemes require $h_i \to 0$ even for well behaved solutions. For a discussion of this issue see [10] and Larsen and Morel[6]. One difference scheme that behaves well in the thick limit is the Modified Linear Discontinuous scheme(MLD)([6]). Not only does it give the proper behavior in the thick limit, but it is very accurate. In [10], we developed a very efficient multigrid algorithm to solve the the discrete equations produced by the Modified Linear Discontinuous scheme in the absence of absorption ($\sigma_t = \sigma_s$). Our analysis shows that, in this case, a V(1,1) cycle version of our algorithm yields a convergence factor ρ bounded as follows:

a) for $\max\limits_i\,(\sigma_t h_i) \ll 1, \quad \rho = O(\max\limits_i\,(\sigma_t h_i)^2)$,
b) for $\min\limits_i\,(\sigma_t h_i) \gg 1, \quad \rho = O(\max\limits_i\,(\frac{1}{\sigma_t h_i}))$.

Numerical results are even better than these theoretical estimates. Experiments yield the following convergence factor, ρ, for a V(1,1) cycle:

a) for $\max\limits_i\,(\sigma_t h_i) \ll 1, \quad \rho = O(\max\limits_i\,(\sigma_t h_i)^3)$,
b) for $\min\limits_i\,(\sigma_t h_i) \gg 1, \quad \rho = O(\frac{1}{\min\limits_i\,(\sigma_t h_i)^2})$.

When there is absorption in the transport equations ($\sigma_t > \sigma_s$) and $\sigma_t h \gg 1$, the two-cell μ-line relaxation used in [10] will leave an error that is essentially independent of angle and continuous but no longer linear across two cells (see Section 3 in [10]). The deviation of the errors from linearity depends on the value of $\gamma = \frac{\sigma_s}{\sigma_t}$ and h. Because the errors after relaxation will not be linear across two-cell pairs, the linear interpolation used in our previous multigrid algorithm is unsuitable. In this paper, we will present a new method which uses kinked elements, whose shape will be determined by the severity of deviation from linearity of the errors after relaxation. When $\gamma = 1$, the kinked element will be linear.

We remark that for small values of the angular discretization dimension , N, the discrete form of (1.4) could easily be solved by direct inversion of the resulting banded matrix. However, direct methods require $O(N^2m)$ operations and $O(N^2m)$ storage, where m is the number of spatial cells. The method described here, as well as the Diffusion Synthetic Acceleration (DSA) [2, 5] to which we compare our method, both require only $O(Nm)$ operations and storage. Thus, these methods are viable for problems that require fine angular resolution. Because of the large number of times equations of the form (1.4) must be solved, efficiency is critical. Moreover, both DSA and our method have obvious generalization to higher spatial dimensions.

In Section 2, we analyze the properties of relaxation when $\gamma \neq 1$. In Section 3, we introduce the kinked element, relaxation-induced interpolation and coarse grid operators used to solve the transport problems when $\gamma \neq 1$. The multigrid algorithm is discussed in Section 4. In Section 5, we examine the properties of the new interpolation. In Section 6, computational results are presented. We show that the multigrid algorithm is faster than the Diffusion Synthetic Acceleration algorithm (DSA) [2, 5] on a representative set of test problems. We include a study of domains with inhomogeneous material properties. Finally, in Section 7, we present our conclusions.

§2. The Discrete Model

We start with an S_N approximation in angle. This corresponds to expanding the angle dependence in terms of the first N Legendre polynomials. Closing the equations with a Galerkin formulation yields a system that resembles collocation at the Gauss quadrature points $\mu_j, j = 1, ..., N$. Similarly, the integral is replaced with a quadrature formula using the Gauss quadrature weights $\omega_j, j = 1, ..., N$. By symmetry, we may denote the positive quadrature points $\mu_j > 0, j = 1, ..., n$, with corresponding weights $\omega_j, j > 1, ..., n$, and negative quadrature points -$\mu_j < 0, j = 1, ..., n$ with corresponding weights $\omega_j, j = 1, ..., n$, where $n = N/2$. We use $\psi^+(x_i, \mu_j) = \psi(x_i, \mu_j)$, $\psi^-(x_i, \mu_j) = \psi(x_i, -\mu_j)$ to denote flux variables at spatial point x_i, angular position μ_j at positive and negative directions respectively.

The spatial difference scheme is the Modified Linear Discontinuous discretization(MLD). Consider the grid $a = x_{1/2} < x_{3/2} < \ldots < x_{m+1/2} = b$, where $x_{i\pm 1/2}$ are cell edges and $x_i = (x_{i-1/2}+x_{i+1/2})/2$ is the cell center. Let $\underline{\psi}_i^+ = (\psi^+(x_i, \mu_1), ..., \psi^+(x_i, \mu_n))^T$ and $\underline{\psi}_i^- = (\psi^-(x_i, \mu_1), ..., \psi^-(x_i, \mu_n))^T$. In matrix form MLD can be written as (for a complete presentation see [6], [10])

$$B_i(\underline{\psi}^+_{i+\frac{1}{2}} - \underline{\psi}^+_{i-\frac{1}{2}}) + \underline{\psi}^+_i = \gamma R(\underline{\psi}^+_i + \underline{\psi}^-_i) + \underline{q}^+_i, \tag{2.1a}$$

$$2B_i(\underline{\psi}^+_{i+\frac{1}{2}} - \underline{\psi}^+_i) + \underline{\psi}^+_{i+\frac{1}{2}} = \gamma R(\underline{\psi}^+_{i+\frac{1}{2}} + 2\underline{\psi}^-_i - \underline{\psi}^-_{i-\frac{1}{2}}) + \underline{q}^+_{i+\frac{1}{2}}, \tag{2.1b}$$

for i = 1,...,m, and

$$B_i(\underline{\psi}^-_{i-\frac{1}{2}} - \underline{\psi}^-_{i+\frac{1}{2}}) + \underline{\psi}^-_i = \gamma R(\underline{\psi}^+_i + \underline{\psi}^-_i) + \underline{q}^-_i, \tag{2.1c}$$

$$2B_i(\underline{\psi}^-_{i-\frac{1}{2}} - \underline{\psi}^-_i) + \underline{\psi}^-_{i-\frac{1}{2}} = \gamma R(\underline{\psi}^-_{i-\frac{1}{2}} + 2\underline{\psi}^+_i - \underline{\psi}^+_{i+\frac{1}{2}}) + \underline{q}^-_{i-\frac{1}{2}}, \tag{2.1d}$$

for $i = 1, ..., m$, with boundary conditions

$$\underline{\psi}^+_{\frac{1}{2}} = \underline{g}^+_a, \quad \underline{\psi}^-_{m+\frac{1}{2}} = \underline{g}^-_b, \tag{2.1e}$$

where B_i and R are the $n \times n$ matrices

$$B_i = \begin{bmatrix} \frac{\mu_1}{\sigma_t h_i} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{\mu_n}{\sigma_t h_i} \end{bmatrix}, \quad R = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \begin{bmatrix} \omega_1 & \cdots & \omega_n \end{bmatrix} = \underline{1\omega}^T. \tag{2.1f}$$

In block matrix form, (2.1) can be written as
(2.2)

$$\begin{bmatrix} I + 2B_i - \gamma R & -2\gamma R & -2B_i & \gamma R \\ 0 & I - \gamma R & -\gamma R & B_i \\ B_i & -\gamma R & I - \gamma R & 0 \\ 0 & -2B_i & -2\gamma R & I + 2B_i - \gamma R \end{bmatrix} \begin{bmatrix} \underline{\psi}^-_{i-\frac{1}{2}} \\ \underline{\psi}^+_i \\ \underline{\psi}^-_i \\ \underline{\psi}^+_{i+\frac{1}{2}} \end{bmatrix} = \begin{bmatrix} \underline{q}^-_{i-\frac{1}{2}} \\ B_i \underline{\psi}^+_{i-\frac{1}{2}} + \underline{q}^+_i \\ B_i \underline{\psi}^-_{i+\frac{1}{2}} + \underline{q}^-_i \\ \underline{q}^+_{i+\frac{1}{2}} \end{bmatrix}.$$

In this paper, we will use an edge/edge notation for the unknown flux variable $\underline{\psi}$. That is, the flux within cell i is linear and is determined by the value at the right and left sides,ψ^+_{ir} and ψ^+_{il}. We define the transformation

$$\begin{bmatrix} \underline{\psi}^-_{il} \\ \underline{\psi}^+_{il} \\ \underline{\psi}^-_{ir} \\ \underline{\psi}^+_{ir} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & -1 \\ -1 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \underline{\psi}^-_{i-\frac{1}{2}} \\ \underline{\psi}^+_i \\ \underline{\psi}^-_i \\ \underline{\psi}^+_{i+\frac{1}{2}} \end{bmatrix}. \tag{2.3a}$$

Then the inverse transformation of (2.3a) will be

$$\begin{bmatrix} \underline{\psi}^-_{i-\frac{1}{2}} \\ \underline{\psi}^+_i \\ \underline{\psi}^-_i \\ \underline{\psi}^+_{i+\frac{1}{2}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \underline{\psi}^-_{il} \\ \underline{\psi}^+_{il} \\ \underline{\psi}^-_{ir} \\ \underline{\psi}^+_{ir} \end{bmatrix}. \tag{2.3b}$$

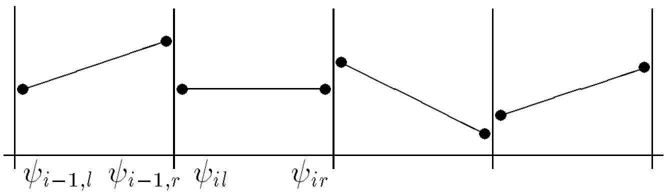

Figure 1.1 Piecewise Discontinuous Elements

Figure 1.1 shows the piecewise discontinuous elements we use. At cell i, two variables ψ_{il}, ψ_{ir} are defined.

