Journal of Computational Physics 274 (2014) 681–694

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A comparison of acceleration methods for solving the neutron transport k-eigenvalue problem

Jeffrey Willert [*], H. Park, D.A. Knoll

Theoretical Division, MS B216, Los Alamos National Laboratory, Los Alamos, NM 87545, United States

ARTICLE INFO

Article history:
Received 28 January 2014
Received in revised form 18 June 2014
Accepted 20 June 2014
Available online 28 June 2014

Keywords:
Neutron transport
k-eigenvalue problem
Jacobian-free Newton–Krylov
Nonlinear Krylov acceleration
Anderson acceleration
Moment-based acceleration

ABSTRACT

Over the past several years a number of papers have been written describing modern techniques for numerically computing the dominant eigenvalue of the neutron transport criticality problem. These methods fall into two distinct categories. The first category of methods rewrite the multi-group k-eigenvalue problem as a nonlinear system of equations and solve the resulting system using either a Jacobian-Free Newton–Krylov (JFNK) method or Nonlinear Krylov Acceleration (NKA), a variant of Anderson Acceleration. These methods are generally successful in significantly reducing the number of transport sweeps required to compute the dominant eigenvalue. The second category of methods utilize Moment-Based Acceleration (or High-Order/Low-Order (HOLO) Acceleration). These methods solve a sequence of modified diffusion eigenvalue problems whose solutions converge to the solution of the original transport eigenvalue problem. This second class of methods is, in our experience, always superior to the first, as most of the computational work is eliminated by the acceleration from the LO diffusion system. In this paper, we review each of these methods. Our computational results support our claim that the choice of which nonlinear solver to use, JFNK or NKA, should be secondary. The primary computational savings result from the implementation of a HOLO algorithm. We display computational results for a series of challenging multi-dimensional test problems.

1. Introduction

We are interested in computing the dominant eigenvalue–eigenvector pair, (k_{eff}, ϕ), of the neutron transport k-eigenvalue problem [13],

$$\hat{\Omega} \cdot \nabla \psi(\hat{\Omega}, \vec{r}, E) + \Sigma_t(E) \psi(\hat{\Omega}, \vec{r}, E) = \frac{1}{4\pi} \left[\int_0^\infty \Sigma_s(E' \to E) \phi(\vec{r}, E') dE' + \frac{\chi(E)}{k_{eff}} \int_0^\infty \nu \Sigma_f(E') \phi(\vec{r}, E') dE' \right]. \tag{1}$$

The various parameters and variables in Eq. (1) are:

- $\vec{r}$ represents a point in $\mathbb{R}^3$.
- $\hat{\Omega}$ is a unit vector in $\mathbb{R}^3$, represents the direction of neutron travel.
- E represents the neutron energy, $E \in (0, \infty)$.
- ψ is the angular flux.
- ϕ is the zeroth angular moment of ψ, $\phi = \int_{4\pi} \psi \, d\hat{\Omega}$.

[*] Corresponding author.

http://dx.doi.org/10.1016/j.jcp.2014.06.044

- Σ_t, Σ_s and Σ_f are the total, scattering and fission macroscopic cross-sections, respectively.
- ν is the mean number of neutrons produced per fission event.
- χ is the fission neutron energy distribution function.

In order to solve Eq. (1), we apply a multi-group discretization in energy [13]. The resulting equation is given by

$$\hat{\Omega} \cdot \nabla \psi_g(\hat{\Omega}, \vec{r}) + \Sigma_{t,g} \psi_g(\hat{\Omega}, \vec{r}) = \frac{1}{4\pi}\left[\sum_{g'=1}^{G} \Sigma_s^{g' \to g} \phi_{g'}(\vec{r}) + \frac{\chi_g}{k_{eff}} \sum_{g'=1}^{G} \nu \Sigma_{f,g'} \phi_{g'}(\vec{r})\right], \tag{2}$$

where

$$\psi_g(\hat{\Omega}, \vec{r}) = \int_{E_g}^{E_{g-1}} \psi(\hat{\Omega}, \vec{r}, E)\, dE. \tag{3}$$

We will utilize an S_n quadrature in angle [13], yielding

$$\hat{\Omega}_m \cdot \nabla \psi_{g,m}(\vec{r}) + \Sigma_{t,g} \psi_{g,m}(\vec{r}) = \frac{1}{4\pi}\left[\sum_{g'=1}^{G} \Sigma_s^{g' \to g} \phi_{g'}(\vec{r}) + \frac{\chi_g}{k_{eff}} \sum_{g'=1}^{G} \nu \Sigma_{f,g'} \phi_{g'}(\vec{r})\right], \tag{4}$$

where

$$\psi_{g,m} = \psi_g(\hat{\Omega}_m), \tag{5}$$

and integration can now be represented

$$\phi_g = \sum_m \psi_{g,m} w_m, \tag{6}$$

in which $\{w_m\}$ is the set of quadrature weights. We will leave the details of the spatial discretization unspecified, but we will utilize either a diamond-difference (DD) or a step-characteristics (SC) discretization in space [13]. The impact of alternative choices for spatial discretization on the main results of this study are negligible.

In order to simplify future discussion, we will represent the eigenvalue problem (Eq. (4)) in operator notation,

$$\mathcal{L}\Psi = \frac{1}{4\pi}\left[\mathcal{S} + \frac{1}{k_{eff}}\mathcal{F}\right]\Phi,$$

in which

$$\mathcal{L} = \hat{\Omega} \cdot \nabla + \Sigma_t$$

$$\mathcal{S} = \Sigma_s$$

$$\mathcal{F} = \chi \nu \Sigma_f.$$

In this setting, Ψ and Φ represent vectors of group angular and scalar fluxes, respectively,

$$\Psi = [\psi_1, \psi_2, \ldots, \psi_G],$$

$$\Phi = [\phi_1, \phi_2, \ldots, \phi_G].$$

Furthermore, we will define the operator $\mathcal{M}_i$ which takes ψ to its ith angular moment,

$$\phi = \mathcal{M}_0 \psi. \tag{7}$$

The remainder of the paper is organized as follows: In Section 1.1 we will present some preliminary results for a series of three one-dimensional test problems as motivation for the rest of the paper. In Section 2 we will present the power iteration algorithm [13], which provides a baseline to compare all other methods against. We will follow this with Section 2.1, in which we describe using well-known nonlinear solvers to accelerate the solution to the k-eigenvalue problem. In Section 3 we will present the most effective family of methods which we have found for solving the eigenvalue problem, commonly referred to as moment-based acceleration methods or HOLO methods. This algorithm is often referred to as nonlinear diffusion acceleration (NDA) within the neutronics community [6–8,10,17]. We will then provide a thorough comparison of each of these methods in Section 4 and finish the paper by summarizing our findings in Section 5.

