Journal of Computational Physics 273 (2014) 706–719

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

S_2SA preconditioning for the S_n equations with strictly nonnegative spatial discretization

Don E. Bruss, Jim E. Morel [*], Jean C. Ragusa

Department of Nuclear Engineering, Texas A&M University, College Station, TX 77843, USA

ARTICLE INFO

Article history:
Received 19 December 2013
Received in revised form 14 May 2014
Accepted 16 May 2014
Available online 27 May 2014

Keywords:
Strictly nonnegative closure
Discrete ordinates method
Radiation transport
Discontinuous finite elements
Diffusion synthetic acceleration
Jacobian-free Newton Krylov
Matrix-free Picard Krylov
Nonlinear Krylov acceleration

ABSTRACT

Preconditioners based upon transport sweeps and diffusion-synthetic acceleration have been constructed and applied to the zeroth and first spatial moments of the 1-D S_n transport equation using a strictly nonnegative nonlinear spatial closure. Linear and nonlinear preconditioners have been derived and analyzed. The effectiveness of various combinations of these preconditioners are compared using the source iteration, matrix-free Picard Krylov, and nonlinear Krylov acceleration methods. In one dimension, preconditioning with a linear S_2SA diffusion equation is found to be essentially equivalent to using a nonlinear diffusion equation. The ability to use a linear diffusion equation has important implications for preconditioning the S_n equations with a strictly nonnegative spatial discretization in multiple dimensions.

1. Introduction

Non-physical solutions containing negative angular fluxes can be obtained in 1-D radiation transport problems containing optically thick cells and in multidimensional problems under a wide variety of conditions. Radiative transfer problems can be especially prone to non-physical negativities as these problems often include regions that are extremely optically thick. A strictly nonnegative spatial discretization for the S_n neutron transport equation in both 1-D and 2-D Cartesian geometries was recently developed by Maginot, Morel, and Ragusa [1,2]. Their nonlinear Consistent Set-to-Zero (CSZ) method exactly solves the zeroth and first spatial moments of the transport equation and reduces to the linear discontinuous Galerkin (LD) method when the LD solution yields a nonnegative solution. As such it can be considered a nonlinear fixup to the LD method. However, unlike standard fixup schemes, it solves both moment equations under all conditions rather than solving only the balance equation when fixup is required. Consequently, it is perhaps best considered as a strictly nonnegative generalization of the LD method. Maginot et al. demonstrated that the CSZ method is a much more accurate alternative to existing fixup schemes for the LD equations as well as existing strictly positive moment-preserving schemes such as the exponential discontinuous method.

In [1] and [2], the CSZ equations were solved using source iteration (SI). SI algorithms suffer from arbitrarily slow convergence in diffusive problems, limiting the applicability of the method. The purpose of this work is twofold. First, we investigate 1-D preconditioning techniques that enable the CSZ equations to be efficiently solved in diffusive problems; and second, we apply a matrix-free Picard Krylov (MFPK) method derived from the Jacobian-free Newton Krylov (JFNK)

[*] Corresponding author.
E-mail addresses: brussd@tamu.edu (D.E. Bruss), morel@tamu.edu (J.E. Morel), jean.ragusa@tamu.edu (J.C. Ragusa).

http://dx.doi.org/10.1016/j.jcp.2014.05.022
0021-9991/© 2014 Elsevier Inc. All rights reserved.

method and the nonlinear Krylov acceleration (NKA) method to the nonlinear system to investigate the effectiveness of the preconditioners when used in conjunction with these methods.

Two distinct preconditioners based on the S_2 synthetic acceleration (S_2SA) method are formulated as preconditioners [3]. The first preconditioner is based upon an LD discretization of the S_2 equations. This approach yields a linear preconditioner but is inconsistent with the CSZ spatial discretization. The second preconditioner uses the S_2 equations discretized with the CSZ method and is therefore consistent but nonlinear. In 1-D slab geometries, analytic S_2SA is equivalent to analytic diffusion-synthetic acceleration (DSA) [4]. DSA preconditioning was developed to overcome the arbitrarily slow convergence of SI schemes in diffusive problems with a scattering ratio c close to one. The application of DSA enables solutions to problems with highly diffusive regions to be efficiently computed. It is non-trivial to derive a diffusion equation that is consistent with the CSZ-discretized equations, especially in two and three spatial dimensions. It would be very advantageous if the inconsistent but linear DSA preconditioner proved effective for multidimensional problems. Here we address their effectiveness for 1-D problems.

The JFNK method directly applies Newton's method to a problem. Any linear fixed-point iteration scheme, such as SI, can be recast as a preconditioner and the preconditioned equations solved via a Krylov method. This is often referred to as "wrapping" an iteration scheme in a Krylov method [5]. Wrapping a fixed-point iteration scheme in a Krylov method can usually be expected to be more efficient than the fixed-point iteration scheme alone [6]. Furthermore, stability is ensured even when the fixed-point iteration scheme is itself unstable [5]. Thus there is great advantage to wrapping DSA in a Krylov method.

The central theme of the JFNK method is to avoid forming the Jacobian matrix since only the action of the Jacobian matrix upon a vector is needed [7]. This action is approximated with finite difference and requires one residual evaluation per Krylov iteration. The CSZ equations were not well suited to this approximation as the finite difference approximation is challenging to apply when the angular flux spans many orders of magnitude. A matrix-free Picard Krylov (MFPK) method derived from the JFNK method is introduced here and applied to the problem. The residual is preconditioned with the sweep and S_2SA.

The NKA method is a Krylov acceleration technique for fixed-point iterative methods. Fixed-point methods of a form associated with the S_n equations can be thought of as an approximate Newton iterative method in which the Jacobian is approximated as the identity. The NKA method uses information derived from successive fixed-point iterations to approximate the action of the Jacobian [8,9]. Thus the NKA method approximates Newton's method and may significantly accelerate the convergence of a fixed-point iterative method.

The full angular flux vector is required to solve the CSZ moment equations. The Krylov subspaces built by the JFNK and MFPK methods are populated with vectors the same length as the full angular flux solution. The NKA method requires storing two vectors of this size for each entry in the NKA subspace. The storage costs associated with the CSZ method are large, but this may simply be the cost of achieving a universally efficient solution technique for a strictly positive generalization of the LD method.

In the following sections, we derive the spatial moment equations discretized with the LD and CSZ methods. Preconditioners based on source iteration schemes are derived using LD and CSZ discretizations, yielding linear and nonlinear preconditioning schemes, respectively. The NKA, JFNK, and MFPK methods are outlined. Finally, computational results are presented for a challenging test problem.

2. The transport equation

The one-speed steady-state neutron transport equation in 1-D Cartesian geometry, assuming isotropic scattering and an isotropic external source, is

$$\mu \frac{\partial}{\partial x}\psi(x,\mu) + \sigma_t(x)\psi(x,\mu) = \frac{\sigma_s(x)}{4\pi}\phi(x) + q(x,\mu), \tag{1}$$

where ψ is the angular flux (p/cm^2 s ster), μ is the directional cosine, σ_t (cm^{-1}) and σ_s (cm^{-1}) are the macroscopic total and scattering cross sections, respectively, and q (p/cm^3 s str) an inhomogeneous source.