By substituting (2.3b) into (2.2) and multiplying both sides of (2.2) by the transformation matrix of (2.3a), we then can write (2.2) in block matrix form as

$$
(2.4) \qquad \begin{bmatrix} A_1 & -C_1 & & & & \\ -D_2 & A_2 & -C_2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & -D_i & A_i & -C_i & \\ & & & \ddots & \ddots & \ddots \\ & & & & -D_m & A_m \end{bmatrix} \begin{bmatrix} \underline{\Psi}_1 \\ \underline{\Psi}_2 \\ \vdots \\ \underline{\Psi}_i \\ \vdots \\ \underline{\Psi}_m \end{bmatrix} = \begin{bmatrix} \underline{Q} \end{bmatrix},
$$

where $\underline{Q}$ is the right-hand side and

$$
(2.5a) \qquad A_i = \begin{bmatrix} I + B_i - \gamma R & -\gamma R & -B_i & 0 \\ -\gamma R & I + B_i - \gamma R & 0 & B_i \\ B_i & 0 & I + B_i - \gamma R & -\gamma R \\ 0 & -B_i & -\gamma R & I + B_i - \gamma R \end{bmatrix},
$$

$i = 1, ..., m$,

$$
(2.5b) \qquad C_i = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 2B_i & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, i = 1, ..., m-1,
$$

$$
(2.5c) \qquad D_i = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2B_i \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, i = 2, ..., m,
$$

$$
(2.5d) \qquad \underline{\Psi}_i = (\underline{\psi}_{il}^-, \underline{\psi}_{il}^+, \underline{\psi}_{ir}^-, \underline{\psi}_{ir}^+)^T, i = 1, ..., m.
$$

In the following section, we will examine the two-cell μ-line relaxation and its properties.

§3 Relaxation

By two-cell red-black μ-line relaxation, we mean that at cells $2i-1$ and $2i$, we set $\underline{\psi}_{(2i-2)r}^+$ and $\underline{\psi}_{(2i+1)l}^-$ as boundary values and solve for all other interior values in cells $2i-1$ and $2i$ simultaneously. This relaxation is carried out in a red-black ordering. Since relaxation uses a red-black ordering, we can look at each two-cell pair individually. For a two-cell pair, for example cells $2i-1$ and $2i$, the errors after the relaxation will be

$$\begin{bmatrix} \underline{e}^-_{(2i-1)l} \\ \underline{e}^+_{(2i-1)l} \\ \underline{e}^-_{(2i-1)r} \\ \underline{e}^+_{(2i-1)r} \\ \underline{e}^-_{(2i)l} \\ \underline{e}^+_{(2i)l} \\ \underline{e}^-_{(2i)r} \\ \underline{e}^+_{(2i)r} \end{bmatrix} = \begin{bmatrix} A_{2i-1} & -C_{2i-1} \\ -D_{2i} & A_{2i} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 2B_{2i-1}\hat{\underline{e}}^+_{(2i-2)r} \\ 0 \\ 0 \\ 0 \\ 0 \\ 2B_{2i}\hat{\underline{e}}^-_{(2i+1)l} \\ 0 \end{bmatrix}, \tag{3.1}$$

where $\hat{\underline{e}}$ on the right-hand side is the error at the outside border of the two-cell pair before relaxation. As we have shown in our previous paper(Manteuffel et.al. [10]), the inversion of the $\begin{bmatrix} A_{2i-1} & -C_{2i-1} \\ -D_{2i} & A_{2i} \end{bmatrix}$ can be performed in O(n) operations by using the Sherman-Morrison formula. In [10], only the case $\gamma = 1$ was examined. The following result for $\gamma = 1$ was proved.

THEOREM 1. *Suppose $\gamma = 1$ in (3.1) and $\min(\sigma_t h_{2i-1}, \sigma_t h_{2i}) \gg 1$; then, the errors after the two-cell red-black μ-line relaxation will be independent of angle, continuous and piecewise linear up to accuracy of $O(\max(\frac{1}{\sigma_t h_{2i-1}}, \frac{1}{\sigma_t h_{2i}}))$.*

Proof: See Theorem 3 in [10].

In this paper, we will discuss the MLD equation with absorption($\gamma \neq 1$). In the thin limit, γ will not affect the behavior of μ-line relaxation. The proof of Theorem 2 that appears in [10] does not depend on γ. We restate Theorem 2 in [10] for $\gamma \leq 1$ in the following way:

THEOREM 2. *Suppose $\gamma \leq 1$ and $\max(\sigma_t h_{2i-1}, \sigma_t h_{2i}) \ll 1$; then, the errors after two-cell red-black μ-line relaxation will be continuous and piecewise linear across two cells up to accuracy of $O(\max(\sigma_t^2 h_{2i-1}^2, \sigma_t^2 h_{2i}^2))$.*

Proof: See the proof of Theorem 2 in [10].

In the thick limit, the size of $1-\gamma$ relative to $\frac{1}{\sigma_t h_{2i-1}}$ and $\frac{1}{\sigma_t h_{2i}}$ will determine the effectiveness of μ-line relaxation. For simplicity, we restrict our analysis to uniform $\sigma_t h_i$ and let h be the uniform mesh size. The extension to nonuniform meshes is straightforward, but messy. We will analyze the μ-line relaxation when

$a)$ $1-\gamma \gg \frac{1}{\sigma_t h}$
$b)$ $1-\gamma = \frac{1}{(\sigma_t h)^p}$, p=1,2,3.

In the second case, when $\sigma_t h \gg 1$, the larger p is, the closer γ is to 1.

THEOREM 3. *When $1-\gamma \gg \frac{1}{\sigma_t h}$ and $\sigma_t h \gg 1$, the two-cell μ-line relaxation alone will reduce errors at any two neighboring cells $2i-1, 2i$ by a factor of $O(\frac{1}{\sigma_t h})$.*

Proof: When $\gamma \neq 1$,

$$
\begin{bmatrix} I-\gamma R & -\gamma R \\ -\gamma R & I-\gamma R \end{bmatrix}^{-1} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} + \frac{\gamma}{1-\gamma} \begin{bmatrix} R & R \\ R & R \end{bmatrix}. \tag{3.2}
$$

Write

(3.3)

$$
\begin{bmatrix} A_{2i-1} & -C_{2i-1} \\ -D_{2i} & A_{2i} \end{bmatrix} = \begin{bmatrix}
I-\gamma R & -\gamma R & & & & & & \\
-\gamma R & I-\gamma R & & & & & & \\
& & I-\gamma R & -\gamma R & & & & \\
& & -\gamma R & I-\gamma R & & & & \\
& & & & I-\gamma R & -\gamma R & & \\
& & & & -\gamma R & I-\gamma R & & \\
& & & & & & I-\gamma R & -\gamma R \\
& & & & & & -\gamma R & I-\gamma R
\end{bmatrix}
$$

$$
- \begin{bmatrix}
B & 0 & -B & 0 & 0 & 0 & 0 & 0 \\
0 & B & 0 & B & 0 & 0 & 0 & 0 \\
B & 0 & B & 0 & -2B & 0 & 0 & 0 \\
0 & -B & 0 & B & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & B & 0 & -B & 0 \\
0 & 0 & 0 & -2B & 0 & B & 0 & B \\
0 & 0 & 0 & 0 & B & 0 & B & 0 \\
0 & 0 & 0 & 0 & 0 & -B & 0 & B
\end{bmatrix} = H_0 - H_1,
$$

where H_0 is the first matrix and H_1 the second in (3.3). By (3.2), H_0 is not singular, so

$$
H_0^{-1} H_1 = \frac{1}{\sigma_t h} \begin{bmatrix}
M & 0 & -M & 0 & 0 & 0 & 0 & 0 \\
0 & M & 0 & M & 0 & 0 & 0 & 0 \\
M & 0 & M & 0 & -2M & 0 & 0 & 0 \\
0 & -M & 0 & M & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & M & 0 & -M & 0 \\
0 & 0 & 0 & -2M & 0 & M & 0 & M \\
0 & 0 & 0 & 0 & M & 0 & M & 0 \\
0 & 0 & 0 & 0 & 0 & -M & 0 & M
\end{bmatrix} \tag{3.4}
$$

$$
+ \frac{\gamma}{(1-\gamma)\sigma_t h} \begin{bmatrix}
RM & RM & -RM & RM & 0 & 0 & 0 & 0 \\
RM & RM & -RM & RM & 0 & 0 & 0 & 0 \\
RM & -RM & RM & RM & -2RM & 0 & 0 & 0 \\
RM & -RM & RM & RM & -2RM & 0 & 0 & 0 \\
0 & 0 & 0 & -2RM & RM & RM & -RM & RM \\
0 & 0 & 0 & -2RM & RM & RM & -RM & RM \\
0 & 0 & 0 & 0 & RM & -RM & RM & RM \\
0 & 0 & 0 & 0 & RM & -RM & RM & RM
\end{bmatrix},
$$

where $M = \begin{bmatrix} \mu_1 & & \\ & \ddots & \\ & & \mu_n \end{bmatrix}$.