One motivation for this study, and paper, comes from [15]. In reference [15], both [8] and [19] are referenced as applying JFNK to the transport eigenvalue problem. While this is accurate on some level, it is misleading at the most important level, as we will demonstrate. The work in [8] uses NDA to accelerate the scattering source and then uses JFNK to accelerate the fission source, but in the LO space. Again, as we will demonstrate, the computational gains made (or lost) by choosing JFNK or NKA are minimal when compared to the computational savings of utilizing NDA.

Table 1
Material properties 1-D test problems.

Property	Material 1	Material 2	Material 3
x-range (cm)	[0, 5]	$[5, \tau - 5]$	$[\tau - 5, \tau]$
Σ_t (cm^{-1})	1	1	1
Σ_s (cm^{-1})	.856	.856	.856
$\nu\Sigma_f$ (cm^{-1})	0	.144	0

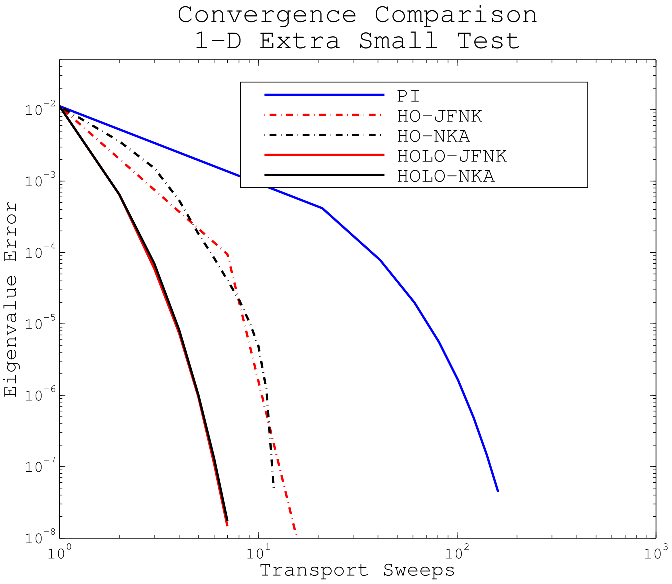

Fig. 1. Methods comparison for 1-D k-eigenvalue problem. 15 mfp domain. Dominance ratio is .3018.

1.1. Motivation

In order to motivate the remainder of the paper, let us consider three one-dimensional, one-group test problems as described in Table 1. Each of these problems consists of the same basic Reflector-Fissionable-Reflector configuration, the only difference is the length of the domain, τ. By increasing τ, the difficulty of the problem increases since the dominance ratio approaches one.

For each of these problems, we will consider five methods. As a baseline, we will solve each of these problems using the power iteration (PI) algorithm [13]. Then, we will solve each problem using the nonlinear formulation which we will discuss in Section 2.1. As nonlinear solvers, we will use both a Jacobian-Free Newton–Krylov (JFNK) method [1–3] and Nonlinear Krylov Acceleration (NKA) [15,14,16]. Details of options and settings used for these nonlinear solvers will be discussed subsequently. Lastly, we will solve each eigenvalue problem using a HOLO method [8]. We will use both JFNK and NKA as LO solvers for comparison purposes. We choose to solve this problem for $\tau = 15$, 35, and 210.

As we can see in Figs. 1–3, as τ increases, so do the number of transport sweeps required by the power iteration algorithm. A *transport sweep* is defined to be the inversion of the operator $\mathcal{L}$, and we will use the number of transport sweeps as a measure of computational cost. The number of transport sweeps required are roughly 180, 4000 and 70 000 for increasing values of τ. Using the nonlinear acceleration techniques provides a reduction in transport sweeps by 1 to 3 orders of magnitude, depending on the dominance ratio of the problem. However, we see that as the dominance ratio approaches one, the number of transport sweeps required by HO-JFNK and HO-NKA increases considerably. We find that the number of transport sweeps is far less problem dependent for the HOLO method, regardless of problem dominance ratio. For each τ, the number of sweeps required by the HOLO method is between 4 and 7. The HOLO represents a 2–15× speedup over the nonlinear acceleration techniques and a 25–15 000× speedup over power iteration.

We will demonstrate the same behavior in two-spatial dimensions, often to a greater extent. The HOLO methods are the most efficient for each of the problems examined. These results are displayed in Section 4.

2. Traditional eigenvalue methods

The fundamental algorithm for solving the neutron transport k-eigenvalue problem is known as “Power Iteration” or PI [13]. This method has distinct similarities to the well known “Power Method” from numerical linear algebra, but with a couple of subtle differences. The PI algorithm is described in Algorithm 1.

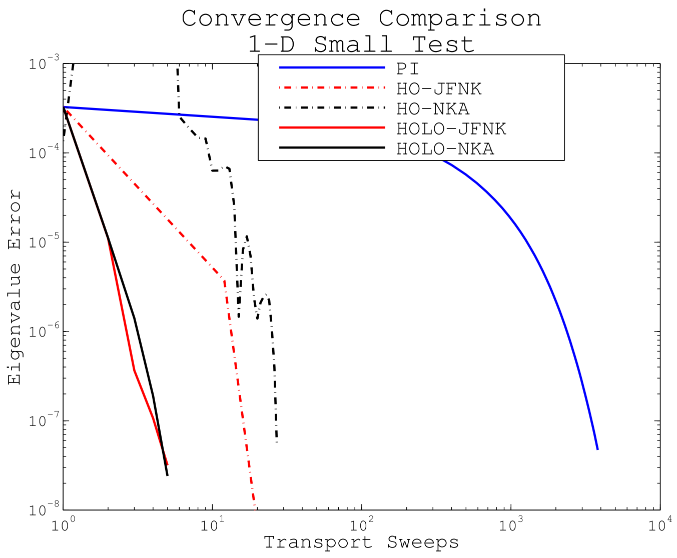

Fig. 2. Methods comparison for 1-D k-eigenvalue problem. 35 mfp domain. Dominance ratio is .8185.

