A Spectral Analysis of the Domain Decomposed Monte Carlo Method for Linear Systems

Stuart R. Slattery^{a,2}, Thomas M. Evans^{a,1}, Paul P.H. Wilson^{b,3}

^a*Oak Ridge National Laboratory, 1 Bethel Valley Rd., Oak Ridge, TN 37831, U.S.A.*

^b*University of Wisconsin - Madison, 1500 Engineering Dr., Madison, WI 53706, U.S.A.*

Abstract

The domain decomposed behavior of the adjoint Neumann-Ulam Monte Carlo method for solving linear systems is analyzed using the spectral properties of the linear operator. Relationships for the average length of the adjoint random walks, a measure of convergence speed and serial performance, are made with respect to the eigenvalues of the linear operator. In addition, relationships for the effective optical thickness of a domain in the decomposition are presented based on the spectral analysis and diffusion theory. Using the effective optical thickness, the Wigner rational approximation and the mean chord approximation are applied to estimate the leakage fraction of random walks from a domain in the decomposition as a measure of parallel performance and potential communication costs. The one-speed, two-dimensional neutron diffusion equation is used as a model problem in numerical experiments to test the models for symmetric operators with spectral qualities similar to light water reactor problems. In general, the derived approximations show good agreement with random walk lengths and leakage fractions computed by the numerical experiments.

Keywords: Monte Carlo, domain decomposition, MCSA, parallel computing, linear solvers

1. Introduction

An alternative approach to solving linear systems is to employ Monte Carlo methods that sample a distribution with an expectation value equivalent to that of the inverted linear operator. Such methods have been in existence for decades with the earliest reference noted here an enjoyable manuscript published in 1950 by Forsythe and Leibler [1]. In their outline, Forsythe and Leibler in fact credit the creation of this technique to J. Von Neumann and S.M. Ulam some years earlier than

^{*}Notice: This manuscript has been authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

Email addresses: slatterysr@ornl.gov (Stuart R. Slattery), evanstm@ornl.gov (Thomas M. Evans), wilsonp@engr.wisc.edu (Paul P.H. Wilson)

¹Computational Engineering and Energy Sciences Group, Computer Science and Mathematics Division

²Radiation Transport Group, Reactor and Nuclear Systems Division

³Engineering Physics Department, College of Engineering

its publication. In 1952 Wasow provided a more formal explanation of Von Neumann and Ulam’s method [2] and Hammersley and Handscomb’s 1964 monograph [3] and Spanier and Gelbard’s 1969 book [4] present additional detail on a method adjoint to that described by Neumann and Ulam. To improve their convergence, Halton leveraged the Monte Carlo algorithm in an iterative refinement scheme where a residual Monte Carlo solution serves as a correction to a fixed point iteration [5]. In recent years, the adjoint Neumann-Ulam method has been leveraged in Monte Carlo Synthetic Acceleration (MCSA) methods that, for certain classes of problems, are competitive with conventional methods and are applicable to a variety of linear systems [6, 7]. To apply these new MCSA methods to engineering problems, our research aims to improve the underlying Monte Carlo algorithms to handle large-scale problems in parallel computing environments.

To date, parallel Neumann-Ulam methods have been limited to full domain replication with parallelism exploited through individual random walks [8]. In reactor physics Monte Carlo applications, however, domain decomposition has been identified as a key principle in moving forward in high performance computing to enable higher fidelity simulations [9–11]. To accomplish this, we recognize from the literature that random walks must be transported from domain to domain as the simulation progresses and they transition to states that are not in the local domain. Because we have chosen a domain decomposition strategy in a parallel environment, this means that communication of these random walks must occur between compute nodes owning neighboring pieces of the global domain. We wish to characterize this communication not only because communication is in general expensive, but also because these nearest-neighbor communication sequences, specifically, have poor algorithmic strong scaling due to the simultaneous reduction of on-process work and increase in communication as processor counts grow and local problem sizes shrink [12].

The purpose of this study is to provide a simple, analytic theory based on the properties of the linear system that will allow for estimates of the domain decomposed behavior of the adjoint Neumann-Ulam method. When solving problems where the linear operator is symmetric, a host of analytic theories exist based on the eigenvalue spectrum of the operator that characterize their behavior in the context of deterministic linear solvers. Using past work, these theories are adapted to the domain decomposed adjoint Neumann-Ulam method using the one-speed, two-dimensional neutron diffusion equation. In this paper we describe the adjoint Neumann-Ulam Monte Carlo method followed by a presentation of the model problem. Using the linear system generated by the discretization of the model problem, we use a spectral analysis to generate analytic relations for the eigenvalues of the operator based on system parameters. Using the eigenvalue spectra, we then build relationships to characterize the transport of random walks in a decomposed domain and the fraction of random walks that leak from a domain and will therefore have to be communicated. Finally, we compare these analytic results to numerical experiments conducted with the model problem and draw conclusions looking towards future work.