When $\sigma_t h \gg 1$ and $(1-\gamma) \gg \frac{1}{\sigma_t h}$, then $(1-\gamma)\sigma_t h \gg 1$ and we can find a constant c such that

$$||H_0^{-1}H_1||_2 \le \frac{c}{\sigma_t h}. \tag{3.5}$$

So

$$(I - H_0^{-1}H_1)^{-1} = I + H_0^{-1}H_1 + O(\frac{1}{\sigma_t^2 h^2}). \tag{3.6}$$

Since H_0^{-1} is constant, we have

(3.7)

$$\begin{bmatrix} A_{2i-1} & -C_{2i-1} \\ -D_{2i} & A_{2i} \end{bmatrix}^{-1} = (I - H_0^{-1}H_1)^{-1}H_0^{-1} = H_0^{-1} + H_0^{-1}H_1H_0^{-1} + O(\frac{1}{\sigma_t^2 h^2}) = H_0^{-1} + O(\frac{1}{\sigma_t h}).$$

From (3.2) and (3.3), we have

(3.8 $$||H_0^{-1}||_2 \le 1 + \frac{\gamma}{1-\gamma} = \frac{1}{1-\gamma}.$$

Therefore, by (3.8), we can find a constant c such that

$$||\begin{bmatrix} A_{2i-1} & -C_{2i-1} \\ -D_{2i} & A_{2i} \end{bmatrix}^{-1}||_2 \le c. \tag{3.9}$$

Substituting (3.9) into (3.1) and noting that $B_i = \frac{1}{\sigma_t h}M$, we have

$$||(e^-_{(2i-1)l}, e^+_{(2i-1)l}, e^-_{(2i-1)r}, e^+_{(2i-1)r}, e^-_{(2i)l}, e^+_{(2i)l}, e^-_{(2i)r}, e^+_{(2i)r})^T||_2 \le c\frac{1}{\sigma_t h}. \tag{3.10}$$

This completes the proof. $\square$

When $\gamma = 1 - \frac{1}{(\sigma_t h)^p}$ and $\sigma_t h \gg 1$, p=1,2,3, the two-cell μ-line relaxation will produce errors that are continuous and essentially independent of angle but kinked across two cells. To show this behavior, we rewrite (3.1) as

(3.11)

$$\begin{bmatrix} \underline{e}^-_{(2i-1)l} \\ \underline{e}^+_{(2i-1)l} \\ \underline{e}^-_{(2i-1)r} \\ \underline{e}^+_{(2i-1)r} \\ \underline{e}^-_{(2i)l} \\ \underline{e}^+_{(2i)l} \\ \underline{e}^-_{(2i)r} \\ \underline{e}^+_{(2i)r} \end{bmatrix} = \begin{bmatrix} A_{2i-1} & -C_{2i-1} \\ -D_{2i} & A_{2i} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 2B_{2i-1} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \underline{e}^+_{(2i-2)r} + \begin{bmatrix} A_{2i-1} & -C_{2i-1} \\ -D_{2i} & A_{2i} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 2B_{2i} \\ 0 \end{bmatrix} \underline{e}^-_{(2i+1)l}.$$

When $\sigma_t h \gg 1$ and $\gamma = 1 - \frac{1}{(\sigma_t h)^p}$, we can expand matrix $\begin{bmatrix} A_{2i-1} & -C_{2i-1} \\ -D_{2i} & A_{2i} \end{bmatrix}^{-1}$ into Taylor series as $O(\sigma_t h) + O(1) + O(\frac{1}{\sigma_t h})$. We omit the details of the expansion because of the complexity. Let $\delta = \frac{1}{\sigma_t^p h^p}$ and $\epsilon = \frac{1}{\sigma_t h}$. After the expansion, we have

$$\begin{bmatrix} A_{2i-1} & -C_{2i-1} \\ -D_{2i} & A_{2i} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 2B_{2i-1} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.12}$$

$$= \frac{\xi}{\sigma_t h} \begin{bmatrix} 2RM \\ 2RM \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \frac{2\xi^2}{\sigma_t^3 h^3} \begin{bmatrix} 0 \\ 0 \\ \frac{4\xi c_0}{\sigma_t h} R \\ \frac{4\xi c_0}{\sigma_t h} R \\ 2R \\ 2R \\ 0 \\ 0 \end{bmatrix} P(c_0 M^2 R + c_1 M R) + O(\frac{1}{\sigma_t h}),$$

where
(3.13)

$$\xi = \frac{1}{1-\zeta}, \quad \zeta = \gamma \sum_{j=1}^{n} \left(\frac{2\omega_j (1 + \frac{\mu_j}{\sigma_t h})}{1 + 2\frac{\mu_j}{\sigma_t h} + 2(\frac{\mu_j}{\sigma_t h})^2}\right), \quad c_0 = \sum_{j=1}^{n} \omega_j \mu_j, \quad c_1 = \sum_{j=1}^{n} \omega_j \mu_j^2 = \frac{1}{6}$$

and

$$P = (1 - \frac{2\epsilon}{\delta + 4c_0\epsilon} MR)(1 + \frac{2\epsilon}{\delta + 2c_1\epsilon^2} MR)(1 - \frac{2\epsilon^2(\delta + 2\epsilon c_0)}{8c_0 c_1 \epsilon^3 + 2c_0\delta\epsilon + \delta^2} M^2 R). \tag{3.14}$$

Here, ξ is not expanded yet. Its expansion will depend on the power p in $\gamma = 1 - \frac{1}{(\sigma_t h)^p}$. We will discuss the three cases p=1,2,3 separately.

1) $p = 1$

When $\sigma_t h \gg 1$, ζ in (3.13) can be expanded as

$$\zeta = \gamma \sum_{j=1}^{n} \left(\frac{2\omega_j (1 + \frac{\mu_j}{\sigma_t h})}{1 + 2\frac{\mu_j}{\sigma_t h} + 2(\frac{\mu_j}{\sigma_t h})^2}\right) \tag{3.15}$$

$$= \gamma \sum_{j=1}^{n} 2\omega_j \Big(1 + \frac{\mu_j}{\sigma_t h}\Big)\Big(1 - 2\frac{\mu_j}{\sigma_t h} + O\Big(\frac{1}{\sigma_t^2 h^2}\Big)\Big) = \gamma\Big(1 - 2\frac{c_0}{\sigma_t h}\Big) + O\Big(\frac{1}{\sigma^2 h^2}\Big).$$

So, by (3.15), ξ in (3.13) can be expanded as

$$\xi = \frac{1}{1-\zeta} = \frac{1}{1 - \gamma + \frac{2c_0\gamma}{\sigma_t h} + O(\frac{1}{\sigma_t^2 h^2})} = \frac{\sigma_t h}{1 + 2c_0} + O\Big(\frac{1}{\sigma_t^2 h^2}\Big). \tag{3.16}$$

When $p = 1$, P of (3.14) can be expressed as

$$P = (1 + MR)\Big(1 - \frac{2}{1 + 4c_0} MR\Big) + O\Big(\frac{1}{\sigma_t h}\Big) \tag{3.17}$$

By (3.16) and (3.17), the norm of the second term in (3.12) is $O(\frac{1}{\sigma_t h})$. So

$$\begin{bmatrix} A_{2i-1} & -C_{2i-1} \\ -D_{2i} & A_{2i} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 2B_{2i-1} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \frac{1}{1+2c_0} \begin{bmatrix} 2RM \\ 2RM \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + O\Big(\frac{1}{\sigma_t h}\Big). \tag{3.18}$$

Similarly, we have

$$\begin{bmatrix} A_{2i-1} & -C_{2i-1} \\ -D_{2i} & A_{2i} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 2B_{2i} \\ 0 \end{bmatrix} = \frac{1}{1+2c_0} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 2RM \\ 2RM \end{bmatrix} + O\Big(\frac{1}{\sigma_t h}\Big). \tag{3.19}$$

By (3.18) and (3.19), the errors after relaxation will be essentially independent of angle but not piecewise linear. To graphically show this, we plot the errors after the μ-line relaxation in Figure 3.1. Figure 3.1a depicts the first term in (3.11) for positive angles. Figure 3.1b depicts the second term in (3.11) for positive angles. The real errors after relaxation should be a linear combination of Figure 3.1a and Figure 3.1b. The error distribution for negative angles will have the same pattern. From Figure 3.1 we see that, after relaxation, $\underline{e}^{+}_{(2i-1)r}$ equals $\underline{e}^{+}_{(2i)l}$ and $\underline{e}^{-}_{(2i-1)r}$ equals $\underline{e}^{-}_{(2i)l}$ up to accuracy of $O(\frac{1}{\sigma_t h})$. We can see that the error after relaxation is kinked and that linear interpolation will not be suitable for this case. However, regardless of the size of $\underline{e}^{+}_{(2i-2)r}$ and $\underline{e}^{+}_{(2i+1)l}$, the errors at the boundary between cell $2i-1$ and cell $2i$ are $\mathrm{O}(\frac{1}{\sigma_t h})$.

