SIAM J. SCI. COMPUT.
Vol. 35, No. 5, pp. S62–S83

HYBRID DETERMINISTIC/MONTE CARLO NEUTRONICS*

JEFFREY WILLERT†, C. T. KELLEY†, D. A. KNOLL‡, AND H. PARK‡

Abstract. In this paper we describe a hybrid deterministic/Monte Carlo algorithm for neutron transport simulation. The algorithm is based on nonlinear accelerators for source iteration, using Monte Carlo methods for the purely absorbing high-order problem and a Jacobian-free Newton–Krylov iteration for the low-order problem. We couple the Monte Carlo solution with the low-order problem using filtering to smooth the flux and current from the Monte Carlo solver and an analytic Jacobian-vector product to avoid numerical differentiation of the Monte Carlo results. We use a continuous energy deposition tally for the Monte Carlo simulation. We conclude the paper with numerical results which illustrate the effectiveness of the new algorithm.

Key words. JFNK methods, neutron transport, Monte Carlo simulation, hybrid methods

AMS subject classifications. 45L10, 65H10, 82D75

DOI. 10.1137/120880021

1. Introduction. In this paper we propose and study a hybrid deterministic/Monte Carlo (MC) solver for neutron transport. We base this algorithm on a standard deterministic neutron transport accelerator for source iteration, nonlinear diffusion acceleration (NDA) [9], and MC methods for the solution of the underlying transport equation. We study the performance of the algorithm using a one-dimensional, slab geometry, fixed source, model problem. Previous work [9], which combined the Jacobian-free Newton–Krylov (JFNK) method with NDA to provide source iteration acceleration for a deterministic transport solver, partially motivates the work in this paper. In [9] the Newton-based accelerator outperformed the previous Picard-based accelerator in certain regimes.

Acceleration techniques based on NDA are a popular approach in nuclear reactor physics applications [1, 19]. Until recently, the NDA method has been limited to acceleration of the deterministic transport methods. MC methods have the advantage of not requiring discretization of the phase space variables of space, angle, and energy, and have significant advantages in high-fidelity simulations. Moreover, this geometrical flexibility of MC can alleviate complex meshing issues arising from the detailed reactor core simulations. Continuous angular treatment can treat highly anisotropic solutions more efficiently. Furthermore, (nearly) continuous energy MC can bypass the preprocessing of cross section libraries, which is required by all deterministic transport calculations.

Although the MC method can provide a more accurate solution and has more flexibility, it is very computationally intensive in diffusive regions. Several recent papers [12, 13, 14, 21] have studied accelerating the convergence of the fission source

*Received by the editors June 6, 2012; accepted for publication (in revised form) January 24, 2013; published electronically October 28, 2013.
http://www.siam.org/journals/sisc/35-5/88002.html
†North Carolina State University, Department of Mathematics, Box 8205, Raleigh, NC 27695-8205 (Tim_Kelley@ncsu.edu, jawiller@ncsu.edu). The work of these authors has been partially supported by the Consortium for Advanced Simulation of Light Water Reactors (www.casl.gov), an Energy Innovation Hub (http://www.energy.gov/hubs) for Modeling and Simulation of Nuclear Reactors under U.S. Department of Energy Contract DE-AC05-00OR22725, National Science Foundation Grant CDI-0941253, and Army Research Office Grants W911NF-11-1-0367, and W911NF-07-1-0112.
‡Theoretical Division, MS B216, Los Alamos National Laboratory, Los Alamos, NM 87545 (nol@lanl.gov, hkpark@lanl.gov).

for the reactor eigenvalue problems. In this work, we will extend the previous work to acceleration of (isotropic) scattering sources.

Here we focus solely on the Newton-based accelerator and consider MC methods for the solution of the underlying transport equation. With the combination of nonlinear elimination [9, 11, 22] and the JFNK method, the dependent variable in the equations is the scalar flux, which depends only on space. We derive a new analytic formula for the required Jacobian-vector product within the JFNK algorithm. The motivation for this is to alleviate the MC noise issues which will be amplified within the standard finite-difference Jacobian-vector product approximation. We also consider the impact on algorithm performance driven by different choices for MC tally procedures.

In section 2 we provide a coarse overview of the neutron transport equation, source iteration, and nonlinear acceleration of source iteration. We describe the use of MC simulation to solve the neutron transport equation in section 3. In section 4 we show how JFNK-NDA must be modified if we use MC methods for the transport sweep. We formally describe the JFNK-NCA(MC) algorithm in section 5 and directly show how the number of particles in the MC simulation must be reflected in the nonlinear iteration. We report results of some computational experiments in section 6.

2. The transport equation and nonlinear accelerators. We consider the monoenergetic transport equation in slab geometry with isotropic scattering [3, 15]

$$\mu\frac{\partial\psi(x,\mu)}{\partial x}+\Sigma_t(x)\psi(x,\mu)=\frac{1}{2}\left[\Sigma_s(x)\int_{-1}^{1}\psi(x,\mu')\,d\mu'+q(x)\right] \tag{2.1}$$

for $0<x<\tau$ and $\mu\in[-1,0)\cup(0,1]$. We impose boundary conditions

$$\psi(0,\mu)=\psi_l(\mu),\mu>0;\psi(\tau,\mu)=\psi_r(\mu),\mu<0. \tag{2.2}$$

In (2.1)

- ψ is intensity of radiation or angular flux at point x at angle $\cos^{-1}(\mu)$;
- $\phi=\phi(x)=\int_{-1}^{1}\psi(x,\mu)\,d\mu$ is the scalar flux, the 0th angular moment of the angular flux;
- $\tau<\infty$, length of the spatial domain;
- $\Sigma_s\in C([0,\tau])$ is the scattering cross section at x;
- $\Sigma_t\in C([0,\tau])$ is the total cross section at x;
- ψ_l and ψ_r are incoming intensities at the bounds;
- $q\in C([0,\tau])$ is the fixed source.

2.1. Discretizing the transport equation. When solving the transport equation deterministically, we must discretize in angle and space. In this paper, we utilize the S_N [15] method for our angular discretization whenever a deterministic solution to the transport equation is desired.

The S_N method approximates the neutron transport equation at a finite set of angles, $\{\mu_j\}_{j=1}^{N}$, yielding the following set of coupled ODEs:

$$\mu_j\frac{\partial\psi^j(x)}{\partial x}+\Sigma_t(x)\psi^j(x)=\frac{1}{2}\left[\Sigma_s(x)\phi(x)+q(x)\right], \tag{2.3}$$

where $\psi^j(x)\approx\psi(x,\mu_j)$. It is customary to choose the μ's as the nodes from a Gaussian quadrature rule so that along with the associated weights w_j, we can approximate

the angular moments, ϕ and J, of ψ using a simple summation:

$$\phi(x) = \int_{-1}^{1} \psi(x,\mu)d\mu \approx \sum_{j=1}^{N} w_j \psi^j(x),$$

$$J(x) = \int_{-1}^{1} \mu\psi(x,\mu)d\mu \approx \sum_{j=1}^{N} \mu_j w_j \psi^j(x).$$

We use a simple *diamond difference* spatial differencing scheme

$$\mu_j \frac{\psi^j_{i+\frac{1}{2}} - \psi^j_{i-\frac{1}{2}}}{\Delta x_i} + \Sigma_{t,i} \frac{\psi^j_{i+\frac{1}{2}} + \psi^j_{i-\frac{1}{2}}}{2} = \frac{1}{2}\left[\Sigma_{s,i}\phi_i + q_i\right]$$

in which $\psi^j_{i-\frac{1}{2}} = \psi(x_{i-\frac{1}{2}}, \mu_j)$, $\Delta x_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}$, and $\Sigma_{k,i} = \Sigma_k(x_i)$. With this discretization, the angular fluxes are stored at cell edges and scalar fluxes are stored at cell centers. Our angular integration formulas now become

$$\phi_i = \sum_{j=1}^{N} w_j \frac{\psi^j_{i+\frac{1}{2}} + \psi^j_{i-\frac{1}{2}}}{2} \text{ and}$$

$$J_{i+\frac{1}{2}} = \sum_{j=1}^{N} \mu_j w_j \psi^j_{i+\frac{1}{2}}.$$

2.2. Source iteration. A classical deterministic approach to solving for ψ is *source iteration*. Source iteration takes a current approximation for the flux ϕ_c, performs a transport sweep to obtain an output intensity ψ_+ by solving

$$\mu\frac{\partial\psi_+}{\partial x} + \Sigma_t\psi_+(x,\mu) = \frac{1}{2}\left[\Sigma_s\phi_c(x) + q(x)\right], \tag{2.4}$$

and then updates ϕ_c to ϕ_+ by

$$\phi_+(x) = \int_{-1}^{1} \psi_+(x,\mu')\, d\mu'.$$

The map from ϕ_c to ϕ_+ is called the *source iteration map.* The problem has been solved when $\phi_c = \phi_+$. The unit of work in the solution of the neutron transport equation is a *transport sweep*, the action of recovering ψ_+ from ϕ_c. We wish to minimize the number of transport sweeps needed to find a solution.