The transport equation is discretized in angle with the discrete ordinates (S_n) approximation. This corresponds to collocation at a set of quadrature points $\{\mu_d\}_{d=1}^n$ with weights $\{w_d\}_{d=1}^n$. The angle-integrated scalar flux ϕ is approximated as

$$\phi(x) = \sum_{d=1}^{n} w_d \psi_d(x), \tag{2}$$

where

$$\psi_d(x) = \psi(x, \mu_d). \tag{3}$$

The zeroth and first spatial moments of the transport equation are generated by multiplying Eq. (1) by basis functions $B_0(x) = \frac{1}{h_i}$ and $B_1(x) = \frac{6}{h_i^2}(x - x_i)$ and integrating the resultant equations over cell i, centered at x_i, with edges at $x_{i\pm\frac{1}{2}}$, and width h_i. This yields the following respective equations:

$$\frac{\mu_d}{h_i}(\psi_{d,i+\frac{1}{2}} - \psi_{d,i-\frac{1}{2}}) + \sigma_{t,i}\psi_{A,d,i} = \frac{\sigma_{s,i}}{4\pi}\phi_{A,i} + q_{A,i,d}, \tag{4a}$$

$$\frac{3\mu_d}{h_i}(\psi_{d,i+\frac{1}{2}} - 2\psi_{A,d,i} + \psi_{d,i-\frac{1}{2}}) + \sigma_{t,i}\psi_{X,d,i} = \frac{\sigma_s}{4\pi}\phi_{X,i} + q_{X,i,d}, \tag{4b}$$

where the average angular flux $\psi_{A,d}$ and the slope of the angular flux $\psi_{X,d}$ in cell i are respectively defined by:

$$\psi_{A,d,i} = \frac{1}{h_i}\int\limits_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \psi_d(x)dx, \tag{5a}$$

and

$$\psi_{X,d,i} = \frac{6}{(h_i)^2}\int\limits_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \psi_d(x)(x - x_i)dx. \tag{5b}$$

The moment equations (4) for all cells and all directions can be written succinctly in operator form as

$$\mathbf{L}\psi = \frac{\sigma_s}{4\pi}\mathbf{P}\psi + q. \tag{6}$$

Here $\mathbf{L}$ includes the streaming and interaction terms, $\mathbf{P}$ is the angle integration operator, and ψ includes ψ_A and ψ_X in all directions and cells. The two-moment equations in (4) include three unknowns: $\psi_{A,d,i}$, $\psi_{X,d,i}$, and the outflow, $\psi_{d,i\pm\frac{1}{2}}$ (depending upon the sign of μ). A third equation is required to close this set of equations. Most closures relate the outflow to the average and slope of the flux within the cell.

2.1. Linear discontinuous method

The linear discontinuous Galerkin (LD) method is perhaps the most widely used closure. The discontinuous Galerkin method was first proposed for the neutron transport equation in [10]. The LD method assumes a linear flux distribution in every cell:

$$\psi(x)_{LD} = a_{LD} + \frac{2}{h_i}(x - x_i)b_{LD}. \tag{7}$$

The values of a_{LD} and b_{LD} are the primary unknowns that must be determined in each cell and direction.

The value of the angular flux ψ at the interfaces between cells is defined in the following manner. In the positive directions ($\mu_d > 0$), Eq. (7) applies for $x \in (x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$ and the incoming flux is upwinded from the previous cell,

$$\psi(x_{i-\frac{1}{2}}) = \lim_{\epsilon \to 0+} \psi(x_{i-\frac{1}{2}} - \epsilon). \tag{8a}$$

In the negative directions ($\mu_d < 0$), Eq. (7) is valid for $x \in [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}})$, with the value of the angular flux at the right face of the cell upwinded from the next cell,

$$\psi(x_{i+\frac{1}{2}}) = \lim_{\epsilon \to 0+} \psi(x_{i+\frac{1}{2}} + \epsilon). \tag{8b}$$

The definitions of ψ_A and ψ_X given in (5) yield $\psi_A = a_{LD}$, $\psi_X = b_{LD}$, $\psi_{i+\frac{1}{2}} = a_{LD} + b_{LD}$ when $\mu > 0$, and $\psi_{i-\frac{1}{2}} = a_{LD} - b_{LD}$ when $\mu < 0$. With these definitions, the moment equations (4) can be written entirely in terms of a_{LD} and b_{LD}. We denote the LD-discretized operators with a “hat,”

$$\widehat{\mathbf{L}}\psi = \frac{\sigma_s}{4\pi}\widehat{\mathbf{P}}\psi + q. \tag{9}$$

The vector ψ contains the unknowns a_{LD} and b_{LD} (which are equivalent to ψ_A and ψ_X) for every cell and direction.

2.2. Consistent set-to-zero method

The CSZ closure modifies the LD closure for cases where LD yields negative outflows or a negative flux within the cell. If the LD method yields a positive solution at all points within a cell, then the CSZ method is identical to the LD method in that cell [1,2].

Even though the CSZ solution is not necessarily linear, there is always a linear distribution $\widetilde{\psi}_{CSZ}$ associated with the CSZ solution within a portion of the cell,

$$\widetilde{\psi}_{CSZ}(x) = a_{CSZ} + \frac{2}{h_i}(x - x_i) b_{CSZ}. \tag{10}$$

The CSZ angular flux, ψ_{CSZ}, is defined as equal to $\widetilde{\psi}_{CSZ}$ at all points where $\widetilde{\psi}_{CSZ}$ is positive and zero at all points where $\widetilde{\psi}_{CSZ}$ is negative.

$$\psi_{CSZ}(x) = \begin{cases} \widetilde{\psi}_{CSZ}(x) & \text{if } \widetilde{\psi}_{CSZ}(x) \geq 0; \\ 0 & \text{if } \widetilde{\psi}_{CSZ}(x) < 0. \end{cases} \tag{11}$$

The average flux ψ_A and the slope of the flux ψ_X are determined from $\widetilde{\psi}_{CSZ}$ by Eq. (5), but the spatial integration is restricted to the portion of the cell where $\widetilde{\psi}_{CSZ}$ is positive.

With these definitions for ψ_A, ψ_X, and the outflow from each cell, the moment equations (4) can be written as a nonlinear system with unknowns a_{CSZ} and b_{CSZ}. The CSZ streaming and interaction operator $\mathbf{L}$ and the angle integration operator $\mathbf{P}$ are nonlinearly dependent upon ψ. We represent these operators as $[\mathbf{L}(\psi)]$ and $[\mathbf{P}(\psi)]$, yielding

$$\big[\mathbf{L}(\psi)\big]\psi = \frac{\sigma_s}{4\pi}\big[\mathbf{P}(\psi)\big]\psi + q. \tag{12}$$

The CSZ vector of unknowns ψ contains a_{CSZ} and b_{CSZ} for every cell and direction and must be integrated to obtain ψ_A and ψ_X.

3. Preconditioners

The CSZ moment equations (12) are preconditioned with transport sweeps and with a linear or a nonlinear S_2SA strategy. These three preconditioners are derived in this section in the context of the SI solution method. The NKA method is applied to the equations in this form. The JFNK and MFPK methods require residual calculations; minor algebraic rearrangement of the SI equations yield the residual as a function of the angular flux, $f(\psi)$. Both the SI and residual forms of the preconditioned systems are presented in this section.

3.1. Source iteration and sweep preconditioning

Starting from an initial guess for the angular flux, the SI method lags the source terms and solves for an updated angular flux. The transport operator $[\mathbf{L}(\psi)]$ is block lower triangular in each direction. It is therefore simple to invert and apply this operator. This is referred to as sweep preconditioning, and applying it to Eq. (12) with iteration index (ℓ) yields

$$\psi^{(\ell+1)} = \big[\mathbf{L}\big(\psi^{(\ell)}\big)\big]^{-1}\left(\frac{\sigma_s}{4\pi}\big[\mathbf{P}\big(\psi^{(\ell)}\big)\big]\psi^{(\ell)} + q\right). \tag{13}$$

To calculate the residual in this form, the SI iteration index is dropped and algebraic rearrangement yields

$$f(\psi) = \left(\mathbf{I} - \big[\mathbf{L}(\psi)\big]^{-1}\frac{\sigma_s}{4\pi}\big[\mathbf{P}(\psi)\big]\right)\psi - \big[\mathbf{L}(\psi)\big]^{-1}q. \tag{14}$$

When the scattering ratio, $c = \sigma_s/\sigma_t$, is much less than unity, sweep preconditioning is highly effective. However, if c is close to unity, source iteration very effectively attenuates error modes in the scalar flux with a rapid spatial dependence but does not effectively attenuate error modes with a slow spatial dependence [4,6]. These types of slowly varying error modes have eigenvalues close to one in diffuse problems.