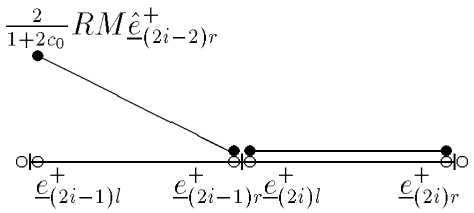

Figure 3.1a

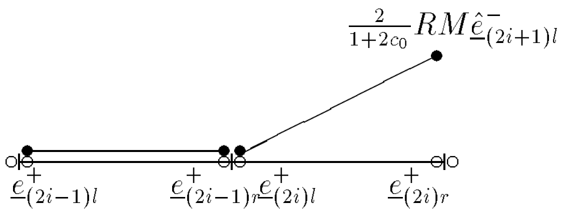

Figure 3.1b

2) p = 2

When $p = 2$ and $\sigma_t h \gg 1$, with ζ as in (3.15), ξ can be expanded as

$$\xi = \frac{1}{1-\zeta} = \frac{1}{1-\gamma+\frac{2c_0\gamma}{\sigma_t h}+O(\frac{1}{\sigma_t^2 h^2})} = \frac{\sigma_t h}{2c_0} + O(\frac{1}{\sigma_t h}). \tag{3.20}$$

and P in (3.14) be expanded as

$$P = \frac{\sigma_t h}{1+4c_1} MR + O(1). \tag{3.21}$$

Substituting (3.20) and (3.21) into (3.12), we have

$$\begin{bmatrix} A_{2i-1} & -C_{2i-1} \\ -D_{2i} & A_{2i} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 2B_{2i-1} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{c_0} RM \\ \frac{1}{c_0} RM \\ \frac{2c_1}{c_0(1+4c_1)} RM \\ \frac{2c_1}{c_0(1+4c_1)} RM \\ \frac{2c_1}{c_0(1+4c_1)} RM \\ \frac{2c_1}{c_0(1+4c_1)} RM \\ 0 \\ 0 \end{bmatrix} + O(\frac{1}{\sigma_t h}) \tag{3.22}$$

and

$$\begin{bmatrix} A_{2i-1} & -C_{2i-1} \\ -D_{2i} & A_{2i} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 2B_{2i+1} \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \frac{2c_1}{c_0(1+4c_1)} RM \\ \frac{2c_1}{c_0(1+4c_1)} RM \\ \frac{2c_1}{c_0(1+4c_1)} RM \\ \frac{2c_1}{c_0(1+4c_1)} RM \\ \frac{1}{c_0} RM \\ \frac{1}{c_0} RM \end{bmatrix} + O(\frac{1}{\sigma_t h}). \tag{3.23}$$

We plot (3.22) in Figure 3.2. The errors for negative angles are similar. We can see once again the errors after relaxation are essentially independent of angle and continuous across cells but kinked linear. In fact, they will be sublinear; that is, the error at the interface between cell $2i-1$ and cell $2i$ is less than the average of the errors at the edges of the two-cell pair.

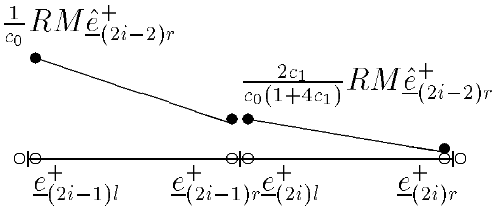

Figure 3.2a

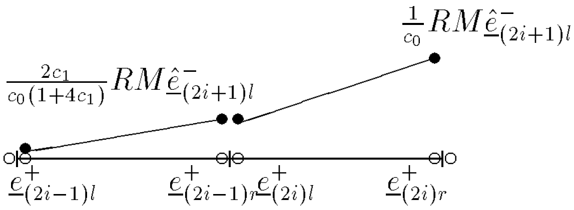

Figure 3.2b

3) p = 3

When $p=3$ and $\sigma_t h \gg 1$, with ζ in (3.15), ξ can be expanded as

$$\xi = \frac{1}{1-\zeta} = \frac{1}{1-\gamma+\frac{2c_0\gamma}{\sigma_t h}+O(\frac{1}{\sigma_t^2 h^2})} = \frac{\sigma_t h}{2c_0} + O(\frac{1}{\sigma_t^2 h^2}). \tag{3.24}$$

and P in (3.14) be expanded as

$$P = \frac{\sigma_t h}{4c_1}MR + O(1). \tag{3.25}$$

Substitute (3.24) and (3.25) into (3.12), we have

$$\begin{bmatrix} A_{2i-1} & -C_{2i-1} \\ -D_{2i} & A_{2i} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 2B_{2i-1} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{c_0}RM \\ \frac{1}{c_0}RM \\ \frac{1}{2c_0}RM \\ \frac{1}{2c_0}RM \\ \frac{1}{2c_0}RM \\ \frac{1}{2c_0}RM \\ 0 \\ 0 \end{bmatrix} + O(\frac{1}{\sigma_t h}) \tag{3.26}$$

and

$$
\begin{bmatrix} A_{2i-1} & -C_{2i-1} \\ -D_{2i} & A_{2i} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 2B_{2i+1} \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{2c_0}RM \\ \frac{1}{2c_0}RM \\ \frac{1}{2c_0}RM \\ \frac{1}{2c_0}RM \\ \frac{1}{c_0}RM \\ \frac{1}{c_0}RM \end{bmatrix} + O(\frac{1}{\sigma_t h}). \tag{3.27}
$$

We plot (3.26) in Figure 3.3. Again (3.27) is similar. In this case the error after relaxation is independent of angle, linear and continuous across a two-cell pair up to accuracy of $O(\frac{1}{\sigma_t h})$.

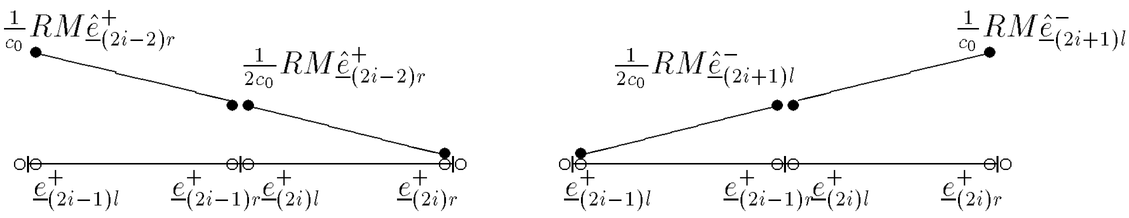

Figure 3.3a Figure 3.3b

In any case, after μ-line relaxation the error across a two-cell pair will be independent of angle, continuous and sublinear up to order $O(\frac{1}{\sigma_t h})$. The error at the boundary between cell $2i-1$ and cell $2i$ can be expressed as

$$
\underline{e}^{\pm}_{(2i-1)r} = \underline{e}^{\pm}_{(2i)l} + O(\frac{1}{\sigma_t h}) = (\frac{1}{2} - d)(\underline{e}^{\pm}_{(2i-1)l} + \underline{e}^{\pm}_{(2i)r}) + O(\frac{1}{\sigma_t h}) \tag{3.28}
$$

where $0 \leq d \leq \frac{1}{2}$ and $\pm$ denotes variables in either positive or negative directions. We summarize the above discussion in the following theorem.

THEOREM 4. *Suppose* $\gamma = 1 - \frac{1}{(\sigma_t h)^p}$ *and* $\sigma_t h \geq 1$. *After two-cell μ-line relaxation, the error across a two-cell pair will be independent of angle and continuous up to* $O(\frac{1}{\sigma_t h})$. *Further, the error at the boundary between the two-cell pair can be written as in (3.28), where*

i) $d = \frac{1}{2}$ *for* $p \leq 1$

ii) $0 < d < \frac{1}{2}$ *for* $p = 2$

iii) $d = 0$ *for* $p \geq 3$.

Proof: The proof follows from the discussion above. □

When $p = 3$, the μ-line relaxation will produce errors of which the dominant terms are piecewise linear. But, in a multigrid algorithm, the mesh size of the coarse grid is twice as large as the fine grid. Since γ is the same on all grids, although $p = 3$ on the finest grid, as the grids go from finest to coarsest level, the relative size of $1-\gamma$ to $\frac{1}{\sigma_t h}$ will change. To show this, assume that the finest grid is uniform and the number of cells is a power of 2. Now, let $\sigma_t h = 100$ and $\gamma = 1 - \frac{1}{(\sigma_t h)^3} = .999999$, that is, let $p_h = 3$. On the finest level interpolation should be linear. However, four levels down we have $\sigma_t 16h = 1600$ and $\gamma = 0.999999 = 1 - \frac{1}{(\sigma_t 16h)^{1.873}}$, that is $p_{16h} = 1.873$. From the discussion above we see that linear interpolation is not adequate on this level, even though linear interpolation is sufficient on the finest grid.

We use the parameter d to indicate the severity of deviation of errors from linearity after relaxation. If $\sigma_t h \gg 1$ on the finest level and the relation between γ and $\sigma_t h$ yields $p \leq 1$, then we can set $d = 1/2$. If $p \geq 3$ we can set $d = 0$. If $p = 2$ we can use $d = \frac{1}{2(1+4c_1)}$. Unfortunately, the relation between γ and $\sigma_t h$ does not always yield an integer. However, we can use the relaxation to determine the proper interpolation. Consider equation (3.11). Set $\underline{e}^{+}_{(2i-2)r} = \underline{1}$ and $\underline{e}^{+}_{(2i+1)l} = \underline{0}$ and use the two-cell inversion to compute the result. Then we may define

$$d_l^{\pm} = \frac{1}{2}\frac{\underline{1}^T(\underline{e}^{\pm}_{(2i-1)l} + \underline{e}^{\pm}_{(2i)r} - (\underline{e}^{\pm}_{2i-1)r} + \underline{e}^{\pm}_{(2i)l}))}{\underline{1}^T(\underline{e}^{\pm}_{(2i-1)l} + \underline{e}^{\pm}_{(2i)r})} \tag{3.29}$$

Parameters $d_r^{\pm}$ could be found in a similar manner by setting $\underline{e}^{+}_{(2i-2)r} = \underline{0}$ and $\underline{e}^{+}_{(2i+1)l} = \underline{1}$ in (3.11). Of course, for the uniform grid case $d_l^{\pm} = d_r^{\pm}$. Moreover, $d_r^{+} = d_r^{-} + O\left(\frac{1}{\sigma_t h}\right)$. In practice, we use an average value, $d = (d_r^{+} + d_r^{-})/2$.