If $\tau < \infty$ and $0 \le \Sigma_s(x)/\Sigma_t(x) \le 1$ for all x, which we will assume, then source iteration will converge, but perhaps slowly. When τ is large and/or $\Sigma_s(x)/\Sigma_t(x) \approx 1$ in some regions of the domain, convergence may slow dramatically. One could also apply a Krylov method, such as GMRES [18] to solve the equivalent integral equation [7, 17, 20], but slow convergence can also be a problem there without a quality preconditioner.

In [17, 20], the required transport sweep is executed with a deterministic approach. We could also execute the transport sweep with MC simulations. These simulations may be more expensive, but may give better results in many cases than deterministic simulations. In this study we consider this approach inside of an NDA method [9]. One could also consider a similar moment-based accelerator, *quasi diffusion* (QD) [5, 16], but we limit our scope to NDA for this paper.

2.3. NDA. In this section we follow a similar approach as found in [9] which can be consulted for more detail. NDA makes use of the 0th angular moment of the

transport equation,

$$\frac{dJ}{dx} + (\Sigma_t - \Sigma_s)\phi = q. \tag{2.5}$$

Then NDA will enforce that the "high-order" (HO) for ψ and the "low-order" (LO) equation for ϕ remain consistent down to truncation error. We enforce the consistency with the augmented diffusion current closure,

$$J = -\frac{1}{3\Sigma_t}\frac{d\phi}{dx} + \hat{D}\phi. \tag{2.6}$$

The LO problem is defined by substituting (2.6) into (2.5):

$$\frac{d}{dx}\left[-\frac{1}{3\Sigma_t}\frac{d\phi}{dx} + \hat{D}^{HO}\phi\right] + (\Sigma_t - \Sigma_s)\phi = q. \tag{2.7}$$

To insure consistency, $\hat{D}^{HO}$ is computed using a flux and current from the HO solution,

$$\mu\frac{\partial\psi^{HO}}{\partial x} + \Sigma_t\psi^{HO}(x,\mu) = \frac{1}{2}\left[\Sigma_s\phi^{LO}(x) + q(x)\right], \tag{2.8}$$

via

$$\hat{D} = \frac{J^{HO} + \frac{1}{3\Sigma_t}\frac{d\phi^{HO}}{dx}}{\phi^{HO}}, \tag{2.9}$$

where

$$\phi^{HO}(x) = \int_{-1}^{1}\psi^{HO}(x,\mu')\,d\mu' \text{ and } J^{HO}(x) = \int_{-1}^{1}\psi^{HO}(x,\mu')\mu'\,d\mu'.$$

Again, for more detail, including boundary conditions for the LO problem, see [9].

Note that in (2.7), the scattering term is solved implicitly, not lagged as it is in source iteration. When $\hat{D}$ is evaluated using the HO solution moments and the same spatial discretization as the LO problem, then the solution of the LO problem is guaranteed to have discrete consistency with the solution of the HO problem upon convergence.

The coupled HO-LO problem may be solved using fixed-point iteration as follows. We express the fixed-point iteration formally in Algorithm Picard. The inputs to the algorithm are an initial iterate ϕ^{LO} for the LO flux and relative and absolute error tolerances $\tau_r > 0$ and $\tau_a \geq 0$.

PICARD $(\phi^{LO}, \tau_r, \tau_a)$.
  $\Delta = 2\tau_r\|\phi^{LO}\| + \tau_a + 1.$
  while $\Delta > \tau_r\|\phi^{LO}\| + \tau_a$ **do**
    Solve (2.8) for ψ^{HO}.
    Compute ϕ^{HO} and J^{HO} from ψ^{HO}.
    Compute $\hat{D}$ from (2.9).
    Solve (2.7) for ϕ^{new} to obtain ϕ.
    $\Delta = \|\phi - \phi^{LO}\|$
    $\phi^{LO} \leftarrow \phi$
  end while

We can turn this into a single equation by enslaving (2.4) inside the calculation of $\hat{D}$. In this case we write

$$\hat{D} = \hat{D}(G(\phi)) = \hat{D}(\phi^{HO}, J^{HO}),$$

where G represents the execution of a transport sweep. Now, the transport equation has been solved if

$$\frac{d}{dx}\left[-\frac{1}{3\Sigma_t}\frac{d\phi}{dx} + \hat{D}(\phi^{HO}, J^{HO})\phi\right] + (\Sigma_t - \Sigma_s)\phi = q \tag{2.10}$$

or if $F(\phi) = 0$, where F is given by

$$F(\phi) = \frac{d}{dx}\left[\frac{-1}{3\Sigma_t}\frac{d\phi}{dx} + \hat{D}(\phi^{HO}, J^{HO})\phi\right] + (\Sigma_t - \Sigma_s)\phi - q \tag{2.11}$$

with appropriate boundary conditions.

We can solve $F(\phi) = 0$ using a JFNK algorithm. This will be described in greater detail in section 5. Note that $\hat{D}$ depends on ϕ through the transport sweep, so the equation is nonlinear.

3. Monte Carlo simulation. An MC simulation is often preferred for a nuclear reactor physics calculation due to its (nearly) continuous treatment of energy, angular and spatial variables, and this type of resolution is often crucial for accurate analysis of reactor behavior. NDA is a widely used method for the deterministic solution of the transport equation [1, 19]. Although there have been several recent attempts to extend the NDA methodology employing a stochastic HO solver [12, 13, 14, 21], they are mainly focused on accelerating on the fission source, not the scattering source. Here, we extend the work by Knoll [9] to the context of a stochastic HO solver, which accelerates the scattering source.

3.1. Conventional Monte Carlo tallying. A conventional MC method simulates the physical processes of neutral particles, such as streaming, collision, and emission. In order to obtain a statistical average, the physical events are "tallied." One of the widely accepted tallies is the "track-length" tally [15], which accumulates a particle's flight paths. The track-length tally estimates the average scalar flux ϕ in the volume V with N_{MC} particle histories as follows:

$$\phi = \frac{1}{N_{MC}V}\sum_{n=1}^{N_{MC}} w_n l_n, \tag{3.1}$$

where w_n and l_n are the weight and distance traveled by the nth particle, respectively. The track-length tally generally gives a lower variance than the collision estimator tally [15],

$$\phi = \frac{1}{N_{MC}V\Sigma_t}\sum_{n=1}^{N_{MC}} w_n, \tag{3.2}$$

which values are accumulated only at collision sites.

In NDA, the HO solution is used to compute the consistency term at cell faces,

$$\hat{D}_{i\pm1/2} = \frac{J^{HO}_{i\pm1/2} + \frac{1}{3\Sigma_t}\frac{d\phi}{dx}\big|_{i\pm1/2}}{\phi_i + \phi_{i\pm1}}. \tag{3.3}$$

We estimate the current at a cell face, $J^{HO}_{i\pm1/2}$, with the surface crossing tally,

$$J = \frac{1}{N_{MC}A}\sum_{n=1}^{N_{MC}} sign(\mu)w_n, \tag{3.4}$$

where A is the surface area.

3.2. Monte Carlo transport sweep. A stand-alone MC simulation solves (2.1). Note that in (2.1), we are inverting the streaming, absorption, and scattering operators. However, the scattering operator is considered to be a component of the source term in NDA, which greatly simplifies the MC logistics.

Again, a transport sweep returns a new angular flux, ψ_+, given the current best guess at the scalar flux, ϕ_c. That is, we solve

$$\mu\frac{\partial\psi_+}{\partial x} + \Sigma_t\psi_+ = \frac{1}{2}\left[\Sigma_s\phi_c + q\right]. \tag{3.5}$$

It is clear that this action of computing a new angular flux needs only a source function for the right-hand side. It matters very little whether this source is a fixed source or a scattering source. The transport sweep takes the current estimate of the flux and allows each particle to take a single extra flight path before it is absorbed. In this case the scattering term, $\Sigma_s\phi$, only helps build the source function for the right-hand side.