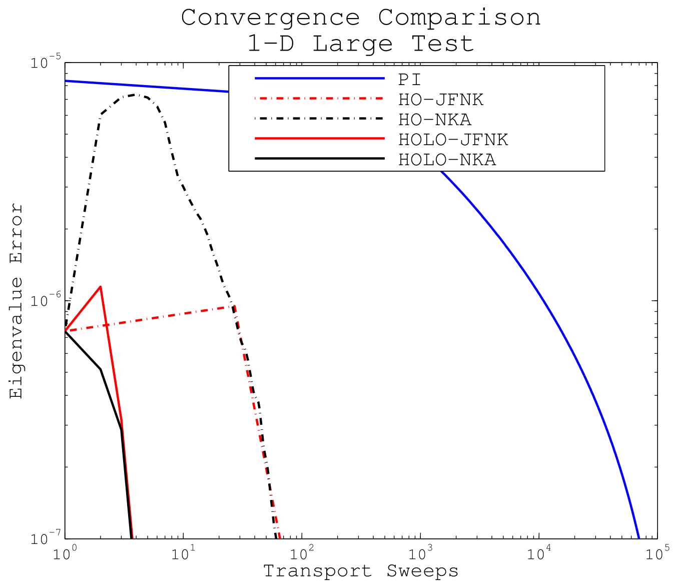

Fig. 3. Methods comparison for 1-D k-eigenvalue problem. 210 mfp domain. Dominance ratio is .99559.

Eq. (8) represents a fixed-source problem which needs to be solved at each outer iteration. If this fixed-source problem is solved exactly, a true power method iteration is achieved. However, it is generally inefficient to solve the fixed-source problem to a very tight tolerance. Instead, this problem is usually solved approximately in which the source iteration convergence tolerance is relatively loose or we only allow some finite number of source iterations per outer iteration. When this is the case, we will write PI ($k$max) to denote the maximum number of inner iteration per outer iteration.

This algorithm may converge prohibitively slowly to be of any practical use for many problems. It is well known that if the fixed-source problem is solved exactly at each outer iteration, the ratio of successive errors approaches $\rho = \frac{|k_2|}{|k_1|}$ as the iteration progresses, in which k_1 is the largest eigenvalue and k_2 is the second largest eigenvalue in magnitude [13]. This ratio, ρ, is known as the dominance ratio. As ρ approaches one, the number of power iterations required to solve the eigenvalue problem increases without bound. If the fixed-source problem is solved only approximately, the convergence rate can be significantly slower than predicted by the dominance ratio. Furthermore, we know that the scattering ratio dictates the convergence rate of each fixed-source problem. The most challenging problems are those which have high scattering and dominance ratios.

Algorithm 1 Power Iteration (PI) algorithm.

```
Power Iteration
  Input $\Phi^0$, $k^0$, outer tolerance, inner tolerance, $k$max.
  while $\frac{|k^{n+1}-k^n|}{k^{n+1}} >$ outer tolerance do
    while $\|\Phi^{n+1,k+1} - \Phi^{n+1,k}\| >$ inner tolerance and $k < k$max do
      Compute new flux via source iteration
          $\mathcal{L}\Psi^{n+1,k+1} = \frac{1}{4\pi}\mathcal{S}\Phi^{n+1,k} + \frac{1}{4\pi}\frac{1}{k_{eff}}\mathcal{F}\Phi^n$      (8)
          $\Phi^{n+1,k+1} = \mathcal{M}_0\Psi^{n+1,k+1}$      (9)
    end while
    Update Eigenvalue
        $k^{n+1} = \frac{W^T\mathcal{F}\Phi^{n+1}}{W^T\mathcal{F}\Phi^n}k^n$      (10)
    in which $W^T\vec{x}$ represents the numerical integral of $\vec{x}$ throughout the domain.
  end while
```

As the scattering ratio helps to determine the difficulty of the problem, reporting only the dominance ratio for a given problem is not always an accurate portrayal of problem difficulty. For this reason, we will define a *scattering plus dominance ratio*, ρ_{s+f}, which is defined to be the limiting ratio of subsequent errors in the eigenvalue when Power Iteration is employed using only a single transport sweep per inner iteration. That is,

$$\rho_{s+f} = \frac{|k^{n+1}_{PI(1)} - k^*|}{|k^{n}_{PI(1)} - k^*|}, \tag{11}$$

where k^* is the true eigenvalue.

Many real world application problems have both high scattering and dominance ratios. This is typical for light water reactor transport eigenvalue problems. For this reason, we need to accelerate standard power iterations. In the remainder of this section we will consider applying well-known nonlinear solvers to the eigenvalue problem. We will see that these nonlinear solvers applied to a simple reformulation of the problem can yield impressive results in Section 4.

2.1. Nonlinear acceleration of transport criticality problems

Several attempts [14,15,19] to accelerate the solution to the k-eigenvalue problem have considered rewriting the neutron transport eigenvalue problem as a system of nonlinear equations. The goal in these attempts is to obtain the quadratic convergence afforded by Newton's method. These problems are generally of the form

$$F\begin{pmatrix}\Phi\\ k\end{pmatrix} = \begin{pmatrix}F_\Phi(\Phi, k)\\ F_k(\Phi, k)\end{pmatrix} = 0 \tag{12}$$

and differ in their definitions of F_Φ and F_k.

In [14,15,19], the authors generally write

$$F_\Phi(\Phi, k) = \Phi - P(k)\Phi. \tag{13}$$

One simple definition of $P(k)$ is given by

$$P_1(k) = \mathcal{M}_0\mathcal{L}^{-1}\left[\frac{1}{4\pi}\left(\mathcal{S} + \frac{1}{k}\mathcal{F}\right)\right]. \tag{14}$$

This represents a single transport sweep in which the scattering and fission sources are computed using Φ. A true power iteration is given by

$$\Phi^{n+1} = P_2\left(k^n\right)\Phi^n \tag{15}$$

in which

$$P_2\left(k^n\right)\Phi^n = \mathcal{M}_0\left(\mathcal{L} - \frac{1}{4\pi}\mathcal{S}\mathcal{M}_0\right)^{-1}\frac{1}{4\pi k^n}\mathcal{F}\Phi^n. \tag{16}$$

We generally avoid iterations involving operators like P_2, as it involves a full inversion of the transport plus scattering operator, that is, the solution to a fixed-source transport problem. This can be exceptionally expensive, especially when the scattering ratio is close to unity. The advantage of the P_1 operator is that it only requires a single transport sweep (inversion of $\mathcal{L}$) per application.

The authors of [15] consider the following nonlinear function for k,

$$F_k(\Phi, k) = \left(1 - \frac{W^T \mathcal{F} P(k) \Phi}{W^T \mathcal{F} \Phi}\right) k, \tag{17}$$

in which W^T represent a weighted numerical integral over the spatial domain.

Given this formulation of the problem, we can apply any nonlinear solver to compute $F = 0$. The usual candidates for nonlinear solvers are Newton's method [1–3] or Nonlinear Krylov Acceleration (NKA) [14–16]. Standard Newton's method implementations are usually Jacobian-free, in which the Jacobian-vector product is computed using a traditional finite difference. The resulting linear system is solved using an appropriate Krylov method, generally GMRES (see Algorithm 2). One significant advantage of NKA, Algorithm 3, is that one never needs to compute the Jacobian or a Jacobian-vector product. With NKA, only the nonlinear function evaluation is required.