Now, suppose we know $\underline{e}^{+}_{(2i-1)l}$,$\underline{e}^{+}_{(2i)r}$ and d; then, we can approximate $\underline{e}^{+}_{(2i-1)r}$ and $\underline{e}^{+}_{(2i)l}$ by interpolation:

$$\begin{bmatrix} \underline{e}^{+}_{(2i-1)l} \\ \underline{e}^{+}_{(2i-1)r} \\ \underline{e}^{+}_{(2i)l} \\ \underline{e}^{+}_{(2i)r} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{1}{2}-d & \frac{1}{2}-d \\ \frac{1}{2}-d & \frac{1}{2}-d \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \underline{e}^{+}_{(2i-1)l} \\ \underline{e}^{+}_{(2i)r} \end{bmatrix}. \tag{3.30}$$

The same interpolation can be used for negative angles. If $\underline{e}^{-}_{(2i-1)l}$,$\underline{e}^{-}_{(2i)r}$ and d are known, we have

$$\begin{bmatrix} \underline{e}^{-}_{(2i-1)l} \\ \underline{e}^{-}_{(2i-1)r} \\ \underline{e}^{-}_{(2i)l} \\ \underline{e}^{-}_{(2i)r} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{1}{2}-d & \frac{1}{2}-d \\ \frac{1}{2}-d & \frac{1}{2}-d \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \underline{e}^{-}_{(2i-1)l} \\ \underline{e}^{-}_{(2i)r} \end{bmatrix}. \tag{3.31}$$

For every two-cell pair, the problem is reduced to finding d, $\underline{e}^{+}_{(2i-1)l}$, $\underline{e}^{+}_{(2i)r}$, $\underline{e}^{-}_{(2i-1)l}$ and $\underline{e}^{-}_{(2i)r}$. If interpolation is relatively accurate, then we can use a coarser grid to approximate $\underline{e}^{+}_{(2i-1)l}$, $\underline{e}^{+}_{(2i)r}$, $\underline{e}^{-}_{(2i-1)l}$ and $\underline{e}^{-}_{(2i)r}$ on the fine grid. In a multigrid algorithm, a coarse grid solution can be approximated by an even coarser grid provided a good interpolation can be found. This process can be accomplished recursively down to the coarsest grid. On the coarsest grid, there will be only a few grid points and the solution on the coarsest grid usually can be solved explicitly. For our case, a two-cell μ-line relaxation will exactly solve the coarsest grid, since it has only two cells. Thus, finding a good interpolation is vital to an efficient multigrid scheme.

We define the restriction operator I_h^{2h} by

$$I_h^{2h} = \begin{bmatrix} S_1 & S_2 & & & \\ & & \ddots & & \\ & & & S_1 & S_2 \end{bmatrix} \otimes I, \tag{3.32}$$

where I is the $N \times N$ identity and

$$S_1 = \begin{bmatrix} \frac{1}{2} & 0 & \frac{1}{4} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{4} \\ 0 & 0 & \frac{1}{4} & 0 \\ 0 & 0 & 0 & \frac{1}{4} \end{bmatrix}, \quad S_2 = \begin{bmatrix} \frac{1}{4} & 0 & 0 & 0 \\ 0 & \frac{1}{4} & 0 & 0 \\ \frac{1}{4} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{4} & 0 & \frac{1}{2} \end{bmatrix}. \tag{3.33}$$

Note that the restriction operator represents full weighting. We also define the interpolation I_{2h}^h as

$$I_{2h}^h = \begin{bmatrix} T_1 & & \\ T_2 & & \\ & \ddots & \\ & & T_1 \\ & & T_2 \end{bmatrix} \otimes I, \tag{3.34}$$

where I is the $N \times N$ identity and

(3.35)

$$T_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \frac{1}{2}-d & 0 & \frac{1}{2}-d & 0 \\ 0 & \frac{1}{2}-d & 0 & \frac{1}{2}-d \end{bmatrix}, \quad T_2 = \begin{bmatrix} \frac{1}{2}-d & 0 & \frac{1}{2}-d & 0 \\ 0 & \frac{1}{2}-d & 0 & \frac{1}{2}-d \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

When $d = 0$, the interpolation operator I_{2h}^h is the transpose of the restriction operator scaled by a factor of $\frac{1}{2}$.

Generalization to a nonuniform grid is straightforward. Suppose a two-cell pair has cell widths $h_{(2i-1)}$ and $h_{(2i)}$. A two-cell inversion in (3.11) with $\underline{e}^+_{(2i-2)r} = \underline{1}$ and $\underline{e}^+_{(2i+1)l} = \underline{0}$ yields the deviation from linearity, which is then built into the interpolation formulas. One could carry a different interpolation for r and l as well as $+$ and $-$ if not in the asymptotic regime. With a uniform grid, the computation need only be done once per grid level, while for the nonuniform grid it must be done for each two-cell pair. However, all two-cell pairs can be calculated in parallel. The restriction operators correspond to full weighting, that is, the transpose of linear interpolation is weighted so that each row sums to 1.

We remark that the two-cell μ-line relaxation can be adapted to the context of mildly anisotropic scattering. For example, if the scattering operator involves P_3 scattering, then it has only 4 nonzero eigenvalues. This will result in a two-cell problem involving an easily solved linear system plus a rank 16 matrix. The two-cell inversion will require the solution of two systems involving a 16×16 matrix. However, the matrix will be the same throughout a particular material and could be factored before the start of the computation. The total computation would be $O(N)$.

In the next section, we will discuss the multigrid algorithm using the kinked interpolation operator I_{2h}^h and restriction operator I_h^{2h}.

§4 Multigrid Algorithm

In this section, we derive the coarse grid operator and show that, although the coarse grid operator no longer represent MLD on the coarse grid, it has the same structure as the fine grid operator and two-cell μ-line relaxation can be accomplished with the same formula and cost.

From (2.4), we define the fine grid operator L^h as

$$L^h = \begin{bmatrix} A^h & -C^h & & \\ -D^h & A^h & \ddots & \\ & \ddots & \ddots & -C^h \\ & & -D^h & A^h \end{bmatrix}, \tag{4.1}$$

where A^h, C^h and B^h are defined by (2.5) for uniform grid h. With restriction operator I_h^{2h} and interpolation operator I_{2h}^h defined by (3.32) and (3.35), the coarse grid operator L^{2h} can be obtained by

$$L^{2h} = I_{2h}^h L^h I_h^{2h} = \begin{bmatrix} A^{2h} & -C^{2h} & & \\ -D^{2h} & A^{2h} & \ddots & \\ & \ddots & \ddots & -C^{2h} \\ & & -D^{2h} & A^{2h} \end{bmatrix}, \tag{4.2}$$

where

$$A^{2h} = \begin{bmatrix} S_1 & S_2 \end{bmatrix} \begin{bmatrix} A^h & -C^h \\ -D^h & A^h \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \end{bmatrix}, \tag{4.3a}$$

$$C^{2h} = \begin{bmatrix} S_1 & S_2 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ C^h & 0 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \end{bmatrix}, \tag{4.3b}$$

$$D^{2h} = \begin{bmatrix} S_1 & S_2 \end{bmatrix} \begin{bmatrix} 0 & D^h \\ 0 & 0 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \end{bmatrix}. \tag{4.3c}$$

It is easy to verify that (see (2.1f), (2.5b) and (3.4))

$$C^{2h} = \frac{1}{\sigma_t h} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ M & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{4.4a}$$

$$(4.4b) \qquad D^{2h} = \frac{1}{\sigma_t h} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & M \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

To derive A^{2h}, we write A^h as

$$(4.5) \qquad A^h = \begin{bmatrix} F^h & 0 & G^h & 0 \\ 0 & K^h & 0 & Z^h \\ Z^h & 0 & K^h & 0 \\ 0 & G^h & 0 & F^h \end{bmatrix} - \begin{bmatrix} a^hR & a^hR & b^hR & b^hR \\ a^hR & a^hR & b^hR & b^hR \\ b^hR & b^hR & a^hR & a^hR \\ b^hR & b^hR & a^hR & a^hR \end{bmatrix}$$

where F^h, G^h, K^h and Z^h are diagonal matrices and a^h and b^h are scalars. We use this notation to explain that the coarse grid operator will follow the same pattern as the fine grid operator; that is, A^h and consecutive two-cell coarse grid operators A^{2h}, A^{4h}... are all composed of a matrix whose components are diagonal matrices plus a rank two matrix. For the fine grid h, $F^h = I + B, G^h = -B, K^h = I + B, Z^h = B, a^h = 1$ and $b^h = 0$. After multiplication, (4.3a) can be written as

$$(4.6) \qquad A^{2h} = \begin{bmatrix} F^{2h} & 0 & G^{2h} & 0 \\ 0 & K^{2h} & 0 & Z^{2h} \\ Z^{2h} & 0 & K^{2h} & 0 \\ 0 & G^{2h} & 0 & F^{2h} \end{bmatrix} - \begin{bmatrix} a^{2h}R & a^{2h}R & b^{2h}R & b^{2h}R \\ a^{2h}R & a^{2h}R & b^{2h}R & b^{2h}R \\ b^{2h}R & b^{2h}R & a^{2h}R & a^{2h}R \\ b^{2h}R & b^{2h}R & a^{2h}R & a^{2h}R \end{bmatrix},$$