We can approximate the transport sweep using a MC simulation. We use our current iterate ϕ_c to build a new source function, $S(x)$,

$$S(x) = \Sigma_s\phi_c + q. \tag{3.6}$$

In essence, we are solving a new problem,

$$\mu\frac{\partial\psi}{\partial x} + \Sigma_t\psi = \frac{1}{2}S(x). \tag{3.7}$$

We now conduct an MC simulation for which $\Sigma_s = 0$. This means that every particle has zero possibility of scattering. Therefore, after a single particle flight (from the particle birthplace to the first collision) the particle history is terminated. This effectively allows for each particle in the simulation to undergo one single additional interaction.

It is important to note that these MC transport sweeps are very cheap (in comparison to classic MC) to compute. Since particles can no longer scatter, each particle history is terminated after only a few brief calculations. Furthermore, each particle affects only a minimal number of cells, which means the tallying process can occur much more quickly as well. These MC transport sweeps can be used in place of any deterministic transport sweep, given that we use enough particles to ensure a physically accurate simulation.

3.3. Continuous energy deposition tallies. In section 3.1, we discussed the most basic method for tallying, which we will refer to as standard tallying for the remainder of this paper. As it stands, however, standard tallying is suboptimal for the following reason.

Let us begin by taking the zeroth angular moment of (3.7). This yields the following (discrete) neutron balance equation in cell i,

$$\frac{J_{i+1/2} - J_{i-1/2}}{\Delta x_i} + \Sigma_{t,i}\phi_i = \frac{1}{2}S_i. \tag{3.8}$$

When we tally the (cell-average) scalar flux and (cell-face) current, ϕ_i and $J_{i\pm 1/2}$, respectively, with conventional track-length and surface crossing estimators, we find that (3.8) is not satisfied. In fact, if we define a residual based on this balance equation,

$$R_i(\Delta x_i; \phi_i, J_{i\pm 1/2}) = \frac{J_{i+1/2} - J_{i-1/2}}{\Delta x_i} + \Sigma_{t,i}\phi_i - \frac{1}{2}S_i, \tag{3.9}$$

we find that $||R|| \gg 0$ when standard tallies are used to compute ϕ_i and $J_{i\pm 1/2}$. Along with the noise we find in the MC transport sweeps, this lack of balance can be detrimental when trying to compute the consistency term $\hat{D}$ and trying to solve the problem.

Now, we present a second tally which we will refer to as the *continuous energy deposition tally* or *CED tally* [4]. This tally is especially suitable for NDA-MC since the MC simulation solves a problem with purely absorbing media. The purpose of the CED tally is to conserve the balance of neutrons which we expect.

The CED tally takes place in two distinct phases:

1. A nonanalog tally in which each particle is forced to travel a fixed mean-free-path distance in which we tally using exponentially decreasing weights.
2. A standard tally from the endpoint of the nonanalog tally using the remaining particle weight.

Our new MC simulation takes place as described in Algorithm CED Monte Carlo Tally.

The CED tally satisfies the balance equation to the point of machine roundoff. We can see this in Figure 3.1.

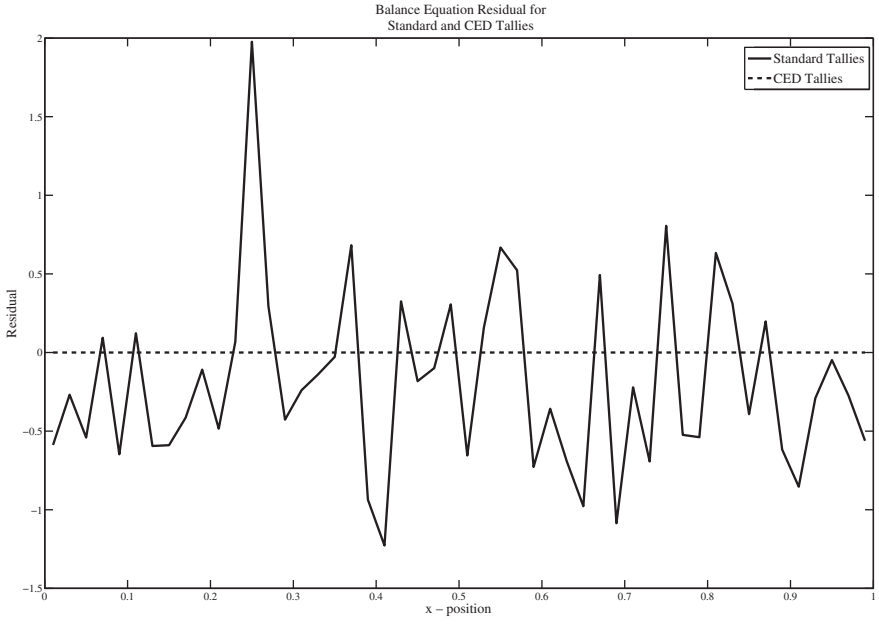

FIG. 3.1. *The neutron balance equation is not satisfied using standard tallies. Using CED tallies, the neutron balance equation is satisfied down to a level of* 10^{-13}. *This result comes from computations using Test* 1 *in section* 6.

CED Monte Carlo Tally (N_{MC}, λ).
for $n = 1$ to N_{MC} **do**
Randomly sample direction cosine μ and particle location x.
Initialize weight for particle n in cell i, w_i.
Set $\lambda_n = 0$.
while $\lambda_n < \lambda$ and particle n remains in the system **do**
Compute the mean-free-path distance particle n travels in cell i:

$$\Delta\lambda_i = \frac{\Delta x_i \Sigma_{t,i}}{|\mu|}. \tag{3.10}$$

Compute the deposited weight in cell i, Δw_i:

$$\Delta w_i = w_i \left(1 - e^{-\Delta\lambda_i}\right). \tag{3.11}$$

Tally the scalar flux using the weight Δw_i using the collision estimator described in section 3.1.
Compute weight at cell interface:

$$w_{i\pm1} = w_i e^{-\Delta\lambda_i}. \tag{3.12}$$

Tally the surface current using the weight $w_{i\pm1}$ using the face tally described in section 3.1.
Update cell location $i = i \pm 1$.
end while
Begin analog tally from new particle location using $w = w_{i\pm1}$ and direction cosine μ.
end for

4. JFNK-NDA(MC). In section 2.3 we presented the NDA algorithm and demonstrated that we could solve the transport equation by finding a root of the nonlinear function, $F(\phi)$, given by

$$F(\phi) = \frac{d}{dx}\left[-\frac{1}{3\Sigma_t}\frac{d\phi}{dx} + \hat{D}(\phi^{HO}, J^{HO})\phi\right] + (\Sigma_t - \Sigma_s)\phi - q.$$

Within the computation of F, we must take the input ϕ and execute a single transport sweep to retrieve a new HO angular flux,

$$\psi^{HO} = G(\phi).$$

Given ψ^{HO}, we can compute the 0th and 1st angular moments, ϕ^{HO} and J^{HO}, respectively.

As was demonstrated in section 3.2, we can compute $G(\phi)$ using an MC simulation. In this case, we will represent the MC transport sweep by $G^{MC}(\phi)$. It is important to recall that an MC transport sweep is the result of a stochastic simulation and the results contain some amount of noise based on the number of particles used in the simulation. Using the central limit theorem as described in [15], we can say

$$\text{Prob}\left(\|\phi^* - \phi\| < \frac{c}{\sqrt{N}}\right) = \text{Erf}\left(\frac{c}{\sqrt{2}\sigma}\right),$$

where

$$\mathrm{Erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-x^2}\, dx,$$

and ϕ^* is the expected value.

We propose a new algorithm, *JFNK-NDA(MC)*, in which we solve $F(\phi) = 0$ using a JFNK method, and solve the HO problem with MC.

4.1. Implementation challenges. While JFNK-NDA has excellent convergence properties, JFNK-NDA(MC) will not converge unless some minor modifications are made. The reason for this convergence difficulty can be traced to the stochastic noise in the function evaluations.

In these MC transport sweeps, we can only use a finite number of particles within the simulation. Therefore, we must be willing to accept some level of noise. These noisy functions can be a challenge in one of two ways:

1. the computation of finite-difference derivatives $\frac{d\phi}{dx}$ and $\frac{d}{dx}(\hat{D}\phi)$;
2. the computation of finite-difference Jacobian-vector products.

Fortunately, we can handle each of these issues with a simple change in the algorithm. By implementing intelligent filters, we can eliminate most or all of the noise associated with an MC transport sweep. After applying filters, we can compute F using smooth functions. Furthermore, we have developed an analytic formula for the Jacobian-vector product that does not require finite differences. These items will be discussed in greater detail in sections 4.1.1 and 4.1.2.

4.1.1. Filtering. With either the standard tally or the CED tally, our function evaluations will return fluxes and currents with some amount of noise. In Figures 4.1 and 4.2, we see the type of noise we must deal with when considering the scalar flux and current.