Algorithm 2 Jacobian-Free Newton–Krylov (JFNK) algorithm.

```
JFNK
  Input initial iterate $x_0$, $\eta \in (0, 1)$.
  Set $n = 0$.
  while $F(x_n) >$ tolerance do
    Compute Newton step $s$ satisfying
        $\| F'(x_n)s + F(x_n) \| < \eta \| F(x_n) \|$                (18)
    by solving
        $F'(x_n)s = -F(x_n)$                                        (19)
    with GMRES.
    Compute next iterate $x_{n+1}$
        $x_{n+1} = x_n + s.$                                        (20)
    Set $n = n + 1$.
  end while
```

Algorithm 3 Nonlinear Krylov Acceleration (NKA) algorithm.

```
NKA
  Input initial iterate $x_0$, history length $M$.
  Set $x_1 = x_0 - F(x_0)$
  Set $n = 1$.
  while $F(x_n) >$ tolerance do
    Compute NKA Update
        $\vec{v}_{n+1} = \sum_{i=n-M+1}^{n} z_i^{(n)} \vec{v}_i + \left( F(x_n) - \sum_{i=n-M+1}^{n} z_i^{(n)} \vec{w}_i \right)$    (21)
        $\vec{v}_i = x_{i-1} - x_i$                                   (22)
        $\vec{w}_i = F(x_{i-1}) - F(x_i)$                             (23)
        $\vec{z}^{(n)} = \arg\min_{y \in \mathbb{R}^M} \left\| F(x_n) - \sum_{i=n-M+1}^{m} y_i \vec{w}_i \right\|$    (24)
    and
        $x_{n+1} = x_n + v_{n+1}.$                                  (25)
    Set $n = n + 1$.
  end while
```

Before moving on, it should be noted that using this formulation of the problem, there is no guarantee that Newton's method or NKA will converge to the dominant eigenpair. This function, as it is written, has a solution for each eigenvalue/eigenvector pair. It was observed in [19] that Newton's method would often converge to a non-dominant eigenpair.

2.2. Nonlinear elimination of the eigenvalue

As with any eigenvalue problem,

$$Ax = \lambda x, \tag{26}$$

if (λ^*, x^*) is an eigenvalue–eigenvector pair, then so is (λ^*, cx^*). When numerical techniques are employed to compute the eigenvalue–eigenvector pair, we must put a constraint on the magnitude of x^* in order to obtain a unique solution. Commonly, we demand that

$$\|x^*\| = 1, \tag{27}$$

however this is purely for convenience – any sensible constraint will do.

For the neutron transport k-eigenvalue problem, we often find it convenient to place a constraint on the eigenvalue–eigenvector pair so that

$$k = \iint \nu\Sigma_f \phi \, dV \, dE \tag{28}$$

in which $\iint \cdot \, dV \, dE$ represents an integral over the spatial and energy domains. Employing this scaling factor, we can eliminate k as an input to the nonlinear function F. We can now write

$$F(\Phi) = \Phi - P\big(k(\Phi)\big)\Phi, \tag{29}$$

in which k, a function of Φ, is computed via the constraint equation,

$$k(\phi) = \iint \nu\Sigma_f \phi \, dV \, dE. \tag{30}$$

We will use this nonlinear formulation of the eigenvalue problem for all of our numerical tests (JFNK-HO or NKA-HO).

3. HOLO methods

HOLO [5] methods have been shown to be highly efficient in several different application areas including neutronics [5,6], thermal radiative transfer [22] and kinetic plasma physics [21]. These methods are occasionally referred to as “Moment-Based Acceleration” techniques and the reader should recognize these acceleration techniques as one and the same.

These methods accelerate solution convergence by alternating between solving a HO kinetic equation and a LO system of moment equations which are forced to remain discretely consistent (down to the level of truncation error) with the original system. Some view HOLO methods as (physics-based) multi-level methods: the transport sweep helps to relax the transport error modes, while the LO system helps to relax the diffusion error mode.

In neutron transport, we take angular moments of the multi-group transport equation which yields a system of multi-group diffusion-like equations which are dependent only on the spatial variable, $\vec{r}$, and energy, E. The dimensionality of this LO problem is significantly smaller and therefore, the low order system is far less expensive to solve. While there are variants on the HOLO method, such as [4], we will exclusively consider nonlinear diffusion acceleration (NDA) [5,6,8]. We derive these equations in the following section.

3.1. Deriving the NDA LO equation

As in [6–8,10], we begin by computing the zeroth angular moment of Eq. (4),

$$\nabla \cdot \vec{J}_g + \big(\Sigma_{t,g} - \Sigma_s^{g\to g}\big)\phi_g = \sum_{g' \neq g} \Sigma_s^{g'\to g}\phi_{g'} + \frac{\chi_g}{k_{eff}} \sum_{g'=1}^{G} \nu\Sigma_{f,g'}\phi_{g'}, \tag{31}$$

in which the current, $\vec{J}$, is defined as the first angular moment,

$$\vec{J}_g(\vec{r}) = \int_{4\pi} \hat{\Omega}\psi_g(\hat{\Omega}, \vec{r}) d\hat{\Omega} = \mathcal{M}_1\psi_g. \tag{32}$$

We refer to Eq. (31) as the neutron balance equation. As in [6–8,10], we use a Fick's law plus drift term closure relationship for the current,

$$\vec{J}_g = -\frac{1}{3\Sigma_{t,g}}\nabla\phi_g + \hat{D}_g\phi_g. \tag{33}$$

Substituting Eq. (33) into Eq. (31) yields the NDA LO system,

$$\nabla \cdot \left[-\frac{1}{3\Sigma_{t,g}}\nabla\phi_g + \hat{D}_g\phi_g\right] + \big(\Sigma_{t,g} - \Sigma_s^{g\to g}\big)\phi_g = \sum_{g' \neq g} \Sigma_s^{g'\to g}\phi_{g'} + \frac{\chi_g}{k_{eff}} \sum_{g'=1}^{G} \nu\Sigma_{f,g'}\phi_{g'}. \tag{34}$$

The consistency term, $\hat{D}_g$, is computed using HO quantities of the scalar flux, ϕ_g^{HO}, and current, $\vec{J}_g^{HO}$, via

$$\hat{D}_g = \frac{\vec{J}_g^{HO} + \frac{1}{3\Sigma_{t,g}} \nabla \phi_g^{HO}}{\phi_g^{HO}}. \tag{35}$$

These HO moments are computed directly via integration of the HO angular flux, which is the results of a single HO transport sweep. That is,

$$\begin{aligned}
\Psi &= \frac{1}{4\pi} \mathcal{L}^{-1} \left[\mathcal{S} + \frac{1}{k_{eff}} \mathcal{F} \right] \Phi, \\
\Phi^{HO} &= \mathcal{M}_0 \Psi \\
\vec{J}^{HO} &= \mathcal{M}_1 \Psi.
\end{aligned}$$

It is important to note that the discretization used when computing $\hat{D}$ must be consistent with the LO discretization in order to obtain discrete consistency.