2. Adjoint Neumann-Ulam Method

We seek solutions of the general linear problem in the following form:

$$\mathbf{A}\mathbf{x} = \mathbf{b} . \quad (1)$$

Choosing the adjoint Monte Carlo method to invert the linear operator, we define the linear system adjoint to Eq (1):

$$\mathbf{A}^T \mathbf{y} = \mathbf{d} , \quad (2)$$

where $\mathbf{y}$ and $\mathbf{d}$ are the yet to be determined adjoint solution and source respectively and $\mathbf{A}^T$ is the adjoint operator. We can derive a Monte Carlo estimator from the adjoint method that will also give the solution vector, $\mathbf{x}$. We first rearrange Eq (2) as:

$$\mathbf{y} = (\mathbf{I} - \mathbf{H}^T)^{-1} \mathbf{d} , \quad (3)$$

where $\mathbf{H} = \mathbf{I} - \mathbf{A}$ is the *iteration matrix*. This yields the adjoint Neumann series:

$$\mathbf{y} = \sum_{k=0}^{\infty} (\mathbf{H}^T)^k \mathbf{d} . \quad (4)$$

For this series to converge, we require that the spectral radius of $\mathbf{H}$ must remain less than unity⁴. We expand the matrix-vector multiply element-wise in this summation to yield a sum of sums:

$$y_i = \sum_{k=0}^{\infty} \sum_{i_1}^N \sum_{i_2}^N \dots \sum_{i_k}^N h_{i_k, i_{k-1}} \dots h_{i_2, i_1} h_{i_1, i} d_{i_k} . \quad (5)$$

To find the solution to Eq (1) we build an estimator for the adjoint solution from this series expansion. The adjoint estimator can be related to the solution by defining the following inner product equivalence [4]:

$$\langle \mathbf{A}^T \mathbf{x}, \mathbf{y} \rangle = \langle \mathbf{x}, \mathbf{A} \mathbf{y} \rangle . \quad (6)$$

From this definition it follows that:

$$\langle \mathbf{x}, \mathbf{d} \rangle = \langle \mathbf{y}, \mathbf{b} \rangle . \quad (7)$$

We have 2 unknowns, $\mathbf{y}$ and $\mathbf{d}$, and therefore we require two constraints to close the system. We use Eq (7) as the first constraint and as a second constraint we select:

$$\mathbf{d} = \boldsymbol{\delta}_j , \quad (8)$$

where $\boldsymbol{\delta}_j$ is one of a set of vectors in which the j^{th} component is the Kronecker delta function, $\delta_{i,j}$. If we apply Eq (8) to our first constraint in Eq (7) we get the following convenient outcome:

$$\langle \mathbf{y}, \mathbf{b} \rangle = \langle \mathbf{x}, \boldsymbol{\delta}_j \rangle = x_j , \quad (9)$$

⁴ $\mathbf{H}^T$ contains the same eigenvalues as $\mathbf{H}$ and therefore has the same spectral radius.

meaning that if we compute the inner product of the original source and the adjoint solution using a delta function source, we recover one component of the original solution.

In the Neumann-Ulam method, a Monte Carlo game is played to construct the terms of the Neumann series in Eq (5). We use random walks of length k to build the series of state transitions in the summation:

$$\nu = i_0 \rightarrow i_1 \rightarrow i_2 \rightarrow \cdots \rightarrow i_{k-1} , \quad (10)$$

where ν is a particular random walk permutation. Starting a random walk permutation requires the computation of an initial starting state, i_0 , and initial weight, W_0 . The initial state is determined by sampling the source vector $\mathbf{b}$ with probabilities:

$$P_{(i_0=i)}(\nu) = \frac{|b_i|}{\|\mathbf{b}\|_1} , \quad (11)$$

and with a corresponding starting weight of:

$$W_0 = \|\mathbf{b}\|_1 \frac{b_{i_0}}{|b_{i_0}|} , \quad (12)$$

which gives the additional useful relation:

$$b_{i_0} = W_0 P_{(i_0=i)} . \quad (13)$$

After a random walk is started, we construct the remaining terms in Eq (5) using a row-scaled transition probability matrix and corresponding transition weights:

$$p_{ij} = \frac{|h_{ji}|}{\sum_j |h_{ji}|} , \quad w_{ij} = \frac{|h_{ji}|}{p_{ij}} , \quad (14)$$

where each row of the probability matrix, $\mathbf{P}_i$, gives a probability distribution function for moving from row i to a new row, j . As a random walk is computed by sampling the probability matrix, every stochastic transition from a state i to a state j has a corresponding weight, w_{ij} , that is used to update the current weight of the random walk. At the m^{th} step of a random walk, its weight is:

$$W_m = W_0 w_{i_0, i_1} w_{i_1, i_2} \cdots w_{i_{m-1}, i_m} , \quad (15)$$

with W_0 given by Eq (12). To stop a random walk, we utilize a relative weight cutoff parameter:

$$W_f = W_c W_0 : \quad (16)$$

where W_c is the *weight cutoff*. The random walk will be terminated after m steps if $W_m < W_f$.