3.2. S_2 synthetic acceleration preconditioning

3.2.1. Diffusion synthetic acceleration preconditioning

Preconditioning with DSA adds an additional step to each source iteration. Let us denote the angular flux after the sweep in Eq. (13) as $\psi^{(\ell+\frac{1}{2})}$ and the solution after the DSA step as $\psi^{(\ell+1)}$. We can solve for an additive correction $\delta\phi^{(\ell+\frac{1}{2})}$ to the scalar flux $\phi^{(\ell+\frac{1}{2})}$ with the diffusion equation,

$$-\frac{d}{dx}\frac{1}{3\sigma_t}\frac{d}{dx}\delta\phi^{(\ell+\frac{1}{2})} + \sigma_a\delta\phi^{(\ell+\frac{1}{2})} = \sigma_s\big[\phi^{(\ell+\frac{1}{2})} - \phi^{(\ell)}\big], \tag{15}$$

and then update the angular flux solution with this correction,

$$\psi^{(\ell+1)} = \psi^{(\ell+\frac{1}{2})} + \frac{1}{4\pi}\delta\phi^{(\ell+\frac{1}{2})}. \tag{16}$$

The cost of solving the diffusion equation is small compared to the cost of solving the transport equation.

The application of DSA preconditioning to source iteration problems reduces the spectral radius from $\rho_{SI} \approx c$ to $\rho_{DSA} \leq 0.2247c$ as the number of discrete angle directions N increases [4,11]. In addition to reducing the magnitude of the spectral radius, wrapping DSA in a Krylov method mitigates DSA degradation in multidimensional problems caused by large discontinuities in cross sections [5]. These results were derived for DSA applied to a linear problem, and it should be noted here that there is no theory guaranteeing that these results will hold when applied to a problem discretized with a nonlinear spatial discretization.

Deriving a diffusion equation that is consistent with the CSZ equations is challenging, especially in two and three dimensional problems; however, as explained in the next section, we can derive a CSZ-consistent diffusion-like nonlinear preconditioner in 1-D. We therefore investigate in 1-D the effectiveness of a linear preconditioner based on the LD equations and a nonlinear preconditioner based on the CSZ equations. It would be desirable if preconditioning with the approximate linear equations proved to be effective in 1-D as this would suggest that one might be able to efficiently use linear diffusion preconditioners in multidimensions and avoid the difficult derivation of CSZ-consistent diffusion preconditioners.

3.2.2. S_2SA preconditioning

The S_2 and P_1 equations are analytically equivalent in one dimension [4]. We apply S_2 Synthetic Acceleration (S_2SA), resulting in a diffusion equation that is discretized in a manner consistent with the S_n equations. S_2SA converges with the same spectral radius as DSA in model infinite medium linear problems [4]. This approach results in an unconditionally convergent acceleration technique. In two and three dimensional problems, no such equivalency exists between the S_2SA and DSA equations [4].

The DSA scheme outlined in Eqs. (15) and (16) is straightforward to apply to our discretized transport equation, Eq. (13), as S_2SA preconditioning. We replace Eq. (15) with an S_2 equation,

$$\mathbf{L}_2 \delta\psi_2^{(\ell+1/2)} = \frac{\sigma_s}{4\pi}\big[\mathbf{P}\big(\psi^{(\ell)}\big)\big]\big(\psi^{(\ell+1/2)} - \psi^{(\ell)}\big), \tag{17}$$

where $\mathbf{L}_2$ contains the S_2 streaming and interaction terms (let us hold off, for a moment, on specifying the spatial discretization used to form this operator). We denote $\delta\psi_2$ with the subscript to emphasize that the solution contains only two angular directions. This system is solved with a direct method. The angular flux solution $\delta\psi_2^{(\ell+1/2)}$ is then interpolated from two angular directions back to N directions. This interpolation is accomplished by calculating $\delta\phi$ and δJ from $\delta\psi_2$ and using the relationship

$$\delta\psi_d = \frac{\delta\phi + 3\mu_d \delta J}{4\pi}, \tag{18}$$

where δJ is the current calculated from $\delta\psi_2$. We denote this interpolation from 2 to N angular directions with the operator $\mathbf{T}$, that is,

$$\delta\psi_d = \mathbf{T}\delta\psi_2. \tag{19}$$

The angular flux solution $\psi^{(\ell+1/2)}$ is updated with this correction $\delta\psi^{(\ell+1/2)}$ to obtain $\psi^{(\ell+1)}$. Applying these steps to the SI form of the transport equation, Eq. (13), and rearranging the terms yields our S_2SA-preconditioned scheme:

$$\begin{aligned}\psi^{(\ell+1)} = {} & \left(\mathbf{I} + \mathbf{T}\mathbf{L}_\mathbf{2}^{-\mathbf{1}} \frac{\sigma_s}{4\pi}\big[\mathbf{P}\big(\psi^{(\ell)}\big)\big]\right)\left(\big[\mathbf{L}\big(\psi^{(\ell)}\big)\big]^{-1}\left(\frac{\sigma_s}{4\pi}\big[\mathbf{P}\big(\psi^{(\ell)}\big)\big]\psi^{(\ell)} + q\right)\right) \\ & - \left(\mathbf{T}\mathbf{L}_\mathbf{2}^{-\mathbf{1}} \frac{\boldsymbol{\sigma}_\mathbf{s}}{\mathbf{4}\boldsymbol{\pi}}\big[\mathbf{P}\big(\psi^{(\ell)}\big)\big]\right)\psi^{(\ell)}.\end{aligned} \tag{20}$$

The residual form of Eq. (20) simplifies to

$$f(\psi) = \left(\mathbf{I} + \mathbf{T}\mathbf{L}_\mathbf{2}^{-\mathbf{1}} \frac{\sigma_\mathbf{s}}{4\pi}\big[\mathbf{P}(\psi)\big]\right)\left(\psi - \big[\mathbf{L}(\psi)\big]^{-1}\left(\frac{\sigma_s}{4\pi}\big[\mathbf{P}(\psi)\big]\psi + q\right)\right). \tag{21}$$

The streaming and interaction terms in the S_2 equations can be constructed using the LD or CSZ definitions, giving rise to a linear LD and a nonlinear CSZ S_2SA preconditioning scheme. The LD matrix, $\widehat{\mathbf{L_2}}$, can be constructed once and re-used at each iteration, but the CSZ matrix depends upon the angular flux, $[\mathbf{L_2}(\psi)]$, and must be constructed at each iteration. This cost is small compared to solving the transport equation.

The restriction from N to two angular directions in the preconditioner strongly dampens negativities in the angular flux. Cells that contain negativities only in steep angular directions often contain strictly positive angular fluxes when discretized with two angular directions. This reduces the number cells where the CSZ equations do not reduce to the LD equations in the S_2 preconditioner.

4. Solution techniques

In the previous section, sweep, LD S_2SA, and CSZ S_2SA preconditioners were outlined as they apply to the SI method. In this section the NKA, JFNK, and MFPK methods are discussed. Each of these methods is based on Newton's method for solving nonlinear systems. The two S_2SA preconditioners will be compared using the SI, NKA, and MFPK methods to assess the relative effectiveness of the preconditioners and nonlinear solvers.