We will express Eq. (34) in operator notation as well,

$$\mathcal{D}\Phi = (\mathcal{S}_U + \mathcal{S}_L)\Phi + \frac{1}{k_{eff}} \mathcal{F}\Phi, \tag{36}$$

in which

$$\begin{aligned}
\mathcal{D}_g \Phi &= \nabla \cdot \left[-\frac{1}{3\Sigma_{t,g}} \nabla + \hat{D}_g \right] \phi_g + \left(\Sigma_{t,g} - \Sigma_s^{g \to g} \right) \phi_g, \\
(\mathcal{S}_{U,g} + \mathcal{S}_{L,g})\Phi &= \sum_{g' \neq g} \Sigma_s^{g' \to g} \phi_{g'}.
\end{aligned}$$

Solving this LO system is preferable to solving the HO system in the sense that we can handle the scattering operator implicitly. When executing transport sweeps, this scattering term must be lagged at the previous iterate. Furthermore, the dimensionality of the LO problem is considerably smaller than that of the HO problem.

3.2. NDA-based eigenvalue methods

When we apply NDA to the neutron transport k-eigenvalue problem [6,8] it yields a set of algorithms in which we alternate between executing HO transport sweeps and solving the LO eigenvalue problem (Eq. (34)). These methods differ only in the way in which the LO eigenvalue problem is solved. We have described the methods formally in Algorithm 4.

Algorithm 4 NDA.

```
Nonlinear Diffusion Acceleration
  Compute initial iterate $\Phi^{(0)}$, initial eigenvalue approximation $k^0$. Set iteration counter $m = 0$.
  while $|k^m - k^{m-1}| > \tau$ do
    Update counter, $m = m + 1$.
    Execute transport sweep and compute new consistency term
      $\Psi^{(m)} = \frac{1}{4\pi} \mathcal{L}^{-1} \left( \mathcal{S} + \frac{1}{k^{m-1}} \mathcal{F} \right) \Phi^{(m-1)},$   (37)
      $\Phi^{HO} = \int \Psi^{(m)} d\hat{\Omega},$   (38)
      $\vec{J}^{HO} = \int \hat{\Omega} \Psi^{(m)} d\hat{\Omega},$   (39)
      $\hat{D}^{(m)} = \frac{\vec{J}^{HO} + \frac{1}{3\Sigma_t} \nabla \Phi^{HO}}{\Phi^{HO}}.$   (40)
    Solve the LO eigenvalue problem for $\Phi^{(m)}$ and $k^m$
      $(\mathcal{D}^{(m)} - \mathcal{S}_U - \mathcal{S}_L)\Phi^{(m)} = \frac{1}{k^{(m)}} \mathcal{F}\Phi^{(m)}.$   (41)
  end while
```

We can solve the LO eigenvalue problem using a variety of methods. If we apply a simple power iteration, we refer to the resulting method as NDA-PI. When a nonlinear solver is used to accelerate the solution to the LO problem, we refer to the method as NDA-NCA for nonlinear criticality acceleration. NDA-NCA was originally designed with the use of JFNK in mind as the LO solver (NDA-NCA-JFNK) [8,7,10], but has since been adapted to utilize either JFNK or NKA (NDA-NCA-NKA).

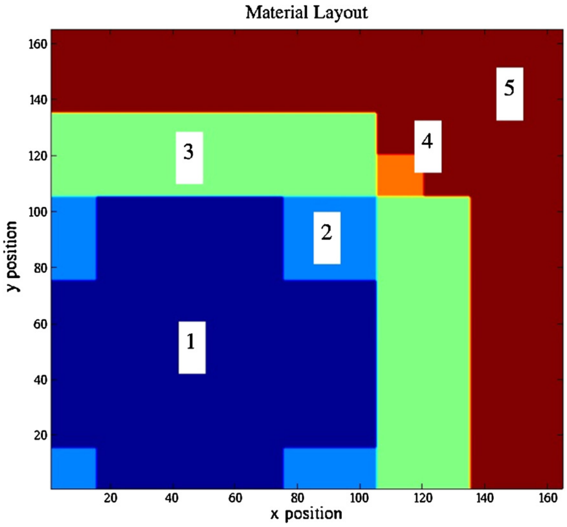

Fig. 4. LRA-BWR material layout.

When a nonlinear solver is applied, we re-write the LO problem as a nonlinear equation

$$F_\Phi(\Phi) = (\mathcal{D} - \mathcal{S}_U - \mathcal{S}_L)\Phi - \frac{1}{k(\Phi)}\mathcal{F}\Phi \quad \text{in which} \tag{42}$$

$$k(\Phi) = \sum_{g=1}^{G} \int dV \, \nu \Sigma_{f,g} \phi_g. \tag{43}$$

We now apply either JFNK or NKA to solve $F(\Phi) = 0$ in which k has been nonlinearly eliminated. For the purpose of this paper, we will not concern ourselves with details regarding the solution to the LO eigenvalue problem beyond the choice of nonlinear solvers. For a more in-depth treatment of the solution to the LO eigenvalue problem, see [8,10]. We precondition the linear solve with the inverse of the matrix

$$\mathcal{M} = \mathcal{D} - \mathcal{S}_L - \frac{1}{k^{m-1}}\mathcal{F}. \tag{44}$$

Here, k has been lagged to the previous iterate of k and the up-scattering, $\mathcal{S}_U$, operator has been ignored. Ignoring the up-scattering operator ensures that $\mathcal{M}$ is block triangular and efficient to invert.

We should also note that a multilevel NDA method has been developed for computing the dominant eigenvalue–eigenvector pair in [17]. The method we have described is a “single-level” NDA, in which the transport equation is accelerated by a multi-group diffusion-like system. In [17], a second level of acceleration is applied by reducing the multi-group diffusion equations to a single-group diffusion equation. While [17] reports positive results, we will not discuss it here as it would only serve to cloud the current discussion. Interested readers should look to [17] directly.