where

$$(4.7a) \qquad F^{2h} = \frac{1}{4}[(2.5-d)F^h + (1-2d)G^h + (0.5-d)K^h + Z^h - (1-2d)B],$$

$$(4.7b) \qquad G^{2h} = \frac{1}{4}[(0.5-d)F^h + (2-2d)G^h + (0.5-d)K^h - (1-2d)B],$$

$$(4.7c) \qquad K^{2h} = \frac{1}{4}[(0.5-d)F^h + G^h + (2.5-d)K^h + (1-2d)Z^h - (1-2d)B],$$

$$(4.7d) \qquad Z^{2h} = \frac{1}{4}[(0.5-d)F^h + (0.5-d)K^h + (2-2d)Z^h - (1-2d)B],$$

$$(4.7e) \qquad a^{2h} = 0.75a^h - \frac{d}{2}a^h + 0.5b^h - \frac{d}{2}b^h,$$

$$(4.7f) \qquad b^{2h} = 0.25a^h - \frac{d}{2}a^h + 0.5b^h - \frac{d}{2}b^h.$$

Since F^h, G^h, K^h, Z^h and B are diagonal matrices, then $F^{2h}, G^{2h}, K^{2h}, Z^{2h}$ are also diagonal matrices. Structuraly, A^h and A^{2h} are the same. For two-cell μ-line relaxation on grid $2h$, we can still easily invert $\begin{bmatrix} A_{2i-1}^{2h} & -C_{2i-1}^{2h} \\ -D_{2i}^{2h} & A_{2i}^{2h} \end{bmatrix}$ with O(n) operations by taking advantage of the fact that the matrix to be inverted consists of an easily invertible matrix with diagonal components and a rank four matrix. The coarse grid operator on grid $4h$ also will have the same structure as on grids h and $2h$.

The generalization to nonuniform grids is again straight forward. Each two-cell pair will yield a different A^{2h} but it will again have the form as in (4.6) and two-cell inversion on each coarse grid will be amenable to fast inversion.

In the next section, we will display properties of the kinked interpolation operator.

§5 Properties of Kinked Interpolation

In this section, we will display the properties of the kinked interpolation operator by solving (2.5) with boundary conditions

$$\underline{\psi}_{0r}^{+} = \underline{1}, \quad \underline{\psi}_{(m+1)l}^{-} = \underline{0} \tag{5.1}$$

on a slab of $x\epsilon(-1,1)$. Suppose h is the mesh size of the finest grid and H is the mesh size of the coarsest grid, which has only two cells. We choose $\sigma_t \gg 1$ and

$$\gamma = 1 - \frac{1}{\sigma_t^q} \tag{5.2}$$

for $q = 1, 2, 3$. The case $q = 1$ represents substantial absorption. The case $q = 2$ is the case of most interest to the transport community and yields the well known thick diffusion limit. The case $q = 3$ also yields the thick diffusive limit, but with little absorption.

Given σ_t and γ and grid size h, the value p that satisfies

$$\gamma = 1 - \frac{1}{(\sigma_t h)^p} \tag{5.3}$$

describes the behavior of the error after relaxation as in Theorem 5. This analysis assumes $\sigma_t h \gg 1$. Equating (5.2) and (5.3) yields

$$p = q\frac{\log(\sigma_t)}{\log(\sigma_t h)}. \tag{5.4}$$

In our analysis, $h < 1$, which implies $p \geq q$ on all grids with $p = q$ on the coarsest grid where $H = 1$. This shows that, if $q = 3$, the error after relaxation will be linear on all grids. It is only for $q < 3$ that kinked elements are necessary. To see this, let $\sigma_t = 100$. Table 5.1 shows the values of p for $q = 1$ and $q = 2$ on grids with m varying from 128 to 2.

m	128	64	32	16	8	4	2
$q = 1$	10.32	4.04	2.51	1.82	1.43	1.17	1
$q = 2$	20.64	8.08	5.02	3.64	2.86	2.34	2

Table 5.1 Value of p on Various Grid Levels

For each set of σ_t, γ, we can find parameter d defined in the previous section on each level(from finest grid to coarsest grid). With d on every grid obtained, we have kinked interpolation on each grid. We define the accumulated interpolation as

$$I_H^h = I_{2h}^h I_{4h}^{2h} \cdots\cdots\cdots I_H^{H/2}. \tag{5.5}$$

With accumulated interpolation so defined, we then solve a homogeneous S_8 transport equation with unit isotropic boundary condition on the left and zero on the right on grids with 8 cells and 128 cells respectively. We choose $\sigma_t = 100$ and $\gamma = 1 - \frac{1}{\sigma_t^p}, p = 1, 2, 3$. We choose an angle line $\mu = 0.525$ for the comparison of the interpolated approximation and the exact discretized solution. With the calculated values $\underline{\psi}_{1l}^+$ and $\underline{\psi}_{mr}^+$, we use the accumulated kinked interpolation operator I_H^h from the coarsest grid to the finest grid to obtain all interior values on finest grid and compare the interpolated approximation with the calculated solution. Figure 5.1 depicts the case in which there is substantial absorption (p = 1). In this case, the relaxation itself is very efficient.

Figure 5.1a

Figure 5.1b

Figure 5.2a

Figure 5.2b

Figure 5.2 shows the case that is of most interest in the transport community (p =

2). Figure 5.3 shows the case in which there is not much absorption (p = 3). Notice how well interpolation approximates the exact MLD on the finest grid. From Figure 5.3, we can see that when the power p increases, the interpolation is more accurate since the kinked elements are less kinked. We can see from these plots that the relaxation-induced interpolation we have designed is accurate and suitable for the multigrid algorithm.

Figure 5.3a

Figure 5.3b

In the next section, we will present computational results of the multigrid algorithm using kinked interpolation.

§6 **Computational Results**

6.1 Multigrid Convergence Factors

The following computational results were conducted on the domain $x \in [0, 1]$ using homogeneous boundary conditions. The domain was divided into $m = 128$ spatial cells and an S_{32} discretization ($n = 16$) was used in angle. The problem is then specified completely by choosing $\sigma_t h$ and $\gamma = \frac{\sigma_s}{\sigma_t}$. Convergence factors were computed by setting the right-hand-side equal to zero, choosing a random initial guess and performing 15 $V(1,1)$-cycles and taking the geometric mean of the last 5 cycles. This process exposes the most slowly converging eigen components. Initial reduction is usually much faster.

From computational results conducted on S_N with N ranging from 2 to 256, we observe the same convergence performance as those shown here.

σ_t h	$\gamma = 0.999999$	$\gamma = 0.999$	$\gamma = 0.9$	$\gamma = 0.7$
10^{-5}	0.37×10^{-9}	0.37×10^{-9}	0.35×10^{-9}	0.33×10^{-9}
10^{-4}	0.31×10^{-6}	0.31×10^{-6}	0.31×10^{-6}	0.30×10^{-6}
10^{-3}	0.11×10^{-3}	0.11×10^{-3}	0.11×10^{-3}	0.9×10^{-4}
10^{-2}	0.50×10^{-2}	0.50×10^{-2}	0.44×10^{-2}	0.38×10^{-2}
10^{-1}	0.68×10^{-2}	0.43×10^{-2}	0.68×10^{-2}	0.68×10^{-2}
10^{-0}	0.70×10^{-2}	0.72×10^{-2}	0.14×10^{-2}	0.19×10^{-2}

Table 6.1 Convergence Factors for Uniform Grid

From Table 6.1, we observe that the convergence factor ρ is $O(\sigma_t^3 h^3)$ when $\sigma_t h \ll 1$.

For the thick limit, we will let $\gamma = 1 - (\frac{1}{\sigma_t h})^p$, since, from our previous section, the power p will influence the error distribution after relaxation. We will present results with $p = 1, 2, 3$ and ∞.

In Table 6.2, the convergence rates ρ are for a multigrid V(1,1) cycle with linear interpolation. Table 6.3 contains the convergence rates for a multigrid V(1,1) cycle with kinked interpolation.