Using our result from Eq (9) the Monte Carlo estimator for the solution from a particular random walk permutation of length k is:

$$X_\nu = \sum_{m=0}^k W_m \delta_{i, i_m} , \quad (17)$$

where the Kronecker delta indicates that the tally contribution for each step of the random walk is only in its current state, i_m . Finally, the expectation value using all random walk permutations is:

$$E\{X\} = \sum_{\nu} P_{\nu} X_{\nu} \quad (18)$$

which if expanded with Eqs (14) and (13) directly recovers the exact solution:

$$\begin{aligned} E\{X_j\} &= \sum_{k=0}^{\infty} \sum_{i_1}^N \sum_{i_2}^N \cdots \sum_{i_k}^N b_{i_0} p_{i_0, i_1} p_{i_1, i_2} \cdots p_{i_{k-1}, i_k} w_{i_0, i_1} w_{i_1, i_2} \cdots w_{i_{k-1}, i_k} \delta_{i_k, j} \\ &= x_j . \end{aligned} \quad (19)$$

3. Model Problem

For our derivations and numerical experiments, we choose the one-speed, two-dimensional neutron diffusion equation as a model problem [13]:

$$-\nabla \cdot D \nabla \phi + \Sigma_a \phi = S , \quad (20)$$

where ϕ is the neutron flux, Σ_a is the absorption cross section, and S is the source of neutrons. In addition, D is the diffusion coefficient defined as:

$$D = \frac{1}{3(\Sigma_t - \bar{\mu}\Sigma_s)} , \quad (21)$$

where Σ_s is the scattering cross section, $\Sigma_t = \Sigma_a + \Sigma_s$ is the total cross section, and $\bar{\mu}$ is the cosine of the average scattering angle. For simplicity, we will take $\bar{\mu} = 0$ for our analysis giving $D = (3\Sigma_t)^{-1}$. In addition, to further simplify we will assume a homogeneous domain such that the cross sections remain constant throughout. Doing this permits us to rewrite Eq (20) as:

$$-D \nabla^2 \phi + \Sigma_a \phi = S . \quad (22)$$

We choose a finite difference scheme on a square Cartesian grid to discretize the problem. We choose the 9-point stencil over a grid of size h for the Laplacian [14]:

$$\begin{aligned} \nabla_9^2 \phi &= \frac{1}{6h^2} [4\phi_{i-1,j} + 4\phi_{i+1,j} + 4\phi_{i,j-1} + 4\phi_{i,j+1} + \phi_{i-1,j-1} \\ &\quad + \phi_{i-1,j+1} + \phi_{i+1,j-1} + \phi_{i+1,j+1} - 20\phi_{i,j}] . \end{aligned} \quad (23)$$

We then have the following linear system to solve:

$$\begin{aligned} -\frac{1}{6h^2} [4\phi_{i-1,j} + 4\phi_{i+1,j} + 4\phi_{i,j-1} + 4\phi_{i,j+1} + \phi_{i-1,j-1} \\ + \phi_{i-1,j+1} + \phi_{i+1,j-1} + \phi_{i+1,j+1} - 20\phi_{i,j}] + \Sigma_a \phi_{i,j} = s_{i,j} , \end{aligned} \quad (24)$$

and in operator form:

$$\mathbf{D}\phi = \mathbf{s} , \quad (25)$$

where $\mathbf{D}$ is the diffusion operator resulting from the finite difference discretization, $\mathbf{s}$ is the source in vector form and ϕ is the vector of unknown fluxes.

3.1. Domain Decomposition of the Model Problem

To investigate domain-to-domain communication in a parallel Monte Carlo algorithm, we must consider the required data with the correct parallel decomposition. In the adjoint Neumann-Ulam method, the domain decomposition consists of all states in the system that are local, and the probabilities and weights for all state transitions possible in the local domain. Given an input parallel decomposition for the linear operator, by definition in Eq (14) the probabilities and weights will have the same parallel decomposition and underlying sparse graph as the input matrix. In the case of the model problem, parallel decomposition of states and subsequently the input matrix is determined by a spatial partitioning of the grid.

Compared to the serial construction of the probabilities and weights, domain decomposition requires that data for all states be on-process so that all possible random walks that leave the local domain may be correctly communicated. As an example of walking through the sparse graph of the matrix and parallel communication, Figure 1 gives the adjacency graph of a single state in the neutron diffusion matrix for the model problem presented in Eq (24). The discretization stencil of the diffusion problem dictates the structure of this graph and the states to which a random walk may transition. Consider a random walk starting at node (i, j) in the mesh (the center node). If we play the Monte Carlo game to move to a new state, then we may move to any of the nodes in this graph, including the node at which we started with a probability proportional to the coefficients of each node in Eq (24). If any node adjacent to node (i, j) is on another processor and the walk moves to that node, the random walk must be communicated to that processor to continue the walk.

Figure 1: **Neutron diffusion equation stencil adjacency graph.** The structure of the graph comes from the discretization of the Laplacian operator describing the diffusion physics in Eq (23). A random walk at mesh point (i, j) may transition to all adjacent mesh points including itself. If a random walk transitions to a node that is not owned by the local processor, it must be communicated to that processor to continue the walk.

4. Spectral Analysis

The convergence of the Neumann series in Eq (4) approximated by the Monte Carlo solver is dependent on the eigenvalues of the iteration matrix. We will compute

these eigenvalues by assuming eigenfunctions of the form [14]:

$$\Phi_{p,q}(x, y) = e^{2\pi ipx} e^{2\pi i qy}, \quad (26)$$

where different combinations of p and q represent the different eigenmodes of the solution. As these are valid forms of the solution, then the action of the linear operator on these eigenfunctions should give the eigenvalues of the matrix as they exist on the unit circle in the complex plane.