4.1. Newton's method

Newton's method for solving nonlinear equations is applied to the residual equations. The method seeks an angular flux solution ψ such that the residual $f(\psi)$ is zero. At each Newton iteration the system is linearized,

$$J\left(\psi^{(m)}\right)\delta\psi^{(m)} = -f\left(\psi^{(m)}\right), \tag{22a}$$

and solved for an additive correction to $\psi^{(m)}$,

$$\psi^{(m+1)} = \psi^{(m)} + \delta\psi^{(m)}. \tag{22b}$$

Here m is the Newton iteration index and $J(\psi^{(m)})$ is the Jacobian of the residual equations at iteration m. Iterations continue until the relative residual is smaller than some tolerance,

$$\frac{\|f(\psi^{(m)})\|_2}{\|\widehat{L}^{-1}q\|_2} < \epsilon_{tol}, \tag{23}$$

where $\widehat{L}^{-1}q$ is the inhomogeneous source after an LD sweep and ϵ_{tol} is a convergence criteria.

Note that with an initial guess of $\psi^{(0)} = 0$, the CSZ operators reduce to the LD operators because $\psi \geq 0$ at all points. Therefore, the solution after the first Newton iteration is the solution to the LD problem.

4.2. Jacobian-free Newton Krylov

4.2.1. JFNK method

Analytically determining the Jacobian matrix is costly in terms of both computation and storage. A Krylov iterative method for solving the linear system of equations requires only the action of the Jacobian upon some vector v. This action is approximated with a finite difference,

$$Jv \approx \frac{f(\psi + \epsilon v) - f(\psi)}{\epsilon}. \tag{24}$$

Thus evaluating Jv requires only residual evaluations. The perturbed residual evaluation is

$$f(\psi + \epsilon v) = (\psi + \epsilon v) - \left[\mathbf{L}(\psi + \epsilon v)\right]^{-1}\left(\frac{\sigma_s}{4\pi}\left[\mathbf{P}(\psi + \epsilon v)\right](\psi + \epsilon v) - q\right). \tag{25}$$

We use GMRES as the Krylov technique employed to solve Eq. (22a) [12].

4.2.2. Magnitude of the perturbation ϵ

The finite difference approximation of Jv can be sensitive to the magnitude of the scalar perturbation ϵ. If ϵ is very small compared to the flux within a particular cell, the local derivative is lost amid round-off errors. If ϵ is similar in magnitude to the average scalar flux within a cell, however, the local derivative is poorly approximated. A formula to determine ϵ was given in [7] as

$$\epsilon = \frac{\sqrt{(1 + \|\psi\|_2)\epsilon_m}}{\|v\|_2}, \tag{26}$$

where ϵ_m is the machine roundoff.

However, this approach is poorly suited for problems in which ψ spans many orders of magnitude. The value of ϵ determined with Eq. (26) may be small compared to ψ in most cells, but comparable to or significantly larger than ψ in strongly absorbing regions where ψ can be very small or zero. Because the CSZ operators nonlinearly depend upon ψ, the finite difference approximation in Eq. (24) may not always be suitable for a JFNK approach. This problem is inherent in the global nature of ϵ and is not negated by using alternative formulas for ϵ presented in [7].

The difficulty in approximating the Jacobian with finite difference due to the nonlinearity of the CSZ operators prevented the JFNK method from converging when applied to the test problem considered in this work. A matrix-free Picard Krylov method was developed from the JFNK method to solve the CSZ equations.

4.3. Matrix-free Picard Krylov method

The problem with the magnitude of ϵ can be avoided by assembling the CSZ operators using the unperturbed value of ψ and then applying the operators to the perturbed ψ. This "freezes" the Jacobian at the latest Newton iteration for ψ and makes it independent of the perturbation ϵv. The "frozen" version of the perturbed residual shown in Eq. (25) can be expressed as

$$f(\psi + \epsilon v) = (\psi + \epsilon v) - \left[\mathbf{L}(\psi)\right]^{-1}\left(\frac{\sigma_s}{4\pi}\left[\mathbf{P}(\psi)\right](\psi + \epsilon v) - q\right) \tag{27}$$

and thus

$$Jv \approx v - \left[\mathbf{L}(\psi)\right]^{-1}\left(\frac{\sigma_s}{4\pi}\left[\mathbf{P}(\psi)\right]v\right). \tag{28}$$

We refer to this approach as a "Matrix-free Picard Krylov" method.

In cells in which ψ in a particular direction is everywhere nonnegative, the CSZ operator reduces to the LD operator, which is independent of ψ. In these cases the Matrix-free Picard Krylov method is equivalent to the Jacobian-free Newton Krylov method. In cells in which ψ is negative over some portion of the cell, the CSZ operators are constructed using the unperturbed value of ψ. This is equivalent to a Picard method in these cells.

Although this strategy resembles a Jacobian-free Newton method, by freezing the Jacobian at the most recent Newton guess and solving the frozen system, this method is equivalent to Picard iteration using GMRES to solve the linear system of equations at each Picard step.

When applied to a problem discretized with the LD definitions, the JFNK and MFPK methods are equivalent and solve the system of linear equations with GMRES. When applied to a problem discretized with the CSZ definitions but with a strictly nonnegative solution, the CSZ method reduces to the LD method and the MFPK and JFNK methods are equivalent.

4.4. Nonlinear Krylov acceleration

The NKA method is an acceleration scheme for fixed-point iterative methods. The NKA method is closely related to the Anderson Mixing method first published in [13]. NKA is essentially a nonlinear version of GMRES [8,14,15]. When applied to a linear problem, and if certain conditions are met, the NKA subspace can be shown to be equivalent to the Krylov subspace built by GMRES [14], and in [15] Anderson Mixing is shown to be "essentially equivalent" to GMRES for a linear problem. The central theme of the NKA method is to use information from previous iterations to inform future additive corrections to the solution. We briefly outline the method here, following [8] and [9].

4.4.1. SI recast as an approximate Newton's Method

Let us compare a fixed-point iterative scheme (such as SI) and Newton's method. The standard SI method is

$$\psi^{(m+1)} = \left[\mathbf{L}\left(\psi^{(m)}\right)\right]^{-1}\left(\frac{\sigma_s}{4\pi}\left[\mathbf{P}\left(\psi^{(m)}\right)\right]\psi^{(m)} + q\right). \tag{29}$$

The residual associated with this system is

$$f\left(\psi^{(m)}\right) = \psi^{(m)} - \psi^{(m+1)}, \tag{30a}$$

$$f\left(\psi^{(m)}\right) = \left(I - \left[\mathbf{L}\left(\psi^{(m)}\right)\right]^{-1}\frac{\sigma_s}{4\pi}\left[\mathbf{P}\left(\psi^{(m)}\right)\right]\right)\psi^{(m)} - \left[\mathbf{L}\left(\psi^{(m)}\right)\right]^{-1}q. \tag{30b}$$

Our fixed point iterative method is simply Eq. (30a):

$$\psi^{(m+1)} = \psi^{(m)} - f\left(\psi^{(m)}\right). \tag{31}$$

Expressed in this fashion, SI is seen to be an approximate Newton's method using the identity instead of the Jacobian.

4.4.2. NKA

This derivation of the NKA method follows [8]. The starting point for the NKA method is to express Newton's method in the following form:

$$\psi^{(m+1)} = \psi^{(m)} - \left[J\left(\psi^{(m)}\right)\right]^{-1} f\left(\psi^{(m)}\right). \tag{32}$$

Note that if we substitute the identity operator for the Jacobian in Eq. (32), we obtain Eq. (31). Thus fixed-point iteration can be thought of as an approximate Newton iteration. The basic idea of the NKA method is to define an acceleration step that is performed after a fixed-point iteration to obtain an approximation to Eq. (32) that is better than a fixed-point iteration alone.