If possible, we'd like to avoid differentiating the scalar flux and current when they are the result of a noisy MC function evaluation. In essence, we must find a way to filter the noise away from the data so that we have smooth functions to differentiate, but we need to be sure not to change the character of these functions in the process. One simple way to do this is to take our noisy function ϕ and define a new function

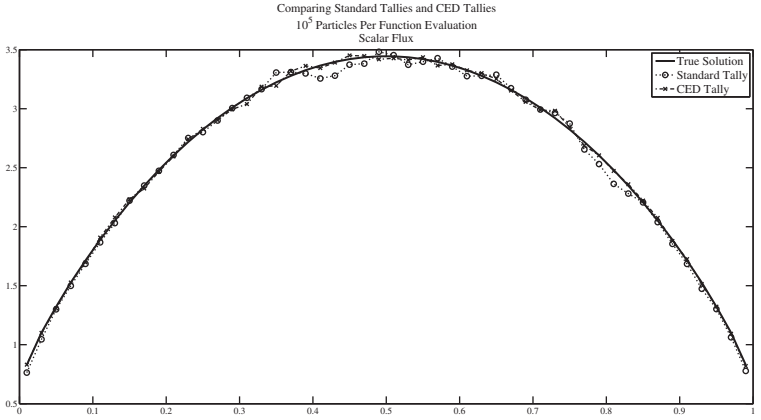

FIG. 4.1. *Noisy scalar fluxes using* 10^5 *particles per function evaluation from Test* 1 *of section* 6.

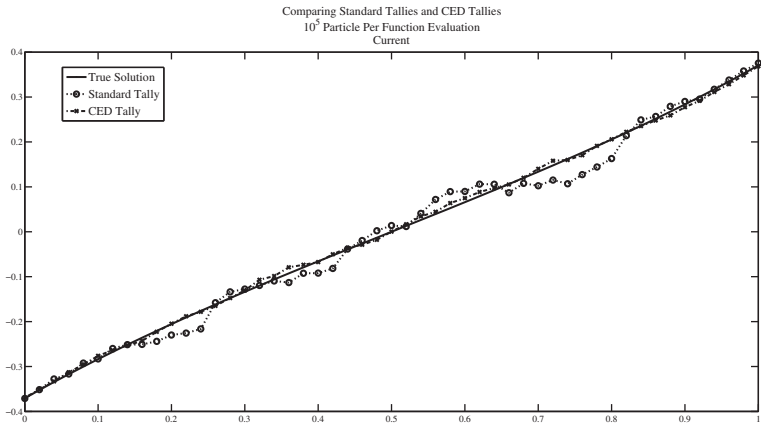

FIG. 4.2. *Noisy currents using* 10^5 *particles per function evaluation from Test* 1 *of section* 6.

ϕ^* by:

$$\phi_i^* = \frac{\phi_{i-1} + 2\phi_i + \phi_{i+1}}{4}. \tag{4.1}$$

This acts as a spatial averaging of the function, but has the potential to change the shape of the function, especially near boundaries or material interfaces. Furthermore, it doesn't truly guarantee that after the application of the filter we have a "smooth" function. Instead, we take a slightly more sophisticated approach. It is important that we do not alter the structure of the solution when attempting to remove noise.

Let us divide our domain into a total of $R + 1$ regions:

$$\{[0, x_1], [x_1, x_2], \ldots, [x_{R-1}, x_R], [x_R, \tau]\}.$$

Now, over each of these regions, we'd like to construct a polynomial of low degree that interpolates the function as closely as possible. We'll denote these degrees d_1, $d_2, \ldots, d_R$, d_{R+1} and we'll also demand that at each of the region boundaries x_i that the spline is continuous and has k_i continuous derivatives. By construction, we must choose $0 \leq k_i \leq \min(d_i, d_{i+1})$. We can perform a least squares fit of this low-degree polynomial spline to create a continuous, differentiable approximation to either the flux or current.

We can smooth the flux and currents from Figures 4.1 and 4.2 by using a least squares interpolation with regions separated by boundary locations $x_1 = .1$, $x_2 = .3$, $x_3 = .7$, and $x_4 = .9$. Over each of the regions we use degree 3 polynomials and demand that the function and its first derivative are continuous. In Figures 4.3 and 4.4 we acknowledge the results of this filter. We compare the MC fluxes and currents to "true" solutions computed using a deterministic solver.

It is important to note that we must use low-degree polynomials for this interpolation. In fact, we would much prefer to use more regions, each equipped with a low-degree polynomial than a few regions with higher degree polynomials. When we begin to use high-degree polynomials, the polynomials tend to adapt to the shape of the noise and this is precisely what we are trying to avoid. By using all low-degree polynomials, our spline is forced to capture the overall shape of the curve and ignore the high-frequency noise.

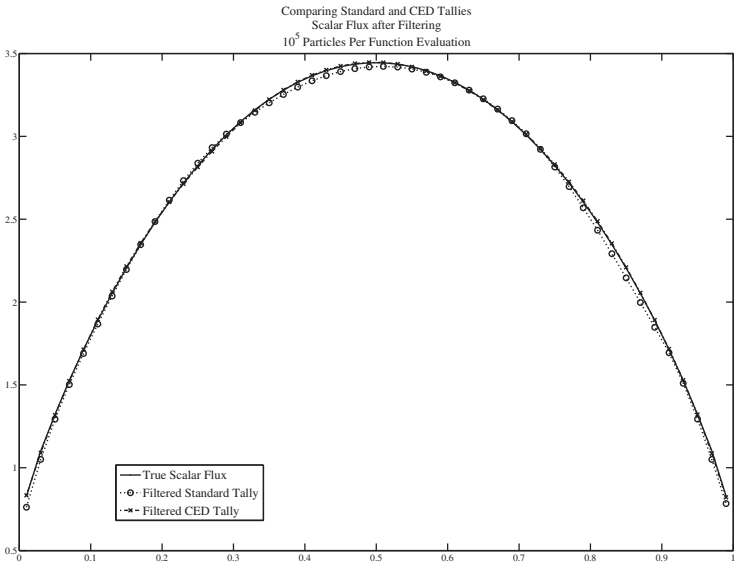

FIG. 4.3. *Standard tallied flux and CED tallied flux after application of the least squares fit filter from Test* 1 *of section* 6.

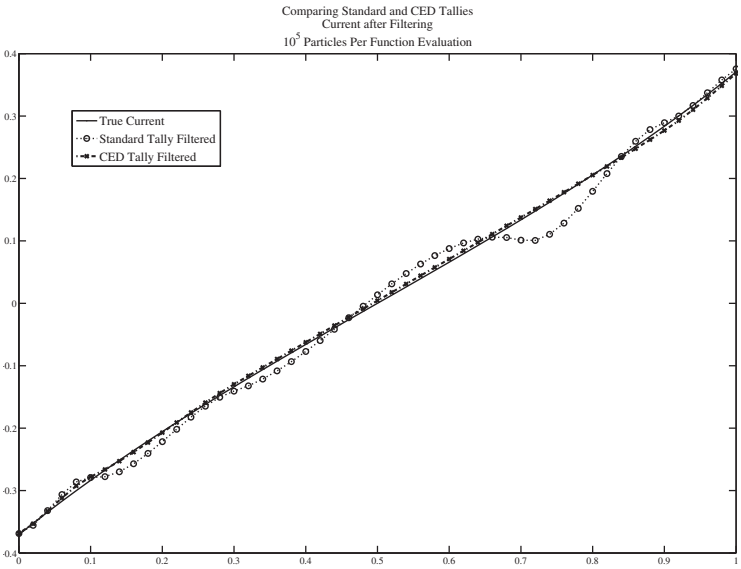

FIG. 4.4. *Standard tallied current and CED tallied current after application of the least squares fit filter from Test* 1 *of section* 6.

We note that it is still beneficial to use as many particle histories as possible, however. Filtering can only work well when the general shape of the flux or current has been well-approximated and the noise is high frequency and relatively small. Reducing the noise in the MC function evaluations by using more particle histories will only provide a better approximation of the desired quantities.

4.1.2. Analytic Jacobian-vector product. Within the Newton–Krylov iteration, we must solve (4.2) for the Newton step, v,

$$F'(\phi)v = -F(\phi), \tag{4.2}$$

where $F'(\phi)$ is the Jacobian. In general, it is too difficult and time consuming to build the Jacobian matrix for the discretized system. Instead, it is standard to compute the Jacobian-vector product using the finite-difference formula

$$F'(\phi)v = \frac{F(\phi + hv) - F(\phi)}{h}. \tag{4.3}$$

Unfortunately, if we choose h too small in relation to the size of the noise in the MC compuation, we may amplify this noise [6]. Furthermore, if we choose h too large, the finite-difference Jacobian-vector product may not be a good approximation of the true Jacobian-vector product. Therefore, we abandon this formula for an analytic formula for the Jacobian-vector product.