4.1. Iteration Matrix Spectrum

We first compute the eigenvalues for the diffusion operator $\mathbf{D}$ by applying the operator to the eigenfunctions and noting that $x = ih$ and $y = jh$ on a square Cartesian grid:

$$\begin{aligned} \mathbf{D}\Phi_{p,q}(x, y) = \lambda_{p,q}(\mathbf{D}) = & -\frac{D}{6h^2} \left[4e^{-2\pi iph} + 4e^{2\pi iph} + 4e^{-2\pi iqh} + 4e^{2\pi iqh} + e^{-2\pi iph} e^{-2\pi iqh} \right. \\ & \left. + e^{-2\pi iph} e^{2\pi iqh} + e^{2\pi iph} e^{-2\pi iqh} + e^{2\pi iph} e^{2\pi iqh} - 20 \right] + \Sigma_a. \end{aligned} \quad (27)$$

Using Euler's formula, we can collapse the exponentials to trigonometric functions:

$$\lambda_{p,q}(\mathbf{D}) = -\frac{D}{6h^2} [8 \cos(\pi ph) + 8 \cos(\pi qh) + 4 \cos(\pi ph) \cos(\pi qh) - 20] + \Sigma_a. \quad (28)$$

As Eq (20) is diagonally dominant, Jacobi preconditioning is sufficient to reduce the spectral radius of the iteration matrix below unity and therefore ensure convergence of the Neumann series⁵. The preconditioner in this case is $\mathbf{M} = \text{diag}(\mathbf{D})$ such that we are solving the following linear system:

$$\mathbf{M}^{-1}\mathbf{D}\phi = \mathbf{M}^{-1}\mathbf{s}. \quad (29)$$

The operator $\mathbf{M}^{-1}\mathbf{D}$ is merely the original diffusion operator with each row scaled by the diagonal component. As we have defined a homogeneous domain, the scaling factor, α , is the same for all rows in the operator and defined as the $\phi_{i,j}$ coefficient from Eq (24):

$$\alpha = \left[\frac{10D}{3h^2} + \Sigma_a \right]^{-1}. \quad (30)$$

Using this coefficient, we have the following spectrum of preconditioned eigenvalues:

$$\lambda_{p,q}(\mathbf{M}^{-1}\mathbf{D}) = \alpha \lambda_{p,q}(\mathbf{D}). \quad (31)$$

The spectral radius of the iteration matrix is obtained by seeking its largest eigenvalue. As with the diffusion operator, we can use the same analysis techniques to find the eigenvalues for the iteration matrix. We use a few simplifications by

⁵Although $\rho(\mathbf{H}) < 1$ is a sufficient criteria for convergence of the Neumann series, recent work by Ji, Mascagni, and Li provides an additional necessary and sufficient condition for convergence of the Neumann-Ulam Monte Carlo algorithm [15]. A diagonal scaling of the model problem also satisfies this condition.

noting that if the Jacobi preconditioned iteration matrix is $\mathbf{H} = \mathbf{I} - \mathbf{M}^{-1}\mathbf{D}$, then we expect all terms on the diagonal of the iteration matrix to be zero such that we have the following stencil:

$$\mathbf{H}\phi = \frac{\alpha D}{6h^2} [4\phi_{i-1,j} + 4\phi_{i+1,j} + 4\phi_{i,j-1} + 4\phi_{i,j+1} + \phi_{i-1,j-1} + \phi_{i-1,j+1} + \phi_{i+1,j-1} + \phi_{i+1,j+1}]. \quad (32)$$

Inserting the eigenfunctions defined by Eq (26) we get:

$$\lambda_{p,q}(\mathbf{H}) = \frac{\alpha D}{6h^2} \left[4e^{-2\pi iph} + 4e^{2\pi iph} + 4e^{-2\pi iqh} + 4e^{2\pi iqh} + e^{-2\pi iph}e^{-2\pi iqh} + e^{-2\pi iph}e^{2\pi iqh} + e^{2\pi iph}e^{-2\pi iqh} + e^{2\pi iph}e^{2\pi iqh} \right], \quad (33)$$

which simplifies to:

$$\lambda_{p,q}(\mathbf{H}) = \frac{\alpha D}{6h^2} [8 \cos(\pi ph) + 8 \cos(\pi qh) + 4 \cos(\pi ph) \cos(\pi qh)], \quad (34)$$

giving the eigenvalue spectrum of the Jacobi preconditioned iteration matrix. We find that the maximum eigenvalue exists when $p = q = 0$ giving the following spectral radius of the Jacobi preconditioned iteration matrix:

$$\rho(\mathbf{H}) = \frac{10\alpha D}{3h^2}. \quad (35)$$

4.2. Neumann Series Convergence

The adjoint Monte Carlo method is effectively an approximation to a stationary method. Stationary methods for linear systems arise from splitting the operator in Eq (1) and iterating:

$$\mathbf{x}^{k+1} = \mathbf{H}\mathbf{x}^k + \mathbf{c}, \quad (36)$$

with $k \in \mathbb{Z}^+$ defined as the *iteration index* and $\mathbf{c}$ a modified source term. Defining $\mathbf{e}^k = \mathbf{u}^k - \mathbf{u}$ as the solution error at the k^{th} iterate, the error after k iterations is:

$$\mathbf{e}^k = \mathbf{H}^k \mathbf{e}^0. \quad (37)$$

By assuming $\mathbf{H}$ is diagonalizable [14], we have:

$$\|\mathbf{e}^k\|_2 \leq \rho(\mathbf{H})^k \|\mathbf{e}^0\|_2. \quad (38)$$

In the adjoint Neumann-Ulam method, k iterations, equivalent to k applications of the iteration matrix, are approximated by a random walk of average length k to yield the summation in Eq (5) [16, 17]. This random walk length, or the number of transitions before the termination of a random walk is therefore approximately the number of stationary iterations required to converge to the specified tolerance. In the case of the adjoint Neumann-Ulam method, no such tolerance exists. However, we have specified a weight cutoff, W_c , that determines when low-weight random walks will be terminated as their contributions are deemed minute. After k iterations, a stationary method is terminated as the error has reached some fraction, ϵ , of the initial error:

$$\|\mathbf{e}^k\|_2 = \epsilon \|\mathbf{e}^0\|_2. \quad (39)$$

Per Eq (38), we see that this fraction is equivalent to $\epsilon = \rho(\mathbf{H})^k$. In the adjoint method, if we take this fraction to be the weight cutoff, a measure of how accurately the contributions of a particular random walk to the solution are tallied, we have the following relationship for k :

$$k = \frac{\log(W_c)}{\log(\rho(\mathbf{H}))} . \quad (40)$$

This gives us a means to estimate the length of the random walks that will be generated from a particular linear operator based on the eigenvalues of its iteration matrix (independent of the linear operator splitting chosen) and based on the weight cutoff parameter used in the Neumann-Ulam method.

4.3. Domain Leakage Approximations

In a domain decomposed algorithm, not all random walks will remain within the domain they started in and must instead be communicated. This communication, expected to be expensive, was analyzed by Siegel and colleagues for idealized, load balanced situations for full nuclear reactor core Monte Carlo simulations [11]. To quantify the number of particles that leak out of the local domain they define a leakage fraction, Λ , as:

$$\Lambda = \frac{\text{average \# of particles leaving local domain}}{\text{total \# of particles starting in local domain}} . \quad (41)$$

For their studies, it was assumed that the value of Λ was dependent on the total cross section of the system via the Wigner rational approximation. Outlined more thoroughly by Hwang's chapter in [18], we will use both the Wigner rational approximation and the mean chord approximation as a means to estimate the leakage fraction.

In the case of parallel linear systems, we can use diffusion theory to estimate the optical thickness of a domain in the decomposition and the corresponding leakage fraction in terms of properties of the linear operator and the discretization. We first calculate the mean distance a random walk will move in the grid by computing the mean squared distance of its movement along the chord of length l defined along a cardinal direction of the domain (i.e. x or y). After a single transition a random walk will have moved a mean squared distance of:

$$\langle \bar{r}_1^2 \rangle = (n_s h)^2 , \quad (42)$$

where h is the size of the discrete grid elements along the chord and n_s is the number of grid elements a random walk will move on average every transition in the direction of the chord. For our diffusion model problem, n_s would equate to the expected number of states in the i (or j as the problem is symmetric) direction that a random walk will move in a single transition and is dependent on the stencil given in Figure 1. After k transitions, the random walk will have moved a mean squared distance of:

$$\langle \bar{r}_k^2 \rangle = k(n_s h)^2 . \quad (43)$$

If our chord is of length l and there are n_i grid elements (or states to which a random walk may transition) along that chord, then $h = l/n_i$ giving:

$$\langle \bar{r}_k^2 \rangle = k \left(\frac{n_s l}{n_i} \right)^2 . \quad (44)$$

From diffusion theory, we expect the average number of interactions along the chord to be:

$$\tau = \frac{l}{2d\sqrt{\langle r_k^2 \rangle}}, \quad (45)$$

where d is the dimensionality of the problem and $\sqrt{\langle r_k^2 \rangle}$ is effectively the mean free path of the Monte Carlo random walk in the domain. We can readily interpret τ to be the *effective optical thickness* of a domain of length l . Inserting Eq (44) we get:

$$\tau = \frac{n_i}{2dn_s\sqrt{k}}, \quad (46)$$

which if expanded with Eq (40) gives us the final relation for the effective optical thickness:

$$\tau = \frac{n_i}{2dn_s} \sqrt{\frac{\log(\rho(\mathbf{H}))}{\log(W_c)}}. \quad (47)$$

For optically thin domains, we expect that most random walks will be communicated, while optically thick domains will leak the fraction of random walks that did not interact within. Using the optical thickness defined in Eq (47), we can complete the leakage approximations by defining the bounds of $\tau \rightarrow 0, \Lambda \rightarrow 1$ and $\tau \rightarrow \infty, \Lambda \rightarrow \tau^{-1}$. With these bounds we define the leakage fraction out of a domain using the Wigner rational approximation:

$$\Lambda = \frac{1}{1 + \tau}, \quad (48)$$

and using the mean-chord approximation:

$$\Lambda = \frac{1 - e^{-\tau}}{\tau}. \quad (49)$$

Our derived leakage fraction is dependent on the eigenvalues of the iteration matrix, the size of the domain, the content of the discretization stencil, and the weight cutoff selected to terminate low weight random walks.