The NKA algorithm requires access to $\psi^{(m)}$, $\psi^{(m-1)}$, and the residuals at those iterations $f(\psi^{(m)})$ and $f(\psi^{(m-1)})$. Note that a fixed-point iteration is required to evaluate $f(\psi^{(m)})$. Also note that initially one has only $\psi^{(0)}$, and after one initial fixed-point iteration one has $f(\psi^{(0)})$, but one needs $\psi^{(-1)}$ and $f(\psi^{(-1)})$ to perform an acceleration step, so the first fixed-point iteration is not accelerated. After the second fixed-point iteration, one has the required quantities for an acceleration step. Thus for each $m > 0$, one performs a fixed-point iteration to obtain $f(\psi^{(m)})$ and constructs the next member of each of the following arrays:

$$v_m = \psi^{(m)} - \psi^{(m-1)}, \tag{33a}$$

$$w_m = f\left(\psi^{(m)}\right) - f\left(\psi^{(m-1)}\right). \tag{33b}$$

If we Taylor-series expand the residual about $\psi^{(m-1)}$, recognizing that the derivative of the residual with respect to ψ is the Jacobian, we obtain

$$f\left(\psi^{(m)}\right) = f\left(\psi^{(m-1)}\right) + \left[J\left(\psi^{(m-1)}\right)\right]\left(\psi^{(m)} - \psi^{(m-1)}\right). \tag{34}$$

It is easily seen from Eq. (34) that v_m and w_m are related by the Jacobian at $\psi^{(m)}$,

$$J\left(\psi^{(m)}\right)v_m = w_m. \tag{35}$$

We define V and W as the spaces spanned by the vectors v_i and w_i, $i = 1, \ldots, m$, respectively. If the problem was linear, the Jacobian would be a constant and all vectors v_i and w_i would be related by the same Jacobian matrix. In addition, any linear combination of vectors v_i and w_i sharing the same coefficients α_i would be related by this constant Jacobian:

$$J\sum_{i=1}^{m}\alpha_i v_i = \sum_{i=1}^{m}\alpha_i w_i, \tag{36a}$$

and

$$\sum_{i=1}^{m}\alpha_i v_i = J^{-1}\sum_{i=1}^{m}\alpha_i w_i. \tag{36b}$$

We next define f_m^W as a projection of the residual at $\psi^{(m)}$ onto the space W,

$$f_m^W = \sum_{i=1}^{m}\alpha_i w_i, \tag{37a}$$

where the α_i coefficients are the solution to the least-squares problem

$$\{\alpha_1, \alpha_2, \ldots, \alpha_m\} = \underset{\alpha' \in \mathbb{R}^m}{\arg\min}\left\|\sum_{i=1}^{m}\alpha_i' w_i - f\left(\psi^{(m)}\right)\right\|_2. \tag{37b}$$

These coefficients can be calculated by a Cholesky decomposition of the Gramian of the w_i vectors.

Recall that the goal of the NKA algorithm is to define an acceleration step for a fixed-point iterative method that approximates Newton's Method, Eq. (32). This is accomplished by first re-expressing a fixed-point iteration,

$$\psi^{(m+1)} = \psi^{(m)} - f\left(\psi^{(m)}\right), \tag{38}$$

in terms of the components of the residual that lie within W, f_m^W, and components of the residual orthogonal to W, $f(\psi^{(m)}) - f_m^W$:

$$\psi^{(m+1)} = \psi^{(m)} - \left\{f_m^W + \left[f\left(\psi^{(m)}\right) - f_m^W\right]\right\}. \tag{39}$$

The action of J^{-1} on f_m^W is given, for a constant Jacobian, in Eq. (36b). We have no way to apply J^{-1} to the portion of the residual that lies outside of W, so the standard fixed point iteration is performed on this portion of the residual. The NKA iteration scheme is

$$\psi^{(m+1)} = \psi^{(m)} - J^{-1}f_m^W - \left[f\left(\psi^{(m)}\right) - f_m^W\right]. \tag{40}$$

Applying the definitions for $J^{-1}f_m^W$ and f_m^W in Eqs. (36b) and (37a), respectively, yields the NKA algorithm:

$$\psi^{(m+1)} = \psi^{(m)} - \sum_{i=1}^{m}\alpha_i v_i - \left[f\left(\psi^{(m)}\right) - \sum_{i=1}^{m}\alpha_i w_i\right]. \tag{41}$$

Note from Eq. (40) that if the residual lies entirely outside of W a fixed-point iteration is performed; if the residual lies entirely within W a Newton iteration is performed under the assumption of a constant Jacobian. In general, a portion of the residual will lie inside of W and the NKA method is expected to be more effective than the fixed-point method but less effective than Newton's method.

In order to mitigate the effect of a rapidly changing Jacobian and to reduce memory requirements, only a limited number of vectors v_i and w_i are stored. The dimension of the subspace W is set by the user. For a dimension of size d, the d most recent vectors v_i and w_i are saved in V and W. Before adding a vector w_i to W, the angles between w_i and each of the existing members of W are calculated. In each case in which this angle is less than a set tolerance, the existing member of W is removed from the subspace. This ensures that the subspace is composed of linearly independent vectors.

The total memory cost of NKA is $2d$ vectors because at each iteration both v_i and w_i are stored. These vectors have the dimension of the unknown vector. Note that the standard GMRES algorithm stores one vector per iteration. To compare

Table 1
Geometry of the five-region Reed-like problem.

Property	Region				
length (cm)	2	1	2	1	2
length (mfp)	100	5	n/a	20	40
σ_s (cm^{-1})	0	0	0	19.99	19.99
σ_t (cm^{-1})	50	5	0	20.00	20.00
c	0	0	n/a	0.9995	0.9995
Q (p/(cm^3 s))	50	0	0	0.1	0

the NKA and JFNK or MFPK methods with similar memory requirements, the Krylov dimension used for the JFNK or MFPK methods should be twice the dimension of the NKA method's subspace.

Thus, as previously noted, we see that the NKA method is a type of approximate Newton's method. It should never be expected to perform better than Newton's method or perform worse than fixed-point iteration. It will clearly work best with either a weakly nonlinear function or an initial guess close to the solution. For this reason, we perform some number of initial fixed-point iterations before applying NKA. These initial iterations increase the stability of the method for the problem we consider in this work.

5. Computational results

A challenging test problem was designed to assess the effectiveness of the LD and CSZ S_2SA preconditioners when applied using the SI, MFPK, and NKA methods. The test problem contains both highly diffusive regions and regions that induce negativities in the LD solution so that the problem benefits from the CSZ spatial discretization and S_2SA preconditioning.

The SI and NKA methods were iterated until the L_2 norm of the residual divided by the L_2 norm of $\widehat{L}^{-1}q$ (that is, the inhomogeneous source after an LD sweep) was smaller than a set tolerance. The Picard iterations in the MFPK method were solved to this tolerance. The linear solve in the MFPK method used a tolerance two orders of magnitude smaller than the Picard tolerance.

The effectiveness of the S_2SA preconditioners is assessed by comparing the number of iterations required to solve the problem. The MFPK method lends itself to two additional metrics for comparing the preconditioners. Although the method is matrix-free, it was possible to assemble the matrix and determine its condition number and eigenvalues for diagnostic purposes. These results are presented with the MFPK results.

The MFPK and NKA methods are compared to assess the relative effectiveness of each method. The methods are compared by the number of iterations and the time required to solve the problem. As the Newton steps are inexpensive compared to the Krylov iterations, the cost of the MFNK method is measured as the total number of Krylov iterations; the cost of the NKA method is measured in the total number of iterations. The timing results are presented with a 95% confidence interval; in the cases in which the 95% range was smaller than one hundredth of a second, the interval was rounded up to one hundredth of a second.

These results were computed using the Texas A&M Nuclear Engineering Department's cluster, *GROVE*, using MATLAB R2008b.