4. Results

In this section we will consider two well-known 2-D quarter-core problems, the LRA-BWR problem [11,12] and the C5G7MOX problem [20]. For each test, we will use an S_{16} angular quadrature. The LRA-BWR problem is a 2 group problem. For each of the following LRA-BWR test runs, we will use a 352 × 352 mesh, unless otherwise noted. On this mesh, the dominance ratio, ρ, is above 0.97, and the scattering plus dominance ratio is roughly 0.993. The C5G7MOX problem is a 7 group problem. For the C5G7MOX problem we will utilize a 357 × 357 mesh for each test run. The dominance ratio for this problem is much smaller, roughly 0.77, however the scattering plus dominance ratio is about 0.989.

The material configuration for the LRA-BWR problem is displayed in Fig. 4 and the material configuration for the C5G7MOX problem is displayed in Fig. 5.

The C5G7MOX problem contains significant upscattering within the thermal groups. This often has the impact of making the linear solve moderately more expensive, as we usually ignore upscattering in the preconditioner.

In the following four subsections, we will look at a series of studies which consider each of the methods discussed above. We will solve the HO problem with both JFNK and NKA. As the performance of both of these algorithms rely on a set of tunable parameters, we will solve each problem a series of times with different parameter sets. Finally, we will solve each problem using the HOLO algorithm using both JFNK and NKA as LO solvers. We use an implementation of NKA as described in [15] which the authors provide at http://sourceforge.net/projects/nlkain/.

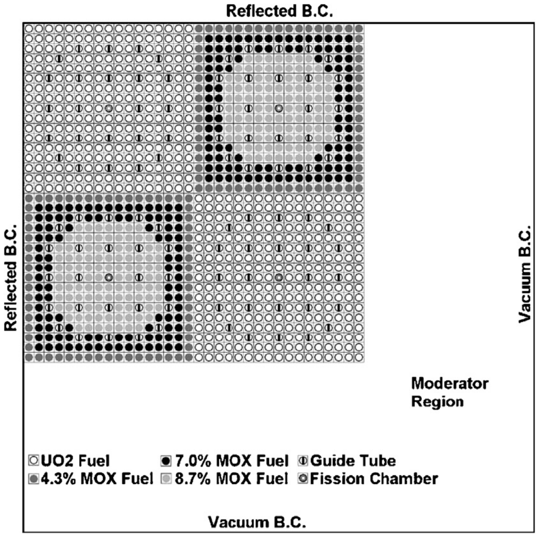

Fig. 5. C5G7MOX material layout [20].

Table 2
HO-JFNK parameter study.

η	Max Krylovs	Sweeps	
		LRA-BWR	C5G7MOX
.001	15	1631	204
.01	15	2882	174
.1	15	2016	152
.001	30	503	179
.01	30	1030	155
.1	30	453	125
.001	150	337	165
.01	150	308	154
.1	150	314	166

For each of the HOLO algorithms (NDA-NCA and NDA-PI) we compute our initial iterate by solving the diffusion eigenvalue problem. This amounts to solving the LO problem in which $\hat{D}$ in Eq. (34) is taken to be zero everywhere. This provides a better initial guess than using a flat source. For JFNK-HO and NKA-HO we begin with 5 initial power iterations. This helps to ensure the solution converges to dominant eigenpair.

4.1. JFNK-HO parameter study

The convergence of a Jacobian-Free Newton–Krylov algorithm depends both on the difficulty of the problem at hand as well as the choice of values for a few key parameters [1,2]. These parameters include the *forcing term*, η, the maximum number of vectors in the Krylov basis, and the maximum number of times GMRES is allowed to restart.

Often times the maximum number of vectors in the Krylov basis is not a choice at all. It is an upper limit defined by the memory storage requirements of keeping around the basis. When the numerical problem becomes large (in terms of degrees of freedom), we may only be able to store a few Krylov vectors. GMRES usually performs best when restarts are not required, and convergence degrades when the size of the Krylov subspace must remain small. The key to good JFNK performance with a minimal number of Krylov vectors is quality preconditioning.

In Table 2 we present results of a JFNK parameter study for both the LRA-BWR and C5G7MOX problems. We allow constant $\eta = .1$, .01 and .001 and allow the maximum number of Krylov vectors before restart to take on values of 15, 30 and 150. We allow a total of 150 GMRES iterates regardless of the basis size. In later sections, we will write HO-JFNK (η, K) to denote JFNK applied to the HO problem, Eq. (29), with forcing parameter η and a maximum storage of K Krylov vectors. We compare performance in terms of transport sweeps, as this is by far the dominant computational cost.

Here we see that the results for the LRA-BWR problem are very sensitive to the choice of the maximum number of Krylov vectors stored. This demonstrates that, as expected, the linear solve required by Newton's method can be very challenging for this problem. When only 15 Krylov vectors are stored, the linear iteration converges very slowly – this points at the need for a good preconditioner. The C5G7MOX problem is much less challenging and we see less of a dependence on the choice of parameters. The convergence is less sensitive to the η parameter.

Table 3
HO-JFNK vs. HO-NKA study.

Method	Sweeps	
	LRA-BWR	C5G7MOX
HO-JFNK(.1, 15)	2016	152
HO-JFNK(.001, 30)	503	179
HO-JFNK(.1, 30)	453	125
HO-JFNK(.01, 150)	308	154
HO-JFNK(.1, 150)	314	166
HO-NKA(15)	301	121
HO-NKA(30)	262	168

Table 4
NDA-NCA mesh study.

Mesh size	Sweeps		k_{eff}	Error	Order
	JFNK	NKA			
0.46875	5	6	1.0014486	2.541×10^{-6}	–
0.3125	5	5	1.0014499	1.025×10^{-6}	1.97
0.234375	6	5	1.0014503	5.617×10^{-7}	2.09
0.1875	6	5	1.0014505	3.598×10^{-7}	2.00
0.15625	5	5	1.0014507	2.496×10^{-7}	2.00

These results highlight two distinct issues with using Newton's method applied directly to the HO problem. First, we do not have an adequate preconditioner for the nonlinear HO problem. This makes JFNK a potentially poor choice, as the Krylov solve may consume a considerable number of transport sweeps. Secondly, we have demonstrated that it may be very challenging to pick the appropriate parameters for JFNK a priori. This can lead to a factor of 2 or 3 more work than necessary.

4.2. JFNK-HO vs. NKA-HO study

In this section we will consider JFNK and NKA applied to the HO problem. These results are presented in Table 3. For NKA, we will use the variant referred to as NKA$_1$ from Ref. [15] for all tests. This means we must only vary the number of vectors stored, m. We will present results for NKA (m) and a selection of results from Table 2.