Since F is Fréchet differentiable, we can utilize the following formula to compute the analytic Jacobian-vector product:

$$F'(\phi)v = \frac{d}{dh}F(\phi + hv)\bigg|_{h=0}. \tag{4.4}$$

In order to facilitate this calculation, we rewrite $F(\phi)$ using operator notation,

$$F(\phi) = \mathcal{L}\phi + \mathcal{N}\phi,$$

where $\mathcal{L}$ is the linear map,

$$\mathcal{L}\phi = \frac{d}{dx}\left[\frac{-1}{3\Sigma_t}\frac{d\phi}{dx}\right] + (\Sigma_t - \Sigma_s)\phi, \tag{4.5}$$

and $\mathcal{N}$ is the nonlinear map,

$$\mathcal{N}\phi = \frac{d}{dx}\left[\hat{D}(\phi^{HO}, J^{HO})\phi\right] - q. \tag{4.6}$$

Now, we can write the Jacobian-vector product in terms of its linear and nonlinear components

$$F'(\phi)v = \mathcal{L}'(\phi)v + \mathcal{N}'(\phi)v. \tag{4.7}$$

It is clear that $\mathcal{L}'(\phi)v$ is given by the simple expression

$$\mathcal{L}'(\phi)v = \mathcal{L}v. \tag{4.8}$$

The expression for $N'(\phi)v$ is somewhat more complicated. Using the formula (4.4) and properties of differentiation, it is easy to show that

$$N'(\phi)v = \frac{d}{dx}\left[\hat{D}'(\phi)v\phi + \hat{D}(\phi)v\right], \tag{4.9}$$

where we have expressed $\hat{D}(\phi^{HO}, J^{HO})$ as $\hat{D}(\phi)$ to make the dependence of $\hat{D}$ on ϕ explicit.

Now, we can complete the formula by finding an expression for $\hat{D}'(\phi)v$. We accomplish this using the same technique we used to derive the expression for the nonlinear term. We find that

$$\hat{D}'(\phi)v = \frac{\phi^{HO}\left[J'^{HO}(\phi)v + \frac{1}{3\Sigma_t}\frac{d\phi'^{HO}(\phi)v}{dx}\right] - \left[J^{HO}(\phi) + \frac{1}{3\Sigma_t}\frac{d\phi^{HO}(\phi)}{dx}\right]\phi'^{HO}(\phi)v}{(\phi^{HO}(\phi))^2}, \tag{4.10}$$

where

$$\phi'^{HO}(\phi)v = \int_{-1}^{1} \xi(x,\mu)d\mu, \tag{4.11}$$

$$J'^{HO}(\phi)v = \int_{-1}^{1} \mu\xi(x,\mu)d\mu. \tag{4.12}$$

Here, $\xi(x,\mu)$ is the solution to the fixed-source scattering-free transport equation where the scalar flux is taken to be $v(x)$,

$$\mu\frac{\partial\xi(x,\mu)}{\partial x} + \Sigma_t\xi(x,\mu) = \frac{1}{2}\Sigma_s v(x). \tag{4.13}$$

By combining (4.7)–(4.12) we have an analytic formula for the Jacobian-vector product.

It should be clear that much like the finite-difference approximation to the Jacobian-vector product, we can compute the analytic Jacobian-vector product with only a single additional transport sweep. This means that, at no added cost to us, we have a formula for a more accurate representation of the Jacobian-vector product.

In practice, we find that when the transport sweep is computed deterministically and the mesh is relatively fine, the analytic and finite-difference Jacobian-vector product agree to several decimal places. However, when the MC transport sweep is used, these two quantities can differ by orders of magnitude when N_{MC} is small, and the finite-difference Jacobian is too inaccurate to be useful.

5. Algorithms. We will express the nonlinear equation from NDA as

$$F(u) = 0 \tag{5.1}$$

on R^N.

We will solve (5.1) with a JFNK algorithm [2, 10]. In these methods the Newton direction is computed by an inner Krylov iteration and the Jacobian vector required by the Krylov method is either approximated with a finite difference or computed analytically. These methods avoid the computation and storage of the Jacobian matrix itself. In this paper we will focus on the Newton-GMRES [6, 8] algorithm. The algorithmic description below is a formal description of a Newton-GMRES iteration. This algorithm includes an Armijo line search, which is necessary to obtain convergence from initial iterates far from a solution [6, 8]. Note that the Newton-GMRES algorithm in [9] did not use a line search and the computations reported in that paper used a diffusion solution as the initial iterate.

The inputs for the algorithm are the nonlinear function F, the initial iterate u, relative and absolute nonlinear error tolerances τ_r and τ_a, and a linear residual tolerance η. The algorithmic parameter α is usually [6] set to 10^{-4}.

The local theory (i.e., when $\lambda = 1$ and the Newton step $s = \lambda d = d$) from [6] not only predicts the convergence speed, but also accounts for errors in the Jacobian and

NEWTON-GMRES $(F, u, \tau_r, \tau_a, \eta)$.
  evaluate $F(u)$; $\tau \leftarrow \tau_r \|F(u)\| + \tau_a$.
  while $\|F(u)\| > \tau$ **do**
    Use GMRES to find d such that $\|F'(u)d + F(u)\| \leq \eta \|F(u)\|$
    If no such d can be found after $I_{\max}$ iterations, terminate with failure.
    $\lambda = 1$
    while $\|F(u + \lambda d)\| > (1 - \alpha\lambda)\|F(u)\|$ **do**
      $\lambda \leftarrow \lambda/2$
      If $\lambda < \lambda_{\min}$, terminate with failure.
    end while
    $u \leftarrow u + \lambda d$
  end while

the function evaluation. The local theory depends on the standard assumptions for nonlinear equations These assumptions are the following.

Assumption 5.1.

1. Equation (5.1) has a solution u^*.
2. $F' : \Omega \to R^{N \times N}$ is Lipschitz continuous with Lipschitz constant γ.
3. $F'(u^*)$ is nonsingular.

Now assume that at a current iteration u_c we approximate Jacobian-vector products $F'(u_c)w$ by $J_c w$ and that, rather than evaluate $F(u_c)$ exactly, we evaluate $\bar{F}_c$. The local theory then says that if the linear solver terminates when

$$\|J_c s + \bar{F}_c\| \leq \eta \|\bar{F}_c\|,$$

where

$$\|J_c - F'(u_c)\| = \Delta_c \text{ and } \|\bar{F}_c - F(u_c)\| = \epsilon_c,$$

and we let the new iterate $u_+ = u_c + s$, then

$$\|e_+\| = O(\|e_c\|^2 + (\Delta_c + \eta)\|e_c\| + \epsilon_c), \tag{5.2}$$

where $e = u - u^*$ is the error.

The theory when the current iterate is far from a solution is more subtle, but for the present paper we need only note that if the Jacobian is highly accurate, then the iteration $\{u_n\}$ will either converge to a solution u^* which satisfies the standard assumptions or fail to converge in one of two ways. We base our approach on the convergence theorem.

5.1. MC formulation and algorithm proposal. Now, consider the case where the internal mapping G is approximated by an MC simulation for which we can control the number of realizations N_{MC}. The standard deviation is $O(1/\sqrt{N_{MC}})$, but we do not assume we know what the constant in the O-term is.

The MC approximation of G results in an error in the evaluation of both the evaluation of $F(u)$ and the Jacobian-vector product $F'(u)w$. Our current approach is to use the Newton-GMRES direction until the line search fails, and then

- increase N_{MC} by a factor of 100 and
- restart the Newton-GMRES iteration from the previous iteration.

We must establish some notation before we formally describe the algorithm. We approximate $F(u)$ with an MC simulation to obtain $F(u, N_{MC})$ and the Jacobian-vector product $F'(u)w$ with an MC simulation with output $DF(u, w, N_{MC})$. The

evaluations of F we need for the Newton step and testing termination tolerances for the nonlinear iteration, or within the line search in the Newton-GMRES code, must be replaced with calls to the MC simulation. Similarly the Jacobian-vector products within the GMRES iteration are MC simulations. In addition to the inputs of Algorithm Newton-GMRES, the new algorithm requires an upper limit $I_{\max}$ on the number of linear iterations, a lower bound $\lambda_{\min}$ on the step size, and an initial value N_{MC} for the number of trials in the MC simulation.