σ_t h	p = 1	p = 2	p = 3	p = ∞
10^1	0.37 $\times 10^{-2}$	0.15	0.32	0.27 $\times 10^{-4}$
10^2	0.78 $\times 10^{-2}$	0.61	0.74	0.22 $\times 10^{-6}$
10^3	0.84 $\times 10^{-2}$	0.82	0.62	0.20 $\times 10^{-8}$
10^4	0.85 $\times 10^{-2}$	0.85	0.34	0.18 $\times 10^{-10}$

Table 6.2. Convergence Factors for Linear Interpolation

σ_t h	p = 1	p = 2	p = 3	p = ∞
10^1	0.22 $\times 10^{-5}$	0.43 $\times 10^{-2}$	0.53 $\times 10^{-2}$	0.85 $\times 10^{-4}$
10^2	0.45 $\times 10^{-6}$	0.25 $\times 10^{-2}$	0.51 $\times 10^{-3}$	0.11 $\times 10^{-5}$
10^3	0.47 $\times 10^{-7}$	0.36 $\times 10^{-3}$	0.89 $\times 10^{-5}$	0.13 $\times 10^{-7}$
10^4	0.49 $\times 10^{-8}$	0.39 $\times 10^{-4}$	0.11 $\times 10^{-6}$	0.14 $\times 10^{-9}$

Table 6.3 Convergence Factors for Kinked Interpolation

Table 6.2 shows that the multigrid $V(1,1)$ cycle with linear interpolation has a convergence factor ρ on the order of $O((\frac{1}{\sigma_t h})^2)$ with $\gamma = 1 (p = \infty)$ in the thick limit. But, when there is absorption($\gamma \neq 1$), the convergence factor for linear interpolation is not satisfactory at all. As we have discussed in previous sections, when $\gamma \neq 1$ the errors after relaxation will be kinked. Linear interpolation will not be suitable for this case since the linear interpolation will simply interpolate the coarse grid approximation to the fine grid as if the error on fine grid after relaxation were linear. So, even if the coarse grid is solved exactly, linear interpolation of this solution to the fine grid will not approximate the error on the fine grid. The situation will be compounded for a multigrid $V(1,1)$ cycle since there are many levels. Every coarse grid solution will not approximate the error on the next finer grid. From our discussion in the previous section, when $p = 3$ on the fine grid, there is relatively little absorption and the errors after relaxation will be nearly linear. But, as we go down to coarser grids, we will reach a level on which $\sigma_t h$ will satisfy $\gamma = 1 - (\frac{1}{\sigma_t h})^2$. Thus, using linear interpolation on this particular level will adversely affect the overall convergence factor. From Table 6.3 we see that when kinked interpolation is used the multigrid V(1,1) has a convergence factor ρ on the order of $O(\frac{1}{\sigma_t h})$ for all levels of absorption. When there is no absorption, the multigrid V(1,1) with kinked interpolation has a convergence factor on the order of $O((\frac{1}{\sigma_t h})^2)$.

σ_t	240	245	250	256	260	265	270	275	280
ρ	0.0082	0.0083	0.0084	0.0085	0.0085	0.0085	0.0084	0.0084	0.0083

Table 6.4. Convergence Factors for Kinked Interpolation,$\gamma = 0.999$

In Table 6.4, we show a range within which the worst convergence factor occurrs when $\gamma = 0.999$ and $m = 256$. In this case, the maximal ρ is 0.0085.

In Table 6.5, we present convergence factors of the multigrid V(1,1) cycle with kinked interpolation for a nonuniform grid. We select $\sigma_t h_i = c \times 10^{2*\eta_i}$, where η_i is a random number between (-1,1). For example, when $c = 1$, $\sigma_t h$ ranges from 0.01 to 100. We use parameter c to shift from the thin limit to the thick limit.

c	$\gamma = 0.999999$	$\gamma = 0.999$
10^{-4}	0.12×10^{-10}	0.12×10^{-10}
10^{-3}	0.32×10^{-7}	0.32×10^{-7}
10^{-2}	0.29×10^{-4}	0.29×10^{-4}
10^{-1}	0.20×10^{-4}	0.10×10^{-2}
10^{-0}	0.22×10^{-3}	0.32×10^{-2}
10^{1}	0.17×10^{-1}	0.54×10^{-2}
10^{2}	0.70×10^{-1}	0.40×10^{-3}
10^{3}	0.78×10^{-1}	0.19×10^{-3}
10^{4}	0.45×10^{-1}	0.17×10^{-4}
10^{5}	0.19×10^{-2}	0.20×10^{-5}

Table 6.5 Convergence Factors for Nonuniform Grid

From Table 6.5, we see that the kinked interpolation is also suitable for nonuniform grid, or equivalently, for nonconstant σ_t.

6.2 DSA Convergence Factors

In this section, we will present a comparison of the DSA algorithm ([2, 5]) and the Multigrid algorithm with kinked interpolation. The slab is assumed to have physical thickness 2. Thus, σ_t represents the width of the slab measured in the number of mean-free-paths. The tests were performed using S_8 and a wide range of σ_t and m(the number of cells) and various values of γ ranging from .9999 to .9. The performance was similar for each value of γ, so we present only the complete results for $\gamma = .999$. The results for other values of γ can be found in ([11]). Tables 6.6 and 6.7 give convergence factors for multigrid and DSA for various values of σ_t and m. The diagonals of these tables represent constant $\sigma_t h$.

In the following tables, we say the convergence factor is 0 if $\rho < 10^{-11}$. For both algorithms, the convergence factors are roughly constant along diagonals. But for each fixed γ, the multigrid convergence factor will approach zero much faster than the DSA convergence factor as σ_t goes to ∞ in the thick limit and σ_t goes to zero in the thin limit.

σ_t	m = 16	m = 64	m = 256	m = 1024
4^{-5}	0.0	0.0	0.0	0.0
4^{-4}	0.32×10^{-10}	0.13×10^{-10}	0.10×10^{-10}	0.10×10^{-10}
4^{-3}	0.71×10^{-8}	0.30×10^{-8}	0.19×10^{-8}	0.20×10^{-8}
4^{-2}	0.12×10^{-5}	0.51×10^{-6}	0.30×10^{-6}	0.30×10^{-6}
4^{-1}	0.61×10^{-4}	0.29×10^{-4}	0.13×10^{-4}	0.11×10^{-4}
1	0.57×10^{-3}	0.37×10^{3}	0.32×10^{-3}	0.31×10^{-3}
4	0.12×10^{-2}	0.12×10^{-2}	0.11×10^{-2}	0.13×10^{-1}
4^2	0.31×10^{-2}	0.36×10^{-2}	0.54×10^{-2}	0.76×10^{-2}
4^3	0.57×10^{-2}	0.79×10^{-2}	0.73×10^{-2}	0.11×10^{-1}
4^4	0.57×10^{-2}	0.69×10^{-2}	0.85×10^{-2}	0.78×10^{-2}
4^5	0.68×10^{-3}	0.58×10^{-3}	0.71×10^{-2}	0.86×10^{-2}
4^6	0.93×10^{-5}	0.71×10^{-3}	0.58×10^{-2}	0.71×10^{-2}
4^7	0.25×10^{-7}	0.95×10^{-5}	0.72×10^{-3}	0.58×10^{-2}
4^8	0.33×10^{-10}	0.25×10^{-7}	0.94×10^{-5}	0.72×10^{-3}
4^9	0.0	0.32×10^{-10}	0.25×10^{-7}	0.94×10^{-5}
4^{10}	0.0	0.0	0.32×10^{-10}	0.25×10^{-7}

Table 6.6 Convergence Factors for Multigrid $\gamma = 0.999$

σ_t	m = 16	m = 64	m = 256	m = 1024
4^{-5}	0.19×10^{-2}	0.19×10^{-2}	0.190×10^{-2}	0.190×10^{-2}
4^{-4}	0.75×10^{-2}	0.75×10^{-2}	0.75×10^{-2}	0.75×10^{-2}
4^{-3}	0.28×10^{-1}	0.28×10^{-1}	0.28×10^{-1}	0.28×10^{-1}
4^{-2}	0.83×10^{-1}	0.83×10^{-1}	0.83×10^{-1}	0.84×10^{-1}
4^{-1}	0.153	0.150	0.150	0.151
1	0.186	0.203	0.204	0.204
4	0.179	0.216	0.216	0.216
4^2	0.142	0.165	0.204	0.210
4^3	0.127	0.142	0.149	0.163
4^4	0.118	0.129	0.143	0.156
4^5	0.102	0.119	0.130	0.144
4^6	0.66×10^{-1}	0.102	0.119	0.130
4^7	0.27×10^{-1}	0.66×10^{-1}	0.102	0.119
4^8	0.82×10^{-2}	0.27×10^{-1}	0.66×10^{-1}	0.102
4^9	0.21×10^{-2}	0.82×10^{-2}	0.27×10^{-1}	0.66×10^{-1}
4^{10}	0.54×10^{-3}	0.21×10^{-2}	0.82×10^{-2}	0.27×10^{-1}

Table 6.7 Convergence Factors for DSA $\gamma = 0.999$

In all cases, the multigrid convergence factors were superior to the DSA convergence factors. However, it is important to adjust for the relative amount of computational work required by each algorithm. Of course, such measures will be machine dependent. On the Cray Y/MP, where these tests were performed, we compared times for N=64 and m=1024. Ten V(1,1) cycles required 93 seconds. This includes 4.2 seconds to obtain the parameter d on every level. The parameter d on the coarse grid will depend on the relaxation on the fine grid. Thus, this setup process is sequential. One V(1,1) cycle required 9.3 seconds and one DSA cycle required 3.7 seconds. The ratio of these times is 2.5. Table 6.8 contains the results of raising each entry in Tables 6.7 to the power 2.5. This is a more fair comparison with the multigrid tables on a serial machine.

From Tables 6.6 - 6.8 we see that the convergence factors are nearly constant along diagonals, that is, for constant $\sigma_t h$. This was true for other values of γ also. In Table 6.9 we present both multigrid and adjusted DSA convergence factors for $m = 1024$, four values of γ and various values of $\sigma_t h$. Again, we see that multigrid is faster in all tests. The two algorithms have nearly equal rates in regimes in which $\sigma_t h \sim 1$. This is the region in which the relaxation will not produce continuous kinked linear errors across two cells and, thus, the kinked interpolation is not as accurate.

On a parallel machine we expect the results to more heavily favor the multigrid algorithm. Both algorithms can be implemented with parallel complexity $O(log(m))$. In this context, however, we expect the times to be more nearly equal. The multigrid algorithm has been implemented on an Thinking Machines Inc. CM-200 [8, 9]. The fundamental step in the DSA algorithm, the transport sweep, has also been implemented on the CM-200. A form of cyclic reduction was used. A comparison of timings on this machine appear in [15].