For both the LRA-BWR and C5G7MOX problems we find that NKA is a superior choice when applied directly to the HO problem. When storage is an issue and only a few Krylov vectors can be stored, NKA is the preferable choice. This choice has less dramatic effects for the easier C5G7MOX problem, but the results for the LRA-BWR problem are clear. Unless one can afford to keep around a substantial number of Krylov vectors, NKA is the better choice. Additionally, one does not need to worry about the potentially costly choice of choosing the wrong value for η when using NKA. In addition, because NKA does not require the computation or approximation of a Jacobian, we don't need to worry about finite-difference perturbations which contain negative components.

4.3. Mesh convergence study

In Table 4, we solve the LRA-BWR problem on a series of meshes to demonstrate the independence of the number of iterations on the mesh size when using the NDA-NCA method and to demonstrate the second-order accuracy of the DD spatial discretization method. Each problem is solved to a tolerance of 10^{-7}. The error is computed against a reference solution on a refined mesh.

In Table 4 we see that NDA-NCA-JFNK and NDA-NCA-NKA perform comparably for this problem on each mesh – the difference on any mesh is at most one iteration. We see in this table how efficient NDA-NCA can be and will demonstrate these results again in Tables 6 and 8 in the following section.

In Table 5 we perform the same study using NKA applied to the HO problem. The eigenvalues obtained coincide with the data from Table 4, however we notice that vastly more sweeps are required when using HO-NKA.

Also, from the acquired data, it appears that NKA is somewhat sensitive to the size of the mesh. We see between 234 and 403 transports sweeps required to satisfy the convergence criterion on a variety of meshes. Contrast this with Table 4 in which NDA-NCA performs comparably on every mesh. This trait of NDA-NCA is highly desirable – we know that we can pick a mesh to satisfy our accuracy requirements and the algorithm performance will not be affected.

4.4. HOLO vs. HO study

In this final results section we wish to demonstrate the superior performance of the NDA-NCA and NDA-PI HOLO algorithms in direct comparison with either JFNK or NKA applied directly to the HO problem. We provide results using PI as

Table 5
HO-NKA(15) mesh study.

Mesh size	Sweeps	k_{eff}	Error	Order
0.937500	296	1.0014417	9.210×10^{-6}	–
0.468750	301	1.0014486	2.280×10^{-6}	2.01
0.312500	300	1.0014499	1.025×10^{-6}	1.97
0.234375	403	1.0014503	5.617×10^{-7}	2.09
0.187500	294	1.0014505	3.598×10^{-7}	2.00
0.156250	234	1.0014507	2.496×10^{-7}	2.00

Table 6
LRA-BWR: HO methods vs. HOLO methods.

Method	Sweeps	Time (s)	HO time (s)	LO time (s)	Factor
NDA-NCA-JFNK	5	115.04	46.04	69.00	1.00
NDA-NCA-NKA	5	140.88	46.02	94.86	1.22
NDA-PI	7	987.62	64.53	923.09	8.59
HO-JFNK(.001, 30)	503	4687.32	4687.32	–	40.75
HO-JFNK(.01, 150)	308	2893.14	2893.14	–	25.15
HO-NKA(15)	301	2768.65	2768.65	–	24.07
HO-NKA(30)	262	2410.32	2410.32	–	20.95
PI(1)	9754	89 795.33	89 795.33	–	780.56
PI(10)	12 501	115 084.21	115 084.21	–	1000.38

Table 7
LRA-BWR: HOLO methods cost breakdown.

Method	Sweeps	LO func. evals.	Precond. const.	Precond. app.
NDA-NCA-JFNK	5	177	15	147
NDA-NCA-NKA	5	187	172	172
NDA-PI	7	1920	–	–

Table 8
C5G7MOX: HO methods vs. HOLO methods.

Method	Sweeps	Time (s)	HO time (s)	LO time (s)	Factor
NDA-NCA-JFNK	6	535.48	214.92	320.56	1.00
NDA-NCA-NKA	6	635.52	214.71	420.81	1.19
NDA-PI	13	2599.61	469.37	2130.24	4.85
HO-JFNK(.001, 30)	179	6512.52	6512.52	–	12.16
HO-JFNK(.01, 150)	154	5620.18	5620.18	–	10.50
HO-NKA(15)	121	4500.79	4500.79	–	8.41
HO-NKA(30)	168	6180.89	6180.89	–	11.54
PI(1)	1454	56 387.71	56 387.71	–	105.30
PI(10)	1970	72 851.56	72 851.56	–	136.05

a baseline comparison. The results are definitive for both the LRA-BWR and C5G7MOX problems as we saw in the earlier 1-D motivating problem. Here, we present results in terms of compute time, in addition to transport sweeps, as the LO solve time is small, but not negligible. In the HOLO approach the method is very effective at reducing transport sweeps and thus the LO compute time can dominate.

In Tables 6 and 8 we see that at worst, NDA-NCA is an 8× improvement over HO-JFNK or HO-NKA and at times over a 40× improvement. This improvement is the direct result of moving a large portion of the computational work to the LO space. While we do incur some cost in solving the LO eigenvalue problems, this cost is completely offset by reducing the number of transport sweeps by a factor of 20–100 times. We only compare the NDA-NCA results against a selection of the best-performing HO-JFNK and HO-NKA results in these tables. It is also important to note that within the HOLO context, the choice of which LO nonlinear solver to use is insignificant. Both JFNK and NKA perform well. In the case of NDA-NCA, we have an excellent preconditioner for the LO nonlinear problem. This allows us to keep the number of Krylov vectors small. For the LRA-BWR problem, we see between 9 and 16 Krylov vectors per Newton step and between 10 and 12 vectors for the C5G7MOX problem. Contrast with the 150 GMRES iterations often required by HO-JFNK. We find that the C5G7MOX problem is a bit more expensive than the LRA-BWR problem to solve for roughly the same number of spatial cells. This is entirely attributed to the difference in the number of energy groups and the presence of upscattering.

While HO-JFNK, HO-NKA and NDA-NCA accelerate the solution to the eigenvalue problem, they do so in dramatically different ways. HO-JFNK and HO-NKA accelerate both the fission and scattering sources within the HO problem. Either Newton's method or NKA is used to provide this acceleration. In the case of NDA-NCA, both the scattering and fission sources

Table 9
C5G7MOX: HOLO methods cost breakdown.

Method	Sweeps	LO func. evals.	Precond. const.	Precond. app.
NDA-NCA-JFNK	6	282	16	250
NDA-NCA-NKA	6	244	228	228
NDA-PI	13	2139	–	–

are accelerated via the inclusion of the LO system. The LO system (NDA) allows for the implicit treatment of the scattering source. In this case, either Newton's method or NKA is used to accelerate the solution to the LO eigenvalue problem, hence accelerating the fission source. Each LO function evaluation is considerably less expensive than a HO function evaluation, as each HO function evaluation contains a transport sweep, which is often the most considerable cost in solving the neutron transport eigenvalue problem.