NEWTON-GMRES-MC $(F, u, \tau_r, \tau_a, \eta, I_{\max}, \lambda_{\min}, N_{MC})$.

Evaluate $R_{MC} = F(u, N_{MC})$; $\tau \leftarrow \tau_r \|R_{MC}\| + \tau_a$.
while $\|R_{MC}\| > \tau$ **do**
  Use GMRES with a limit of $I_{\max}$ iterations to find d such that $\|DF(u, d, N_{MC}) + R_{MC}\| \leq \eta \|R_{MC}\|$
  if the GMRES iteration fails **then**
    $N_{MC} \leftarrow 100 * N_{MC}$
    Evaluate $R_{MC} = F(u, N_{MC})$
  else
    $\lambda = 1$
    Evaluate $R_{Trial} = F(u + \lambda d, N_{MC})$
    while $\|R_{Trial}\| > (1 - \alpha\lambda)\|R_{MC}\|$ and $\lambda \geq \lambda_{\min}$ **do**
      $\lambda \leftarrow \lambda/2$
      Evaluate $R_{Trial} = F(u + \lambda d, N_{MC})$
    end while
    if $\lambda \geq \lambda_{\min}$ **then**
      $u \leftarrow u + \lambda d$
      $R_{MC} = R_{Trial}$
    else
      $N_{MC} \leftarrow 100 * N_{MC}$
      Evaluate $R_{MC} = F(u, N_{MC})$
    end if
  end if
end while

This is a simple modification of the Newton-GMRES algorithm and seems to work well.

6. Results. We present results here for JFNK-NDA(MC). We'll consider three test problems that demonstrate the convergence properties. The first two test problems simulate a highly scattering material ($\Sigma_s/\Sigma_t = .99$) and the third test problem represents a system with a much more moderate scattering ratio ($\Sigma_s/\Sigma_t = .75$). In all test problems, both the flux and current are filtered using a least squares interpolant filter in both the function and Jacobian evaluation. Furthermore, during the function evaluation, $\hat{D}$ is filtered using the same least squares interpolant filter.

In each of the following tests, we use $\phi = 0$ as the initial iterate and precondition the linear solver using the same preconditioner as described in [9]. We choose to use the inverse of the operator M, defined by

$$M = \frac{d}{dx}\left[-\frac{1}{3\Sigma_t}\frac{d}{dx} + \hat{D}^k\right] + (\Sigma_t - \Sigma_s)\,,$$

where $\hat{D}^k$ was computed during the most recent evaluation of F.

6.1. Test 1. For the first test problem, we'll use the following parameters:

Parameter	Value
Σ_t	10
Σ_s	9.9
τ	1
q	.5
Spatial cells	50

In Figure 6.1, we see that using 10^6 particles (Run 1) per function evaluation (using the standard tally) does not provide enough accuracy to converge to the desired tolerance. However, using 10^6 particles per function evaluation does allow us to get a rough shape for the scalar flux. We use this estimate of the flux as the starting point for a new run of JFNK-NDA(MC) using 10^8 particles per function evaluation (Run 2). This provides yet a better estimate for a third pass of the algorithm.

In Figure 6.2, we see that using 10^6 particles per function evaluation provides an order of magnitude decrease in the nonlinear residual from the initial iterate for both standard and CED tallies. Increasing the number of particles by a factor of 100 allows for another order of magnitude decrease. The final run, using 10^{10} particles per function evaluation drops the norm of the nonlinear residual below 5×10^{-3}.

Comparing the standard and CED tallies, we see that the CED tallies are more effective at lower particle counts. By the end of Run 2, the CED tallies had decreased the nonlinear residual by an extra factor of two over the standard tallies and had done so using few function evaluations. At the level of 10^{10} particles per function evaluation, the two tallying methods become indistinguishable.

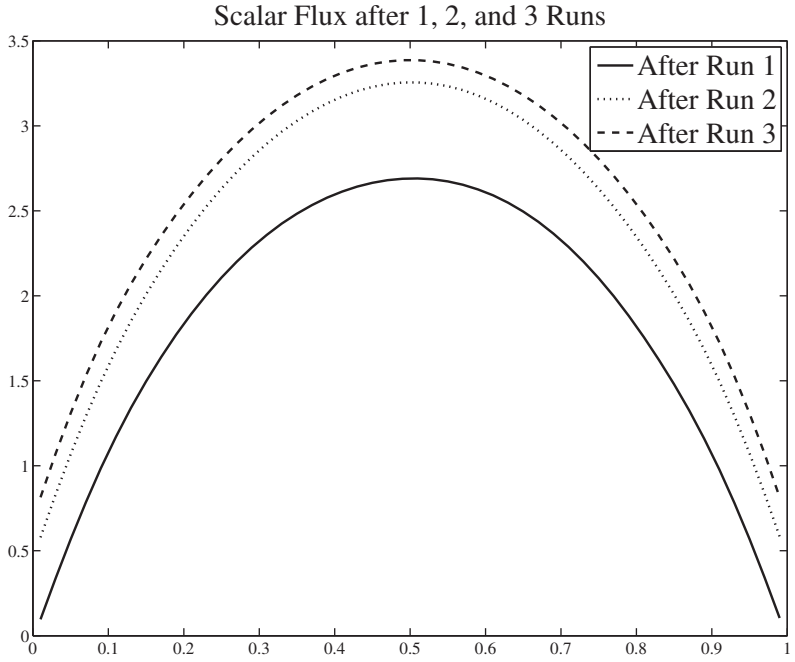

FIG. 6.1. *Fluxes after* $1, 2,$ *and* 3 *runs of JFNK-NDA(MC) with increasing particle counts using standard tallies for Test* 1.

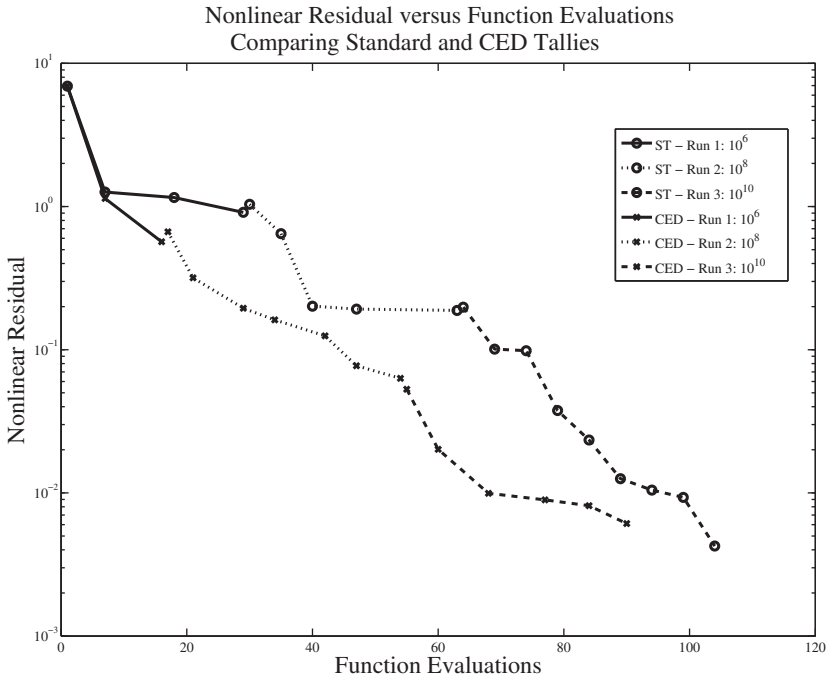

FIG. 6.2. *Nonlinear residuals after* 1, 2, *and* 3 *runs of JFNK-NDA(MC) with increasing particle counts using both standard and CED tallies for Test* 1.

It should be clear from Figure 6.2 that the accuracy of the function evaluation is intimately tied to the ability of the algorithm to decrease the nonlinear residual. As a second example of this, let us consider a new simulation in which we allow for four runs, one run using each of 10^5, 10^7, 10^9, and 10^{11} particles per function evaluation. We see the same convergence behavior in Figure 6.3 using four runs that we saw in Figure 6.2.

6.2. Test 2. For the second test problem, we'll use the following parameters:

Parameter	Value
Σ_t	10
Σ_s	9.9
τ	10
q	.5
Spatial cells	50

In Figure 6.4, we can see that even after one run using 10^6 particles per function evaluation and standard tallies, we have found a very good approximation to the scalar flux. Contrast this with the first test problem (Figure 6.1) in which we needed to increase particles on two occasions before we had an accurate representation of the scalar flux. Minor improvements are seen by increasing the number of particles per function evaluation in the second test, but we may find that for some applications the second or third run of JFNK-NDA(MC) was unnecessary.