5.1. Reed-like problem

Our test problem consists of five regions, including strongly absorbing regions, regions with $c \approx 1$, and a region of vacuum. The problem is based on a problem proposed by Reed in [16]. We solve the problem with an S_{16} angular quadrature, 80 equally-sized spatial cells, and a tolerance of 1E−6 for SI and NKA. The Krylov iterations in the MFPK method used a tolerance of 1E−8 and the MFPK Picard iterations used a tolerance of 1E−6. The NKA method rejected vectors that were within 0.001 radians of the proposed correction, and the first three source iterations were performed without applying NKA.

The geometry and scalar flux solution of the Reed-like problem are presented in Table 1 and Fig. 1, respectively.

5.2. Preconditioner effectiveness

The cost of solving the Reed-like problem with the LD and CSZ spatial discretizations is presented for each solution method. The LD equations are solved with sweep-preconditioning and with LD S_2SA preconditioning. The CSZ equations are solved with sweep-preconditioning, with the approximate linear LD S_2SA preconditioner, and with the nonlinear CSZ S_2SA preconditioner. The results generated with the MFPK method are reported with Krylov subspaces of dimension 10, 20, and with an unrestarted subspace. The NKA results are reported with a subspace of dimension 5, 10, and an unrestarted subspace. Recall that because the NKA method stores two vectors for each entry in the subspace, the memory cost of NKA(5) is comparable to the memory cost of the MFPK(10) method.

5.2.1. Source iteration

The number of source iterations and the time required to solve the problem with sweep-preconditioning, with LD S_2SA, and with CSZ S_2SA preconditioning are presented in Table 2.

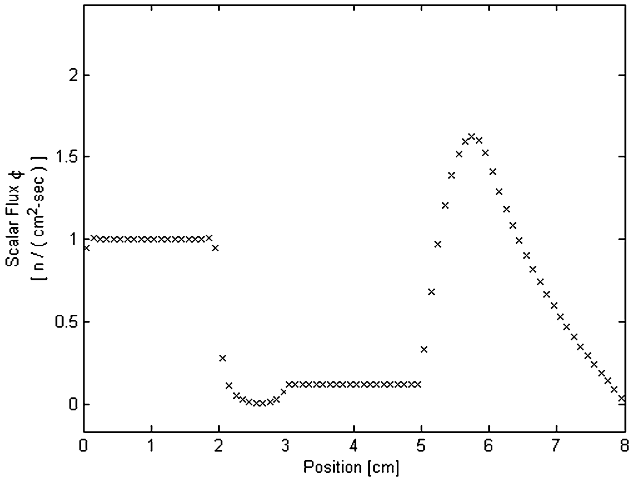


Fig. 1. Scalar flux solution of the Reed-like problem using S_{16} quadrature and 80 cells.

Table 2
Cost of solving the Reed-like problem with source iteration.

Scheme	Source Iterations	Time (s)
LD Equations	5293	68.13 ± 0.22
LD Eqs., LD S_2SA	8	0.17 ± 0.01
CSZ Equations	5293	337.21 ± 0.37
CSZ Eqs., LD S_2SA	8	0.64 ± 0.01
CSZ Eqs., CSZ S_2SA	8	0.68 ± 0.01

Table 3
Number of iterations required to solve the Reed-like problem with nonlinear Krylov acceleration.

Scheme	NKA(5)	NKA(10)	NKA
LD Equations	177	95	45
LD Eqs., LD S_2SA	6	6	6
CSZ Equations	210	84	60
CSZ Eqs., LD S_2SA	10	10	10
CSZ Eqs., CSZ S_2SA	10	10	10

The sweep-preconditioned system is expensive to solve, as is expected for a problem with a scattering ratio close to one. The LD and CSZ equations require exactly the same number of source iterations to solve, but the CSZ equations required about five times as much time to solve as the LD equations. Each nonlinear source iteration is several times more expensive than a linear source iteration. In addition to requiring a check for negativities in each cell in each direction, the CSZ method requires constructing the $L(\psi)$ matrix for each cell in each direction. The LD method only builds the $\widehat{L}$ matrix once per direction. The LD S_2SA-preconditioned CSZ equations were similarly several times more expensive to solve than the LD S_2SA-preconditioned LD equations. Although the S_2SA-preconditioned LD and CSZ equations required the same number of source iterations to converge, the CSZ solution was several times more expensive to generate than the LD solution.

Both of the S_2SA preconditioners proved extremely effective, reducing the required number of iterations by a factor of more than 650 and the required time by a factor of about 400. The CSZ S_2SA preconditioner was slightly more expensive than the LD S_2SA preconditioner, but the difference in the cost associated with each preconditioner was not significant for this problem.

5.2.2. Nonlinear Krylov acceleration

The number of NKA iterations required to solve the problem with a subspace of dimension 5, 10, and with an arbitrarily large subspace are presented in Table 3. The time required to generate these solutions is presented in Table 4. The memory requirements increase as the subspace dimension increases. Note that the full angular flux is stored, thus the memory requirement is significant.

The sweep-preconditioned systems of equations benefited from increasing the dimension of the subspace. As the subspace size increased from five to an unlimited number of vectors, the number of iterations and the amount of time required to solve the problem were both greatly reduced.

As with the SI method, solving the CSZ system of equations required more iterations and significantly more time than solving the LD equations. Although for this particular problem the NKA(10) method solves the CSZ equations in fewer

Table 4
Cost (in seconds) of solving the Reed-like problem with nonlinear Krylov acceleration. Results presented with 95% confidence.

Scheme	NKA(5)	NKA(10)	NKA
LD Equations	2.54 ± 0.01	1.46 ± 0.01	0.77 ± 0.01
LD Eqs., LD S_2SA	0.14 ± 0.01	0.14 ± 0.01	0.14 ± 0.01
CSZ Equations	13.70 ± 0.05	5.62 ± 0.02	4.29 ± 0.01
CSZ Eqs., LD S_2SA	0.78 ± 0.01	0.78 ± 0.01	0.78 ± 0.01
CSZ Eqs., CSZ S_2SA	0.82 ± 0.01	0.82 ± 0.01	0.82 ± 0.01

Table 5
Cost of solving the Reed-like problem with Matrix-free Picard Krylov method. Results are presented as the "Number of Picard iterations, Total number of Krylov iterations" required to solve the problem.

Scheme	MFPK(10)	MFPK(20)	MFPK
LD Equations	1, 320	1, 193	1, 47
LD Eqs., LD S_2SA	1, 8	1, 8	1, 8
CSZ Equations	6, 350	6, 222	6, 76
CSZ Eqs., LD S_2SA	6, 22	6, 22	6, 22
CSZ Eqs., CSZ S_2SA	6, 22	6, 22	6, 22

Table 6
Cost (in seconds) of solving the Reed-like problem with Matrix-free Picard Krylov method. Results are presented with 95% confidence.

Scheme	MFPK(10)	MFPK(20)	MFPK
LD Equations	4.80 ± 0.02	2.88 ± 0.02	0.81 ± 0.01
LD Eqs., LD S_2SA	0.18 ± 0.01	0.18 ± 0.01	0.18 ± 0.01
CSZ Equations	27.10 ± 0.03	16.63 ± 0.05	5.76 ± 0.01
CSZ Eqs., LD S_2SA	2.05 ± 0.01	2.05 ± 0.01	2.05 ± 0.01
CSZ Eqs., CSZ S_2SA	2.08 ± 0.01	2.08 ± 0.01	2.08 ± 0.01

iterations than the LD equations, this will not generally be true. Both the LD and CSZ S_2SA preconditioners were extremely effective at reducing the total cost of solving the problem, in terms of both the required number of iterations and the required time. The CSZ S_2SA preconditioned equations required the same number of iterations and slightly more time to converge than the LD S_2SA-preconditioned equations. The difference in the effectiveness of the two preconditioners was not significant for this particular problem.

5.2.3. Matrix-free Picard Krylov

The number of Picard and Krylov iterations required to solve the problem with the MFPK method are presented in Table 5. The time required to generate these solutions is presented in Table 6. The results are reported using a GMRES Krylov subspace of dimension 10, a subspace of dimension 20, and an unrestarted GMRES subspace. As with the NKA method, the required memory increases with the dimension of the Krylov subspace.