5. Numerical Experiments

To test the relationships developed by the spectral analysis, we form two simple numerical experiments using the diffusion model problem: one to measure the length of the random walks as a function of the iteration matrix eigenvalues, and one to measure the domain leakage fraction as a function of the iteration matrix eigenvalues and the discretization properties. Before doing this, we verify our computation of the spectral radius of the iteration matrix by numerically computing the largest eigenvalue of the diffusion operator using an iterative eigenvalue solver. For this verification, a 100×100 square grid with $h = 0.01$, $h = 0.1$, and $h = 1.0$ was used and the absorption cross varied from 0 to 100 while the scattering cross section was fixed at unity. Figure 2 gives the measured spectral radius of the iteration matrix and the computed spectral radius for the preconditioned diffusion operator using Eq (35) as function of the absorption to scattering ratio (Σ_a/Σ_s). Excellent agreement was observed between the analytic and numerical results with all data points computed within the tolerance of the iterative eigenvalue solver.

Figure 2: Measured and analytic preconditioned diffusion operator spectral radius as a function of the absorption cross section to scattering cross section ratio. Values of $h = 0.01$, $h = 0.1$, and $h = 1.0$ were used. The colored data was computed numerically by an eigensolver while the black dashed data was generated by Eq (35).

5.1. Random Walk Length

We next create an experiment to measure the length of the random walks generated by the adjoint Neumann-Ulam solver. To do this, we again use a 100×100 square grid with $h = 0.1$ and the absorption cross varied from 0 to 100 while the scattering cross section was fixed at unity. Three weight cutoff values of 1×10^{-2} , 1×10^{-4} , and 1×10^{-8} were used with 10,000 random walks generated by a point source of strength 1 in the center of the domain. For each of the random walks, the number of transitions made was tallied to provide an effective value of k for each random walk. This value was then averaged over all random walks to get a measured value of k for the particular operator. Figure 3 presents these measurements as well as the analytic result computed by Eq (40) as a function of the iteration matrix spectral radius, $\rho(\mathbf{H})$. Figure 4 gives the relative error between the predicted and observed results. We note good qualitative agreement between the measured and analytic results. However, we observe a larger relative error for both long and short random walks.

5.2. Domain Leakage

Finally, we measure the leakage of random walks from a domain in a Monte Carlo calculation and assess the quality of our analytic relation for the optical thickness of the domain and the associated leakage approximations. For this experiment, a square grid with $h = 0.1$ was decomposed into 9 square domains, 3 in each cardinal direction with measurements occurring in the central domain without boundary grid points. For cross sections, the absorption cross section was varied from 1 to 100 while the scattering cross section was set to zero to create a purely absorbing environment with weight cutoff of 1×10^{-4} . The optical thickness of these domains

Figure 3: **Measured and analytic random walk length as a function of the iteration matrix spectral radius.** The weight cutoff was varied with 1×10^{-2} , 1×10^{-4} , and 1×10^{-8} . The colored data was computed numerically by an adjoint Neumann-Ulam implementation while the black dashed data was generated by Eq (40).

Figure 4: **Relative error between computed random walk length analytic length computed by Eq (40) as a function of the iteration matrix spectral radius.** The weight cutoff was varied with 1×10^{-2} , 1×10^{-4} , and 1×10^{-8} .

will vary as a function of the absorption cross section if the other parameters are fixed.

To compute the optical thickness, along with the spectral radius as given by

Eq (35), we also need the parameters n_i and n_s which respectively describe the typical domain length and the average number of states moved along that typical length per random walk transition. For our two-dimensional Cartesian grid, the domains are varied in size with 50×50 , 100×100 , and 200×200 cells giving $n_i = 50$, $n_i = 100$, and $n_i = 200$ grid points or states along the typical length of the domain respectively.

To compute n_s , again consider the graph of the diffusion matrix stencil given in Figure 1. We see that all random walk transitions will only move a single state in either the i or j directions due to the symmetry of the problem. Furthermore, if we choose a chord in the i direction, not all states to which we will transition move the random walk in that direction. Instead, n_s will be representative of the average movement in the i direction based on probabilities defined by the definition of the iteration matrix in Eq (32) and the definition of the adjoint probability matrix in Eq (14). For a particular transition starting at state (i, j) , 6 of the 8 possible new states in the stencil move the random walk in i direction with relative coefficients of 4 for moving in the $(\pm i, 0)$ direction and of 1 for moving in the $(\pm i, \pm j)$. These coefficients dictate the frequency those states are visited relative to the others. For those 6 states we can visit along the typical length, their sum is 12 out of the total 20 for the coefficients for all possible states with their ratio giving $n_s = \frac{3}{5}$.