In either setting, parallel or serial, a full multigrid algorithm (FMG) can be implemented ([12]). This would provide a savings in some regions of the tables. Moreover, full multigrid provides a natural framework for adaptive grid refinement.

σ_t	m = 16	m = 64	m = 256	m = 1024
4^{-5}	0.16×10^{-6}	0.16×10^{-6}	0.16×10^{-6}	0.16×10^{-6}
4^{-4}	0.49×10^{-5}	0.49×10^{-5}	0.49×10^{-5}	0.49×10^{-5}
4^{-3}	0.13×10^{-3}	0.13×10^{-3}	0.13×10^{-3}	0.13×10^{-3}
4^{-2}	0.20×10^{-2}	0.20×10^{-2}	0.20×10^{-2}	0.20×10^{-2}
4^{-1}	0.92×10^{-2}	0.87×10^{-2}	0.87×10^{-2}	0.87×10^{-2}
1	0.15×10^{-1}	0.19×10^{-1}	0.19×10^{-1}	0.19×10^{-1}
4	0.14×10^{-1}	0.22×10^{-1}	0.22×10^{-1}	0.22×10^{-1}
4^2	0.76×10^{-2}	0.11×10^{-1}	0.19×10^{-1}	0.22×10^{-1}
4^3	0.57×10^{-2}	0.76×10^{-2}	0.86×10^{-2}	0.11×10^{-1}
4^4	0.48×10^{-2}	0.59×10^{-2}	0.77×10^{-2}	0.96×10^{-2}
4^5	0.33×10^{-2}	0.49×10^{-2}	0.61×10^{-2}	0.79×10^{-2}
4^6	0.12×10^{-2}	0.33×10^{-2}	0.49×10^{-2}	0.61×10^{-2}
4^7	0.12×10^{-3}	0.11×10^{-2}	0.33×10^{-2}	0.49×10^{-2}
4^8	0.61×10^{-5}	0.12×10^{-3}	0.11×10^{-2}	0.33×10^{-2}
4^9	0.20×10^{-6}	0.61×10^{-5}	0.12×10^{-3}	0.11×10^{-2}
4^{10}	0.68×10^{-8}	0.20×10^{-6}	0.61×10^{-5}	0.12×10^{-3}

Table 6.8 Adjusted Convergence Factors for DSA $\gamma = 0.999$

$\sigma_t h$	$\gamma = .9999$	$\gamma = .999$	$\gamma = .99$	$\gamma = .9$
2^{-9}	3.2e-04 / 1.9e-02	3.1e-04 / 1.9e-02	3.5e-04 / 1.9e-02	3.2e-04 / 1.5e-02
2^{-7}	1.1e-03 / 2.2e-02	1.3e-02 / 2.2e-02	3.2e-03 / 2.2e-02	1.5e-03 / 1.7e-02
2^{-5}	1.3e-03 / 2.1e-02	7.6e-03 / 2.2e-02	9.8e-03 / 2.1e-02	2.0e-03 / 1.5e-02
2^{-3}	7.1e-03 / 1.2e-02	1.1e-02 / 1.1e-02	1.0e-02 / 4.1e-02	2.3e-03 / 1.2e-02
2^{-1}	3.9e-03 / 8.3e-02	7.8e-03 / 9.6e-03	1.0e-02 / 8.7e-02	7.6e-04 / 5.1e-03
2	2.6e-03 / 7.7e-03	8.6e-03 / 7.9e-03	4.5e-03 / 6.9e-03	3.5e-04 / 2.9e-03
2^{3}	4.1e-03 / 6.2e-03	7.1e-03 / 6.1e-03	8.4e-04 / 4.4e-03	9.0e-05 / 5.6e-04
2^{5}	4.0e-03 / 5.5e-03	5.8e-03 / 4.9e-03	4.5e-04 / 1.8e-03	2.2e-07 / 3.8e-05
2^{7}	2.3e-03 / 5.2e-03	7.2e-04 / 3.3e-03	2.0e-06 / 2.8e-04	2.7e-10 / 1.6e-06
2^{9}	4.8e-04 / 4.4e-03	9.4e-06 / 1.1e-03	3.1e-09 / 1.6e-05	0 / 5.0e-08
2^{11}	2.1e-05 / 2.6e-03	2.5e-08 / 1.2e-04	3.0e-12 / 6.7e-08	0 / 1.7e-09
2^{13}	1.5e-07 / 5.9e-04	3.2e-11 / 6.1e-06	0 / 2.2e-08	0 / 5.5e-11
2^{15}	2.7e-10 / 4.3e-05	0 / 2.0e-07	0 / 7.2e-10	0 / 0
2^{17}	0 / 1.9e-06	0 / 6.8e-09	0 / 2.1e-11	0 / 0

Table 6.9 Convergence Factors for Multigrid/Adjusted DSA

§7 Conclusions

From our analysis and computational results, we conclude that:

- The two-cell μ-line red/black relaxation is inexpensive, highly vectorizable and parallelizable. Modification of this algorithm to incorporate mildly anisotropic scattering is also viable.
- When there is no absorption ($\gamma = 1$), this relaxation will cause the error to be linear, independent of angle and continuous across two cells up to $O(\frac{1}{\sigma_t h})$ accuracy when $\sigma_t h \gg 1$.
- When $\gamma < 1$, the error after relaxation will be essentially kinked linear. The severity of deviation from linearity of the error after relaxation is determined by the values γ, $\sigma_t h$ and p in $\gamma = 1 - (\frac{1}{\sigma_t h})^p$. In the thick limit, the error after relaxation will be independent of angle up to accuracy of $O(\frac{1}{\sigma_t h})$.
- Relaxation induced kinked interpolation is fairly accurate, and it can be obtained by the relaxation operator on each level to ensure that each level will have a good coarse grid approximation. This methodology can be extended to higher dimensions.
- The error reduction factor ρ for a multigrid V(1,1)-cycle is $\rho = O((\sigma_t h)^3)$ for the thin limit and $\rho = O(\frac{1}{\sigma_t h})$ for the thick limit when $\gamma = 1 - (\frac{1}{\sigma_t h})^p$ and $1 < p < 3$. When p is smaller than 1, there will be more absorption and the multigrid performance will be better. When p is more than 3, there will be hardly any absorption and the multigrid performance will be close to that of the performance when $\gamma = 1$ ([10]).
- The multigrid scheme was shown computationally to be effective for highly irregular meshes and heterogeneous material.
- The multigrid algorithm is more efficient than the DSA algorithm in all regimes.

REFERENCES

[1] M. L. Adams and W. R. Martin, *Diffusion-Synthetic Acceleration of Discontinuous Finit-Element Transport Iterations,* to appear in Nucl. Sci. Eng.

[2] R. E. Alcouffe, *Diffusion Synthetic Acceleration Methods for the Diamond-Differenced Discrete-Ordinates Equations*, Nucl. Sci. Eng., 64, *p. 344, 1977.*

[3] A. Brandt, *Multi-level adaptive solution to boundary-value problems*, Math. Comp., 31 (1977),333-390.

[4] V. Faber and T. A. Manteuffel, *Neutron transport from the viewpoint of linear algebra*, Transport Theory, Invariant Imbedding and Integral Equations*(Nelson, Faber, Manteuffel, Seth and White, eds.)* Lecture Notes in Pure and Applied Mathematics, 115, p. 37-61, Marcel-Dekker, April, 1989.

[5] E. W. Larsen, *Unconditionally Stable Diffusion-Synthetic Acceleration Methods for the Slab Geometry Discrete Ordinates Equations* , Nucl. Sci. Eng., 82, *p. 47, 1982.*

[6] E.W.Larsen, J.E. Morel, *Asymptotic Solutions of Numerical Transport Problems in Optically Thick Diffusive Regimes II,* J. Comp. Phys. 83, *p. 212, 1989.*

[7] E. E. Lewis and W. F. Miller,Jr., *Computational Methods of Neutron Transport* , 1984.

[8] T. A. Manteuffel, S. McCormick, J. Morel, S. Oliveira and G. Yang, *Parallel Multigrid Methods for Transport Equations,* Proceeding of Copper Mountain Conference on Iterative Methods, April 9 - 14, 1992

[9] T. A. Manteuffel, S. McCormick, J. Morel, S. Oliveira and G. Yang, *A Parallel Version of a Multigrid Algorithm for Isotropic Transport Equations,* SIAM Journal of Sci. Comp., Vol 15, No. 2, (1992).

[10] T. A. Manteuffel, S. McCormick, J. Morel, S. Oliveira and G. Yang, *A Fast Multigrid Algorithm for Isotropic Transport Problems I: Pure Scattering*, to appear *SIAM Journal of Sci. Comp.*.

[11] T. A. Manteuffel, S. McCormick, J. Morel, and G. Yang, *A Fast Multigrid Algorithm for Isotropic Transport Problems II: with Absorption*, Los Alamos National Laboratories Report, LA-UR-94-2361, June (1994).

[12] S. F. McCormick, *Multilevel Adaptive Methods for Partial Differential Equations* , SIAM Press, (1989).

[13] J. E. Morel and E. W. Larsen, *A Multiple Balance Approach for Differenceing the S_N Equations* , Nucl. Sci. Eng., 105, *p. 1, 1990.*

[14] J. E. Morel and T. A. Manteuffel: *An Angular Multigrid Acceleration Technique for the S_N Equations With Highly Forward-Peaked Scattering*, Nucl. Sci. Eng., 107 330(1991).

[15] S. Oliveira, *Parallel Multigrid Methods for Transport Equations*, Ph. D. Thesis, Department of Mathematics, University of Colorado at Denver, May, 1993.