As the dimension of the GMRES Krylov subspace is increased, the total number of Krylov iterations required to solve the sweep-preconditioned problems is greatly reduced. The S_2SA-preconditioned LD and CSZ systems of equations required between two and seven Krylov iterations to converge each Picard iteration, so increasing the subspace size from 10 has no effect on the cost of solving these systems.

The MFPK method solves the linear problems with a single Picard iteration. Recall that using an initial guess of $\psi = 0$, the CSZ equations reduce to the LD equations and the CSZ and LD S_2SA preconditioners are identical. Therefore, the first Picard iteration yields the LD solution. The additional Picard iterations yield the nonnegative CSZ solution. That is, using the MFPK(10) method to solve the CSZ equations with sweep-preconditioning, the first Picard step takes 320 Krylov iterations and yields the LD solution. Five additional Picard steps and an additional 30 Krylov iterations are required to converge to the strictly non-negative CSZ solution.

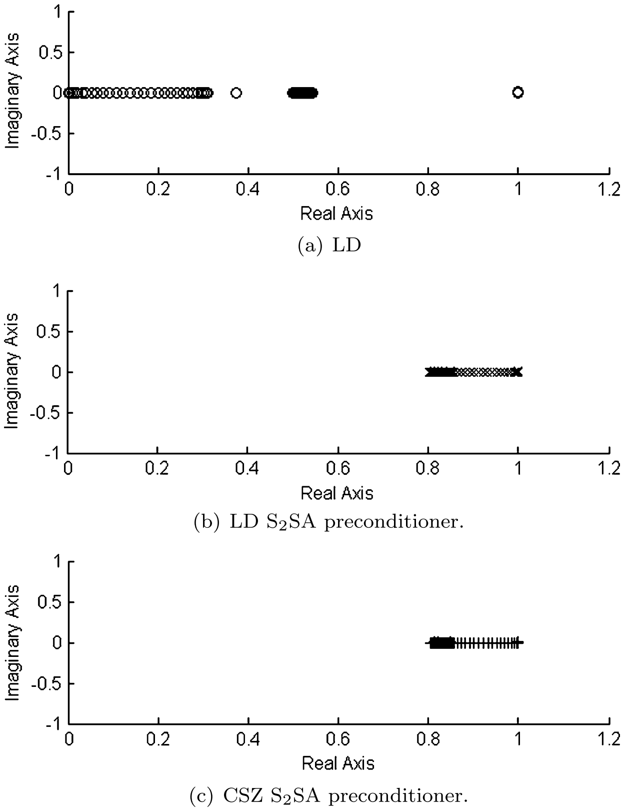
The LD and CSZ S_2SA preconditioners are both extremely effective at reducing the cost of the solution. The CSZ S_2SA preconditioner was slightly more time-consuming than the LD S_2SA preconditioner, but this difference was not significant. The two preconditioners were similarly effective when applied to this particular problem with the MFPK method.

5.2.4. Analysis of the MFPK matrix

Although the MFPK method avoids building the (approximate) Jacobian matrix, it is possible to build the matrix at each Picard step to investigate its properties. The condition number of the matrix at each Picard step for the sweep-preconditioned system, the LD S_2SA-preconditioned system, and the CSZ S_2SA-preconditioned system is presented in Table 7.

Table 7
The condition number of the nonlinear Reed-like problem's approximate Jacobian matrix at each Picard step with sweep-preconditioning, with the LD S_2SA preconditioner, and with the CSZ S_2SA preconditioner.

Scheme	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6
LD	3165	3159	3160	3160	3160	3160
LD S_2SA	5.321	5.317	5.333	5.340	5.341	5.341
CSZ S_2SA	5.321	5.328	5.344	5.352	5.352	5.352



(a) LD

(b) LD S_2SA preconditioner.

(c) CSZ S_2SA preconditioner.

Fig. 2. Eigenvalues of the Reed-like problem at Picard step two for the (a) sweep-preconditioned system, (b) LD S_2SA-preconditioned system, and (c) the CSZ S_2SA-preconditioned system.

The sweep-preconditioned MFPK method had a large condition number. Both of the S_2SA preconditioners yield a matrix with a greatly reduced condition number. The difference in the condition number generated with the two preconditioners does not appear to be significant.

In addition to the condition number, we can examine the eigenvalues of the matrix associated with each of the three systems of equations at each Picard iteration. The eigenspectra at the second Picard iteration is fairly representative of the eigenspectrum at each of the Picard steps and is presented in Fig. 2. Recall that the CSZ equations reduce to the LD equations in the first Picard step with our initial guess of $\psi = 0$; for this reason the eigenspectra are shown for the second Picard step, where the CSZ and LD equations are not identical.

The sweep-preconditioned equations have real eigenvalues spread between 0 and 1. The S_2SA preconditioners effectively move the eigenvalues away from zero and cluster them about 1. The eigenvalues of the DSA iteration matrix for an infinite homogeneous system with isotropic scattering are all positive and lie between 0 and 0.2247. We subtract the S_2SA preconditioned residual from the identity matrix times the residual. We therefore expect this problem's eigenvalues to range between 0.775 and 1 if we use a sufficiently large number of angular directions. Indeed, our smallest eigenvalue with both S_2SA preconditioners is 0.8045 and the largest eigenvalue is 1.0000.

The eigenvalues resulting from the two S_2SA-preconditioned systems of equations largely overlap. As with the condition number, this suggests that the LD S_2SA preconditioner will be as effective as the CSZ S_2SA preconditioner.

The CSZ S_2SA preconditioner does not outperform the LD S_2SA preconditioner in the total number of Krylov iterations required to converge, the condition number or the eigenspectra of the approximate Jacobian matrix. In fact, with these three metrics, the LD S_2SA preconditioner almost exactly matched the performance of the CSZ S_2SA preconditioner when applied with an MFPK method to this particular problem.

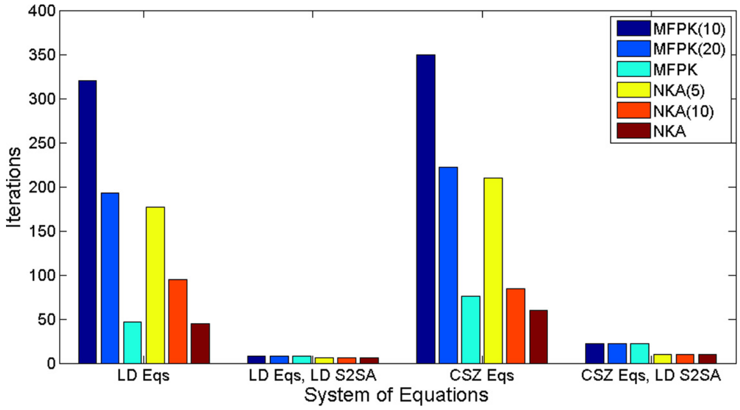


Fig. 3. Iterations required to solve the Reed-like problem with various spatial discretizations and preconditioners using the MFPK and NKA methods.

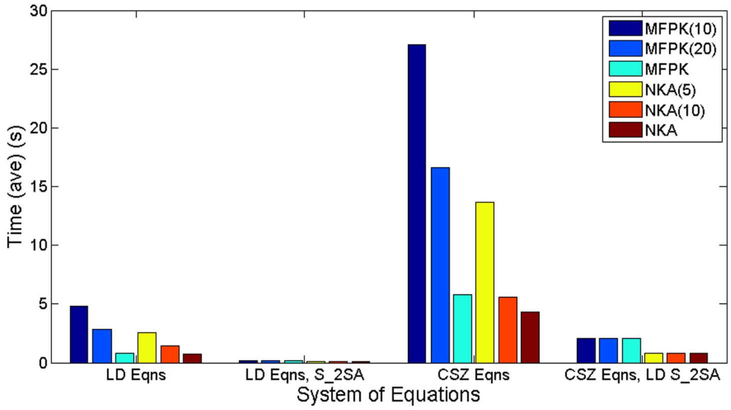


Fig. 4. Cost (in seconds) of solving the Reed-like problem with various spatial discretizations and preconditioners using the MFPK and NKA methods.