The convergence properties for this test problem can be seen in Figure 6.5. Again, we can decrease the nonlinear residual by an order of magnitude with each run of JFNK-NDA(MC). In this case, both tallies perform more efficiently. Both tallies

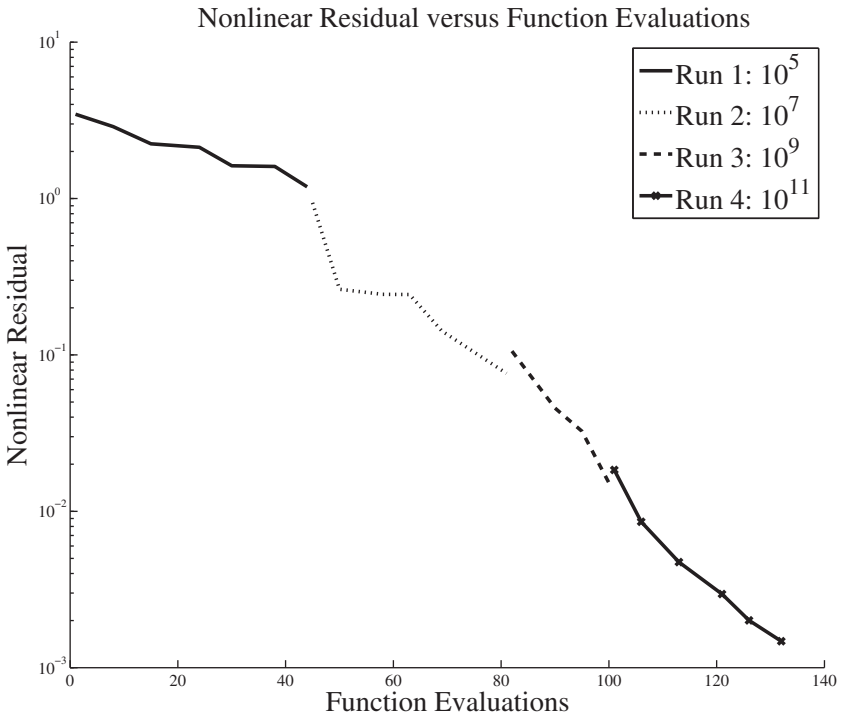

FIG. 6.3. *Nonlinear residuals after* 1, 2, 3, *and* 4 *runs of JFNK-NDA(MC) with increasing particle counts using both standard tallies for Test* 1.

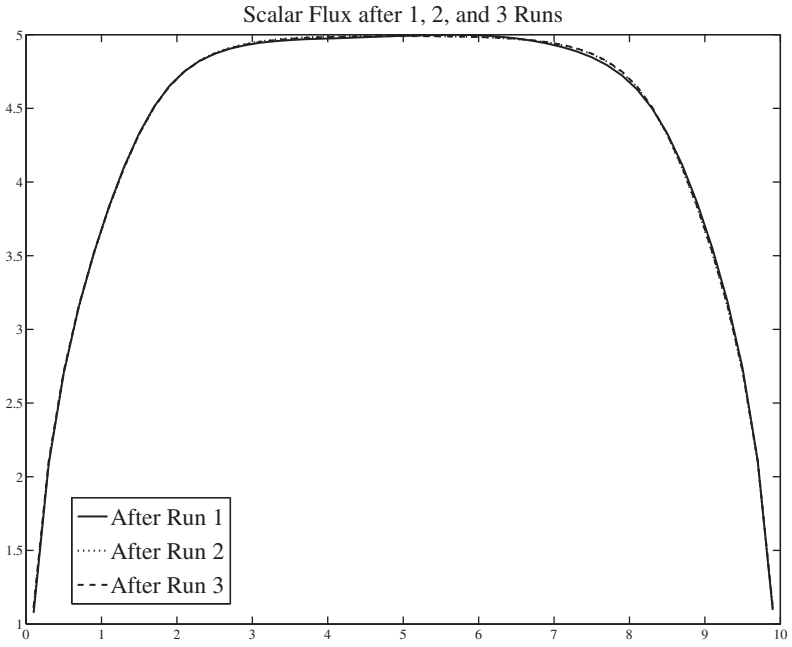

FIG. 6.4. *Fluxes after* 1, 2, *and* 3 *runs of JFNK-NDA(MC) with increasing particle counts using standard tallies for Test* 2.

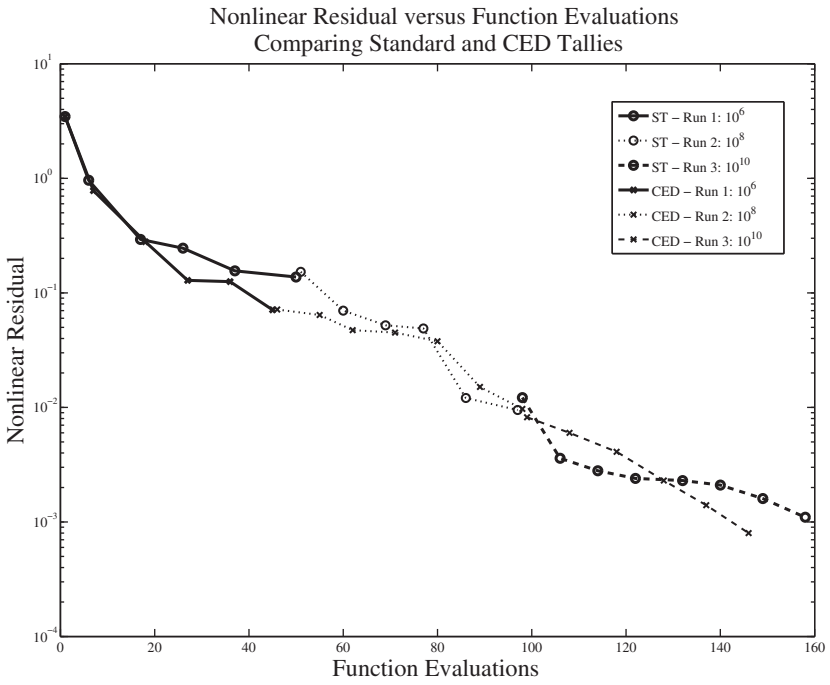

FIG. 6.5. *Nonlinear residuals after* 1, 2, *and* 3 *runs of JFNK-NDA(MC) with increasing particle counts using both standard and CED tallies for Test* 2.

allow us to drive the nonlinear residual down to roughly 10^{-1} using only 10^6 particles per function evaluation. For this problem, the two tallies perform almost exactly the same. The benefits of the CED tally are less noticeable for Test 2. Both tallies allow us to converge the nonlinear residual to 10^{-3}.

We repeat the simulation for Test 2 using a total of four runs beginning at 10^5 particles per function evaluation and finishing with 10^{11} particles per function evaluation. We run this test as a further demonstration that increasing the number of particles per function evaluation by a factor of 100 increases the accuracy of the MC transport sweep by a factor of 10, and ultimately allows us to decrease the residual by a factor of 10. This is demonstrated in Figure 6.6.

6.3. Test 3. For the final test problem, we'll use the following parameters:

Parameter	Value
Σ_t	10
Σ_s	7.5
τ	10
q	.5
Spatial cells	50

This test simulates a medium with a lower scattering ratio to demonstrate the effectiveness of this algorithm across multiple scattering regimes. In Figure 6.7 we can see that for lower scattering ratios, we can lower the nonlinear residual to even tighter tolerances using few total particles. This confirms that our algorithm works across different scattering regimes. Furthermore, the MC transport sweep is more accurate for low scattering environments. Much like Tests 1 and 2, we see that CED tallies marginally improve accuracy, but do not have a profound effect on convergence.