To compute the leakage fraction numerically, 3×10^5 random walks were sampled from a uniform source of strength unity over the global domain. At the start of a stage of random walks, the number of random walks starting in the center domain was computed and as the stage progressed, the number of random walks that exited that domain was tallied with the ratio of the two numbers providing a numerical measure of the leakage fraction. Figure 5 gives the domain leakage measurements for the domain in the center of the global grid as well as the analytic result computed by Eqs (48) and (49) as a function of the iteration matrix spectral radius.

Again, we note good qualitative agreement between the measured and analytic quantities but we begin to see the limits of the leakage approximations. To compare the quality of the two approximations, the absolute error between the computed leakage fraction and that generated by the Wigner rational and mean chord approximations is plotted in Figure 6 for all domain sizes tested. From these error results, the mean chord approximation is shown to have a lower error for ill-conditioned systems as compared to the Wigner approximation while the Wigner approximation produces less error for more well-conditioned systems. We also note that for optically thick domains, the error is likely corresponded to that observed in Figure 3 for the k parameter while the large relative error in k for optically thin domains does not affect the approximation significantly. In general, the mean chord approximation is a better choice to estimate the leakage fraction in a domain from the adjoint Neumann-Ulam method and except for a single data point with $n_i = 50$, the mean chord approximation yielded leakage fractions within 0.05 of the measured results. As the domain becomes more optically thick (with both increasing n_i and decreasing $\rho(\mathbf{H})$), the approximations are more accurate.

6. Conclusion

We have presented an analytic analysis of the domain decomposed behavior of the adjoint Neumann Ulam method for linear systems. Good agreement was

Figure 5: **Measured and analytic domain leakage as a function of the iteration matrix spectral radius.** To test the behavior with respect to domain size, $n_i = 50$, $n_i = 100$, and $n_i = 200$ were used. The colored data was computed numerically by a domain-decomposed adjoint Neumann-Ulam implementation, the black dashed data was generated by Eq (49) using the mean-chord approximation, and the dashed-dotted black data was generated by Eq (48) using the Wigner rational approximation.

Figure 6: **Measured and analytic domain leakage absolute error as a function of the iteration matrix spectral radius.** To test the behavior with respect to domain size, $n_i = 50$, $n_i = 100$, and $n_i = 200$ were used. The dashed lines represent the error using the Wigner rational approximation while the solid lines represent the error using the mean-chord approximation.

observed for the derived analytic relationships for random walk length and leakage fraction when compared to numerical results generated by a domain decomposed implementation of the adjoint method.

In future work, these relationships will serve as guidelines for selecting an appropriate parallel algorithm strategy and provide a basis for performance models for future parallel implementations of the adjoint Neumann-Ulam method. To be applicable to a broader range of problems, these models could potentially be extended to non-symmetric and three-dimensional systems. In addition, these relations will be used to analyze parallel implementations of Monte Carlo synthetic acceleration methods that leverage the adjoint method and serve as an initial grounds for assessing their feasibility for large-scale problems.

7. Acknowledgments

This work was performed under appointment to the Nuclear Regulatory Commission Fellowship program at the University of Wisconsin - Madison Engineering Physics Department.

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research program.

- [1] G. E. Forsythe and R. A. Leibler, “Matrix inversion by a Monte Carlo method,” *Mathematical Tables and Other Aids to Computation*, vol. 4, pp. 127–129, July 1950.
- [2] W. Wasow, “A note on the inversion of matrices by random walks,” *Mathematical Tables and Other Aids to Computation*, vol. 6, pp. 78–81, Apr. 1952.
- [3] J. M. Hammersley and D. C. Handscomb, *Monte Carlo Methods*. Methuen, 1964.
- [4] J. Spanier and E. M. Gelbard, *Monte Carlo Principles and Neutron Transport Problems*. New York: Dover Publications, 1969.
- [5] J. H. Halton, “Sequential Monte Carlo,” *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 58, no. 01, pp. 57–78, 1962.
- [6] T. Evans and S. Mosher, “A Monte Carlo synthetic acceleration method for the non-linear, time-dependent diffusion equation,” *American Nuclear Society - International Conference on Mathematics, Computational Methods and Reactor Physics 2009*, 2009.
- [7] T. Evans, S. Mosher, S. Slattery, and S. Hamilton, “A monte carlo synthetic-acceleration method for solving the thermal radiation diffusion equation,” *Journal of Computational Physics*, vol. 258, pp. 338–358, 2014.
- [8] V. Alexandrov, “Efficient parallel Monte Carlo methods for matrix computations,” *Mathematics and Computers in Simulation*, vol. 47, pp. 113–122, Aug. 1998.
- [9] T. A. Brunner, T. J. Urbatsch, T. M. Evans, and N. A. Gentile, “Comparison of four parallel algorithms for domain decomposed implicit Monte Carlo,” *Journal of Computational Physics*, vol. 212, pp. 527–539, Mar. 2006.