It is not clear that the errors upon convergence must necessarily be the same for the sweep-preconditioned calculations versus the S_2SA calculations even though the same convergence criteria were used because the sweep-only nonlinear system is more poorly conditioned than the S_2SA nonlinear systems. Thus one might expect the sweep-only solution to be somewhat less converged than the S_2SA solutions. However, we have determined that this is not the case for the Reed-like problem. We intend to further investigate this issue in the future.

5.3. Solution method effectiveness

Finally, we compare the relative effectiveness of the SI, NKA, and MFPK methods for solving the linear LD and nonlinear CSZ systems of equations. Recall that the NKA method requires twice as much memory for a given subspace dimension as the MFPK methods. We therefore compare NKA(5) with MFPK(10), NKA(10) with MFPK(20), and the unrestarted NKA and MFPK methods.

The source iteration method was many times more expensive than the NKA and MFPK methods with sweep-preconditioning. However, with LD S_2SA preconditioning, the SI method was actually less expensive than the alternative methods. SI required only 8 iterations to converge the CSZ equations with LD S_2SA preconditioning, while NKA required 10 iterations and MFPK required 22 iterations to converge.

The number of iterations required to solve the system of equations (presented in Tables 3 and 5) is presented graphically in Fig. 3. Each MFPK and NKA iteration requires performing one sweep. The solution cost in seconds (from Tables 4 and 6) is presented in Fig. 4. Note that the SI cost data is not included in these figures.

The additional cost associated with solving the nonlinear CSZ equations instead of the linear LD equations is visible in both Figs. 3 and 4. Although only a few extra iterations are required to solve the nonlinear equations instead of the linear equations, each CSZ iteration is more expensive than an LD iteration. The CSZ equations therefore require significantly more time to solve than the corresponding LD equations.

For this test problem the overheads associated with the nonlinear solvers were insignificant compared to the cost of performing the transport sweeps. The difference in the costs of the MFPK and NKA methods in Fig. 4 therefore closely follow the trends in Fig. 3. The NKA methods are significantly less expensive than the MFPK methods. When applied to the sweep-preconditioned CSZ equations, the NKA(5) method required about half as much time as the MFPK(10) method and was slightly faster than the MFPK(20) method. The NKA(10) method was about as fast as the unrestarted GMRES method. The NKA methods were significantly faster than the MFPK methods when applied to the sweep-preconditioned CSZ equations for all subspace dimensions.

It is an advantage of the MFPK method that the LD solution is calculated in the first Picard iteration. This makes it straightforward to assess the additional cost of solving a nonlinear system of equations and allows the solution generated with CSZ spatial discretization to be compared to the solution generated with the LD discretization.

The LD S_2SA preconditioner was extremely effective at reducing the iteration and time cost of each method. The NKA methods required about half as many iterations and half as much time as the MFPK methods to solve the LD S_2SA-preconditioned CSZ equations.

6. Conclusions

We have efficiently solved the CSZ S_n equations in a 1-D problem with both strongly absorbing and highly diffusive regions using source iteration, the matrix-free Picard Krylov method, and the nonlinear Krylov acceleration method in conjunction with S_2SA preconditioning of the nonlinear S_n system. This approach yielded a robust and efficient solution algorithm for the test problem we considered. Such behavior is in contrast to that of traditional DSA and DSA-like acceleration schemes that can become unstable when applied in conjunction with traditional negative flux fixup algorithms. Furthermore, the scheme is not significantly more complex to implement than traditional S_2SA solution techniques.

Preconditioning with LD S_2SA, which is equivalent to LD DSA in 1-D, significantly accelerated the solution of the CSZ equations. Furthermore, LD S_2SA was as effective as CSZ S_2SA, which is similar but not equivalent to CSZ DSA. This result suggests that it might be possible to substitute LD DSA for CSZ DSA as a preconditioner without a significant penalty in multidimensional CSZ S_n calculations. This question should be addressed in the future since S_2SA is not a good alternative to DSA in multidimensional calculations and LD diffusion discretizations are easier to derive and implement than CSZ diffusion discretizations.

The NKA methods proved significantly less expensive than the comparable MFPK methods when applied to solve the sweep-preconditioned nonlinear CSZ equations. This was similarly true for the nonlinear equations preconditioned with LD S_2SA. Source iteration using LD S_2SA preconditioning was more effective than both the preconditioned MFPK and the preconditioned NKA methods for the problem considered in this work.

References

[1] P.G. Maginot, J.E. Morel, J.C. Ragusa, A non-negative moment-preserving S_n spatial discretization scheme for 1-D and 2-D Cartesian geometries, J. Comput. Phys. 231 (20) (2012) 6801–6826.
[2] P. Maginot, A nonlinear positive extension of the linear discontinuous spatial discretization of the transport equation, Masters thesis, Texas A&M University, College Station, TX, December 2010.
[3] L.J. Lorence Jr., J.E. Morel, E.W. Larsen, An S_2-synthetic acceleration scheme for the one-dimensional S_n equations with linear-discontinuous spatial differencing, Nucl. Sci. Eng. 101 (1989) 341–351.
[4] M.L. Adams, E.W. Larsen, Fast iterative methods for discrete-ordinates particle transport calculations, Prog. Nucl. Energy 40 (2002) 3–159.
[5] J.S. Warsa, T.A. Wareing, J.E. Morel, Krylov iterative methods and the degraded effectiveness of diffusion synthetic acceleration for multidimensional S_N calculations in problems with material discontinuities, Nucl. Sci. Eng. 147 (2004) 218–248.
[6] E.W. Larsen, J.E. Morel, Advances in discrete ordinates methodology, in: Y. Azmy, E. Sartory (Eds.), Nuclear Computational Science: A Century in Review, Springer, New York, NY, 2010.
[7] D.A. Knoll, D.E. Keyes, Jacobian-free Newton–Krylov methods: a survey of approaches and applications, J. Comput. Phys. 193 (2) (2004) 357–397.
[8] M.T. Calef, E.D. Fichtl, J.S. Warsa, M. Berndt, N.N. Carlson, A nonlinear Krylov accelerator for the Boltzmann *k*-eigenvalue problem, J. Comput. Phys. 238 (2013) 188–209.
[9] E.D. Fichtl, J.S. Warsa, M.T. Calef, Nonlinear acceleration of S_N transport calculations, in: International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2011), Rio de Janeiro, RJ, Brazil, May 2011, pp. 8–12.
[10] W.H. Reed, T.R. Hill, Triangular mesh methods for the neutron transport equation, Technical report LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.
[11] E.W. Larsen, Unconditionally stable diffusion-synthetic acceleration methods for the slab geometry discrete-ordinates equations. Part I: Theory, Nucl. Sci. Eng. 82 (1982) 47–64.
[12] Y. Saad, M.H. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 7 (3) (1986) 856–869.
[13] D.G. Anderson, Iterative procedures for nonlinear integral equations, J. ACM 12 (4) (1965) 547–560.
[14] N.N. Carlson, K. Miller, Design and application of a gradient-weighted moving finite element code I: in one dimension, J. Sci. Comput. 19 (3) (1998) 728–765.
[15] H.F. Walker, P. Ni, Anderson acceleration for fixed-point iterations, SIAM J. Numer. Anal. 49 (2011) 1715–1735.
[16] W.H. Reed, New difference schemes for the neutron transport equation, Nucl. Sci. Eng. 45 (1971) 309–314.