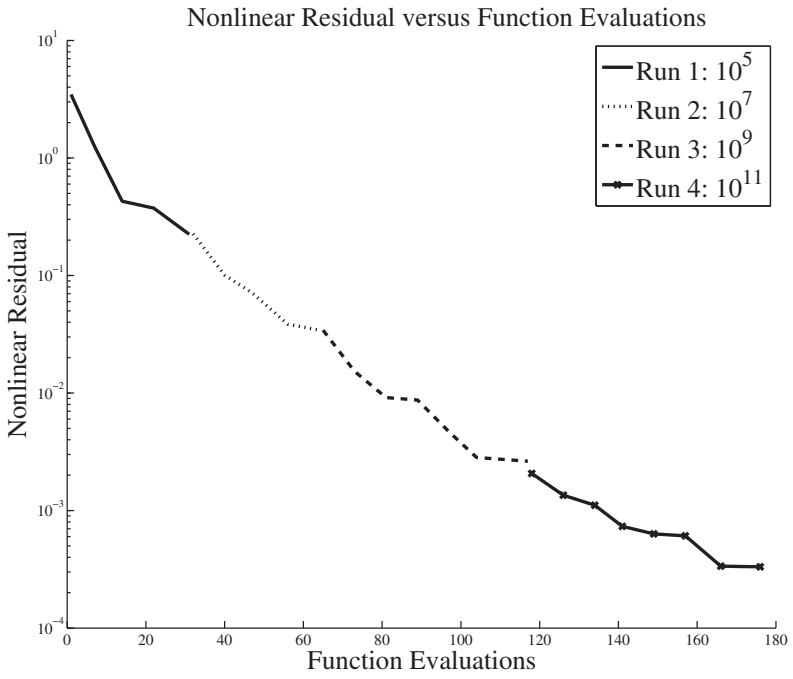

FIG. 6.6. *Nonlinear residuals after* 1, 2, 3, *and* 4 *runs of JFNK-NDA(MC) for Test* 2. *We see that each time the number of particles per function evaluation is increased by a factor of* 100, *we gain another order of magnitude decrease in the nonlinear residual.*

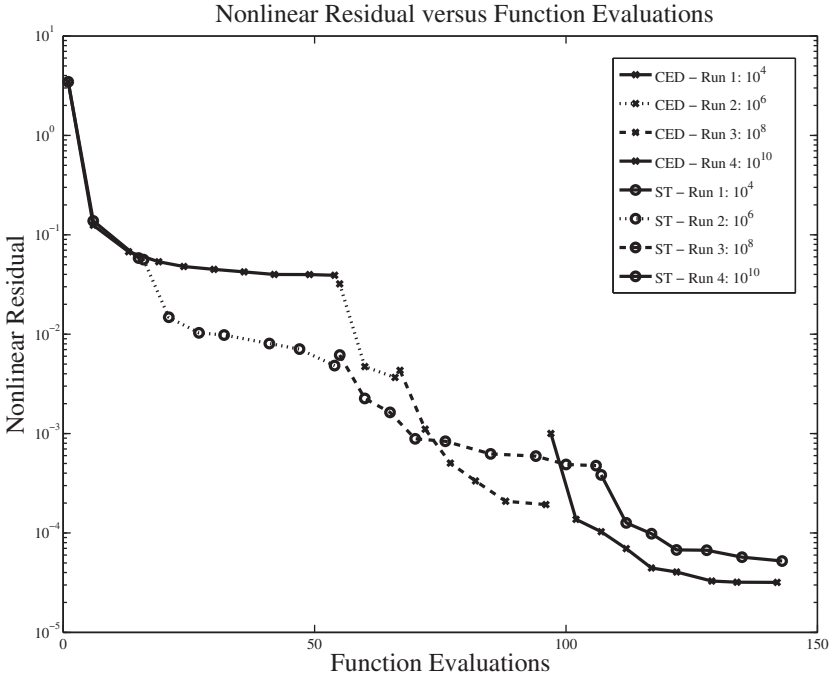

FIG. 6.7. *Nonlinear residuals after* 1, 2, 3, *and* 4 *runs of JFNK-NDA(MC) for Test* 2. *We see that each time the number of particles per function evaluation is increased by a factor of* 100, *we gain another order of magnitude decrease in the nonlinear residual.*

7. Conclusions. We have implemented JFNK-NDA(MC), a new hybrid algorithm for neutronics. We solve the HO problem with MC using CED tallies and the LO problem with a JFNK algorithm, in this case Newton-GMRES. We have discussed the problems in the implementation and our solutions to those problems and presented numerical results.

In the future we plan to do a detailed analysis of the convergence properties of the Newton-MC algorithm we proposed in this paper and apply that algorithm to k-eigenvalue problems in neutronics.

REFERENCES

[1] R. E. Alcouffe, *Diffusion synthetic acceleration methods for the diamond differenced discrete ordinates equations*, Nuclear Sci. Engrg., 66 (1977), pp. 344–355.

[2] P. N. Brown and Y. Saad, *Hybrid Krylov methods for nonlinear systems of equations*, SIAM J. Sci. Stat. Comput., 11 (1990), pp. 450–481.

[3] S. Chandrasekhar, *Radiative Transfer*, Dover, New York, 1960.

[4] J. Fleck and J. Cummings, *Implicit Monte Carlo scheme for calculating time and frequency dependent nonlinear radiation transport*, J. Comput. Phys., 8 (1971), pp. 313–342.

[5] V. Y. Gol'din, *A quasi-diffusion method for solving the kinetic equation*, USSR Comp. Math. Math. Phys, 4 (1967), pp. 136–149.

[6] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, Frontiers in Applied Mathematics 16, SIAM, Philadelphia, 1995.

[7] C. T. Kelley, *Multilevel source iteration accelerators for the linear transport equation in slab geometry*, Transport Theory Statist. Phys., 24 (1995), pp. 679–708.

[8] C. T. Kelley, *Solving Nonlinear Equations with Newton's Method*, Fundamentals of Algorithms 1, SIAM, Philadelphia, 2003.

[9] D. Knoll, H. Park, and K. Smith, *Application of the Jacobian-free Newton-Krylov method to nonlinear acceleration of transport source iteration in slab geometry*, Nuclear Sci. Engrg., 167 (2011), pp. 122–132.

[10] D. A. Knoll and D. E. Keyes, *Jacobian-free Newton Krylov methods: A survey of approaches and appliciations*, J. Comput. Phys., 193 (2004), pp. 357–397.

[11] P. J. Lanzdron, D. J. Rose, and J. T. Wilkes, *An analysis of approximate nonlinear elimination*, SIAM J. Sci. Comput., 17 (1996), pp. 538–559.

[12] E. Larsen and J. Yang, *A functional Monte Carlo method for k-eigenvalue problems*, Nuclear Sci. Engrg., 159 (2008), pp. 107–126.

[13] M. J. Lee, H. G. Joo, D. J. Lee, and K. S. Smith, *Investigation of CMFD accelerated Monte Carlo eigenvalue calculation with simplified low dimensional multigroup formulation*, in Proceedings of 2010 International Conference on the Physics of Reactors (PHYSOR 2010), Pittsburgh, PA, 2010.

[14] M. J. Lee, H. G. Joo, D. J. Lee, and K. S. Smith, *Multigroup Monte Carlo reactor calculation with coarse mesh finite difference formulation for real variance reduction*, in Proceedings of the 2010 Joint International Conference on Supercomputing in Nuclear Applications and Monte Carlo (SNA+MC 2010), Tokyo, Japan, 2010.

[15] E. E. Lewis and W. F. Miller, *Computational Methods of Neutron Transport*, American Nuclear Society, La Grange Park, IL, 1993.

[16] M. M. Miften and E. W. Larsen, *The quasi-diffusion method for solving transport problems in planar and spherical geometries*, Transport Theory Statist. Phys., 22 (1993), pp. 165–186.

[17] J. E. Morel, *Basic Krylov methods with application to transport*, in Mathematics and Computation, Supercomputing, Reactor Physics and Nuclear and Biological Applications, Palais des Papes, Avignon, France, 2005, American Nuclear Society.

[18] Y. Saad and M. H. Schultz, *GMRES a generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 856–869.

[19] K. S. Smith and J. D. Rhodes III, *Full-core 2-D LWR core calculation with CASMO-4E*, in Proceedings of the International Conference on New Frontiers of Nuclear Technology: Reactor Physics, Safety and High-Performance Computing (PHYSOR 2002), Seoul, Korea, American Nuclear Society, 2002.

[20] J. S. Warsa, T. A. Wareing, and J. E. Morel, *Krylov iterative methods and the degraded effectiveness of diffusion synthetic acceleration for multidimensional s_n calculations in problems with material discontinuities*, Nuclear Sci. Engrg., 147 (2004), pp. 218–248.

[21] E. R. Wolters, E. W. Larsen, and W. R. Martin, *Generalized hybrid Monte Carlo-CMFD methods for fission source convergence*, in Proceedings of the International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2011), Rio de Janeiro, RJ, Brazil, American Nuclear Society, 2011.

[22] D. Young, W. Huffman, R. Melvin, C. Hilmes, and F. Johnson, *Nonlinear elimination in aerodynamic analysis and design optimization*, in Large-Scale PDE-Constrained Optimization, Lect. Notes Comput. Sci. Eng. 20, Springer-Verlag Berlin, 2003, pp. 17–43.