- [10] T. A. Brunner and P. S. Brantley, “An efficient, robust, domain-decomposition algorithm for particle Monte Carlo,” *Journal of Computational Physics*, vol. 228, pp. 3882–3890, June 2009.
- [11] A. Siegel, K. Smith, P. Fischer, and V. Mahadevan, “Analysis of communication costs for domain decomposed Monte Carlo methods in nuclear reactor analysis,” *Journal of Computational Physics*, vol. 231, pp. 3119–3125, Apr. 2012.
- [12] W. D. Gropp, D. K. Kaushik, D. E. Keyes, and B. F. Smith, “High-performance parallel implicit CFD,” *Parallel computing in aerospace*, vol. 27, pp. 337–362, Mar. 2001.
- [13] J. J. Duderstadt and L. J. Hamilton, *Nuclear Reactor Analysis*. Wiley, 1 ed., Jan. 1976.
- [14] R. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*. SIAM, Society for Industrial and Applied Mathematics, July 2007.
- [15] H. Ji, M. Mascagni, and Y. Li, “Convergence Analysis of Markov Chain Monte Carlo Linear Solvers Using Ulam-Von Neumann Algorithm,” *Siam J. Numer. Anal.*, vol. 51, no. 4, pp. 2107–2122, 2013.
- [16] I. Dimov, T. Dimov, and T. Gurov, “A new iterative Monte Carlo approach for inverse matrix problem,” *Journal of Computational and Applied Mathematics*, vol. 92, pp. 15–35, May 1998.
- [17] D. Danilov, S. Ermakov, and J. Halton, “Asymptotic complexity of Monte Carlo methods for solving linear systems,” *Journal of Statistical Planning and Inference*, vol. 85, pp. 5–18, Apr. 2000.
- [18] Y. Azmy and E. Sartori, *Nuclear Computational Science: A Century in Review*. Springer, 1st edition. ed., May 2010.

References

- [1] G. E. Forsythe and R. A. Leibler, “Matrix inversion by a Monte Carlo method,” *Mathematical Tables and Other Aids to Computation*, vol. 4, pp. 127–129, July 1950.
- [2] W. Wasow, “A note on the inversion of matrices by random walks,” *Mathematical Tables and Other Aids to Computation*, vol. 6, pp. 78–81, Apr. 1952.
- [3] J. M. Hammersley and D. C. Handscomb, *Monte Carlo Methods*. Methuen, 1964.
- [4] J. Spanier and E. M. Gelbard, *Monte Carlo Principles and Neutron Transport Problems*. New York: Dover Publications, 1969.
- [5] J. H. Halton, “Sequential Monte Carlo,” *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 58, no. 01, pp. 57–78, 1962.
- [6] T. Evans and S. Mosher, “A Monte Carlo synthetic acceleration method for the non-linear, time-dependent diffusion equation,” *American Nuclear Society - International Conference on Mathematics, Computational Methods and Reactor Physics 2009*, 2009.
- [7] T. Evans, S. Mosher, S. Slattery, and S. Hamilton, “A monte carlo synthetic-acceleration method for solving the thermal radiation diffusion equation,” *Journal of Computational Physics*, vol. 258, pp. 338–358, 2014.
- [8] V. Alexandrov, “Efficient parallel Monte Carlo methods for matrix computations,” *Mathematics and Computers in Simulation*, vol. 47, pp. 113–122, Aug. 1998.
- [9] T. A. Brunner, T. J. Urbatsch, T. M. Evans, and N. A. Gentile, “Comparison of four parallel algorithms for domain decomposed implicit Monte Carlo,” *Journal of Computational Physics*, vol. 212, pp. 527–539, Mar. 2006.
- [10] T. A. Brunner and P. S. Brantley, “An efficient, robust, domain-decomposition algorithm for particle Monte Carlo,” *Journal of Computational Physics*, vol. 228, pp. 3882–3890, June 2009.
- [11] W. D. Gropp, D. K. Kaushik, D. E. Keyes, and B. F. Smith, “High-performance parallel implicit CFD,” *Parallel computing in aerospace*, vol. 27, pp. 337–362, Mar. 2001.
- [12] J. J. Duderstadt and L. J. Hamilton, *Nuclear Reactor Analysis*. Wiley, 1 ed., Jan. 1976.
- [13] R. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*. SIAM, Society for Industrial and Applied Mathematics, July 2007.
- [14] H. Ji, M. Mascagni, and Y. Li, “Convergence Analysis of Markov Chain Monte Carlo Linear Solvers Using Ulam-Von Neumann Algorithm,” *Siam J. Numer. Anal.*, vol. 51, no. 4, pp. 2107–2122, 2013.

- [15] I. Dimov, T. Dimov, and T. Gurov, “A new iterative Monte Carlo approach for inverse matrix problem,” *Journal of Computational and Applied Mathematics*, vol. 92, pp. 15–35, May 1998.
- [16] D. Danilov, S. Ermakov, and J. Halton, “Asymptotic complexity of Monte Carlo methods for solving linear systems,” *Journal of Statistical Planning and Inference*, vol. 85, pp. 5–18, Apr. 2000.
- [17] A. Siegel, K. Smith, P. Fischer, and V. Mahadevan, “Analysis of communication costs for domain decomposed Monte Carlo methods in nuclear reactor analysis,” *Journal of Computational Physics*, vol. 231, pp. 3119–3125, Apr. 2012.
- [18] Y. Azmy and E. Sartori, *Nuclear Computational Science: A Century in Review*. Springer, 1st edition. ed., May 2010.