It is important to note that there is a small, yet noticeable, difference in the amount of time required to solve the LO problem by JFNK and NKA. This difference can be partially attributed to the difference in the number of function evaluations required to solve the LO problem. This difference does not imply JFNK is necessarily more efficient than NKA. When using Newton's method, we generally terminate the LO solve after a single Newton step. It is difficult to define an equivalent amount of work for NKA.

Furthermore, in Tables 7 and 9 we present the number of LO function evaluations, preconditioner constructions and preconditioner applications. One of the other reasons in which JFNK is a more efficient LO solver is that it only requires one preconditioner construction per Newton step. On the other hand, NKA requires one preconditioner construction (and application) for each LO function evaluation until the convergence criteria has been satisfied. In Table 9 we see that NKA actually required fewer function evaluations, but many more matrix constructions. These matrix constructions can be a significant cost for the seven group eigenvalue problem.

5. Conclusion

In this paper we have compared a variety of methods for solving the neutron transport k-eigenvalue problem. We presented results from [9,18] which demonstrate that the eigenvalue problem can be rewritten as a nonlinear function which can be solved using JFNK or NKA. Our results support the conclusions in [15] that NKA is generally more efficient at solving this nonlinear formulation of the HO problem. Subsequently, we described the NDA-NCA HOLO method for solving the k-eigenvalue problem which was originally described in [8]. This method uses a sequence of LO diffusion-like eigenvalue problems to accelerate the solution to the original HO transport problem. By moving a large portion of the work to this LO computational space, we can reduce the overall work considerably.

While both JFNK and NKA applied directly to the HO problem can accelerate the solution dramatically compared to a standard power iteration, they are significantly more expensive when compared to the NDA-NCA algorithm. The NDA-NCA algorithm performance is independent of mesh and is less sensitive to choices of parameters in either JFNK or NKA. Within NDA-NCA we can easily precondition the solution to the LO problem which sets it even further apart from the accelerated HO methods. For both the C5G7MOX and LRA-BWR test problems, we find that the NDA-NCA algorithm is always the most efficient method, often by a factor of 10–40 times.

References

[1] C.T. Kelley, Iterative Methods for Linear and Nonlinear Equations, Society of Industrial and Applied Mathematics, Philadelphia, 1995.
[2] C.T. Kelley, Solving Nonlinear Equations with Newton's Method, Society of Industrial and Applied Mathematics, Philadelphia, 2003.
[3] D.A. Knoll, D.E. Keyes, Jacobian-free Newton–Krylov methods: a survey of approaches and applications, J. Comput. Phys. 193 (2004) 357–397.
[4] R.E. Alcouffe, Diffusion synthetic acceleration methods for the diamond difference discrete ordinates equations, Nucl. Sci. Eng. 64 (1977) 344–355.
[5] K.S. Smith, J.D. Rhodes III, Full-core, 2-D, LWR core calculations 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, October 7–10, 2002, American Nuclear Society, 2002.
[6] D.A. Knoll, Kord Smith, H. Park, Application of the Jacobian-Free Newton–Krylov method to nonlinear acceleration of transport source iteration in slab geometry, Nucl. Sci. Eng. 167 (2) (February 2011) 122–132.
[7] Jeffrey Willert, Hybrid deterministic/Monte Carlo methods for solving the neutron transport equation and k-eigenvalue problem, PhD thesis, North Carolina State University, Raleigh, North Carolina, 2013.
[8] H. Park, D.A. Knoll, C.K. Newman, Nonlinear acceleration of transport criticality problems, Nucl. Sci. Eng. 172 (2012) 52–65.
[9] D.A. Knoll, H. Park, C. Newman, Acceleration of k-eigenvalue/criticality calculations using the Jacobian-free Newton–Krylov method, Nucl. Sci. Eng. 167 (2011) 133–140.
[10] Jeffrey Willert, C.T. Kelley, Efficient solutions to the NDA-NCA LO eigenvalue problem, in: Proceedings of the International Conference on Mathematics and Computational Methods Applied to Nuclear Science & Engineering, 2013.
[11] K.S. Smith, An analytic nodal method for solving the two-group, multidimensional static and transient neutron diffusion equations, Masters thesis, Massachusetts Institute of Technology, 1979.
[12] Argonne National Laboratory, Argonne Code Center: benchmark problem book, Tech. Rep. ANL-7416, prepared by the Computational Benchmark Problems Committee of the Mathematics and Computational Division of the American Nuclear Society, Supplement 2, 1977.
[13] E.E. Lewis, W.F. Miller, Computational Methods of Neutron Transport, American Nuclear Society, Inc., La Grange Park, 1993.
[14] Erin D. Fichtl, James S. Warsa, Matthew T. Calef, Nonlinear acceleration of S_n transport calculations, in: Proceedings of the International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering, 2011.

[15] Matthew T. Calef, Erin D. Fichtl, James S. Warsa, Markus Berndt, Neil N. Carlson, Nonlinear Krylov acceleration applied to a discrete ordinates formulation of the *k*-eigenvalue problem, J. Comput. Phys. 238 (2013) 188–209.

[16] Neil N. Carlson, Keith Miller, Design and application of a gradient-weighted moving finite element code I: in one dimension, SIAM J. Sci. Comput. 19 (1998) 728–765.

[17] Dmitriy Y. Anistratov, Multilevel NDA methods for solving multigroup eigenvalue neutron transport problems, Nucl. Sci. Eng. 174 (2012) 150–162.

[18] Daniel F. Gill, Yousry Y. Azmy, Newton's method for solving *k*-eigenvalue problems in neutron diffusion theory, Nucl. Sci. Eng. 167 (2011) 141–153.

[19] Daniel F. Gill, Yousry Y. Azmy, James S. Warsa, J.D. Densmore, Newton's method for the computation of *k*-eigenvalues in S_n transport applications, Nucl. Sci. Eng. 168 (2011) 37–58.

[20] M. Smith, E. Lewis, B. Na, Benchmark on deterministic 2-D MOX fuel assembly transport calculations without spatial homogenization, Prog. Nucl. Energy 45 (2–4) (2004) 107.

[21] William T. Taitano, Dana A. Knoll, Luis Chacón, Guangye Chen, Development of a consistent and stable fully implicit moment method for Vlasov–Ampère particle in cell (PIC) system, SIAM J. Sci. Comput. 35 (5) (2013) 126–149.

[22] H. Park, D.A. Knoll, R.M. Rauenzahn, A.B. Wollaber, J.D. Densmore, A consistent, moment-based, multiscale solution approach for thermal radiative transfer problems, Transp. Theory Stat. Phys. 41 (3–4) (2012) 284–303.