

APPLICATION OF THE JACOBIAN-FREE NEWTON-KRYLOV METHOD IN COMPUTATIONAL REACTOR PHYSICS

D.A. Knoll and H. Park
Multiphysics Methods Group
Idaho National Laboratory
Idaho Falls, ID 83415
dana.knoll@inl.gov; ryosuke.park@inl.gov

Kord Smith
Studsvik Scandpower
Idaho Falls, ID 83404
kord.smith@studsvik.com

ABSTRACT

The combination of the Jacobian-free Newton-Krylov method and physics-based preconditioning are finding increased use as a robust, accurate, and efficient algorithm for multiphysics problems. In this study we explore two creative uses for this algorithmic idea within the context of reactor physics. We consider the problems of nonlinear acceleration of the transport equation and the calculation of the dominant eigenvalue.

Key Words: Newton-Krylov Methods, nonlinear accelerators, eigenvalue, power iteration

1. Introduction

The Jacobian-Free Newton-Krylov (JFNK) method is a synergistic combination of Newton's method for nonlinear iteration and Krylov-based linear iterative methods [1]. The algorithm is growing in application space in general computational science and engineering. In this paper we give a brief overview of the method and then focus on new uses for the algorithm which are connected to computational reactor physics. In the application areas considered, we make use of two unique features of the JFNK algorithm: 1) its ability to perform a Newton iteration on a complicated multiscale function where forming the Jacobian is not an option, and 2) its ability to use another solver for the same problem as a preconditioner.

We consider two application areas. First, we consider using JFNK to execute a Newton iteration for nonlinear diffusion acceleration. This nonlinear iteration is currently done with a fixed point iteration. This will demonstrate performing a Newton iteration on a sophisticated, multiscale, function. Second, we will demonstrate the ability to bring the standard k -eigenvalue iteration inside the JFNK iteration. Here, k will become part of the Krylov solution vector (in addition to the flux) and a normalized eigenvalue update can be used as the additional required equation. We will consider the power iteration as a preconditioner.

2. The Jacobian-Free Newton Krylov Method

2.1. The Algorithm

The traditional Newtons method solves the problem

$$\mathbf{F}(\phi) = 0, \quad (1)$$

with the iteration

$$\mathbf{J}^k \delta\phi^k = -\mathbf{F}(\phi^k), \quad (2)$$

$$\phi^{k+1} = \phi^k + d * \delta\phi^k. \quad (3)$$

When using a Krylov method (such as GMRES [2]) to solve the linear problem on each Newton iteration the Jacobian, \mathbf{J} , need not be formed. Here only the action of the Jacobian times a vector is required. This product can be adequately approximated by a first-order Taylor series expansion,

$$\mathbf{J}\mathbf{v} \approx \frac{\mathbf{F}(\phi + \epsilon\mathbf{v}) - \mathbf{F}(\phi)}{\epsilon}, \quad (4)$$

where ϵ is a perturbation parameter. Discussion on the evaluation of ϵ can be found in [1].

An inexact Newton iteration [3] is typically employed with JFNK. Here, on any given Newton iteration, the convergence criteria for GMRES is proportional to the current Newton residual. This limits the linear solver work per Newton iteration, until a region of strong nonlinear convergence is achieved.

2.2. Preconditioning

The key to the efficient application of the JFNK method in typical computational science and engineering problems is preconditioning. The goal of preconditioning is to cluster eigenvalues and thus limit Krylov iterations per Newton iteration. Right preconditioning of the linear Newton problem is expressed as,

$$\mathbf{J}^k \mathbf{M}^{-1} \mathbf{M} \delta\phi^k = -\mathbf{F}(\phi^k). \quad (5)$$

Here, \mathbf{M} is the preconditioning operator, and \mathbf{M}^{-1} is the preconditioning process. When implementing this in JFNK, the matrix-vector multiply with right preconditioning becomes,

$$\mathbf{J}\mathbf{M}^{-1}\mathbf{v} \approx \frac{\mathbf{F}(\phi + \epsilon\mathbf{M}^{-1}\mathbf{v}) - \mathbf{F}(\phi)}{\epsilon}. \quad (6)$$

This process is actually done in two steps;

1. Perform $\mathbf{y} = \mathbf{M}^{-1}\mathbf{v}$
2. Perform $\mathbf{J}\mathbf{y} \approx [\mathbf{F}(\phi + \epsilon\mathbf{y}) - \mathbf{F}(\phi)] / \epsilon$.

In the concept of physics-based preconditioning many options are available for \mathbf{M}^{-1} , as will be discussed.

3. Applications

3.1. Nonlinear Acceleration

Consider the 1-D mono-energetic transport equation in homogeneous slab geometry with isotropic scattering and a fixed source

$$\mu \frac{\partial}{\partial x} \psi(x, \mu) + \Sigma_t \psi(x, \mu) = \frac{1}{2} (\Sigma_s \phi(x) + Q), \quad (7)$$

$$-1 \leq \mu \leq 1, \quad 0 \leq x \leq L, \quad (8)$$

$$\phi(x) = \int_{-1}^1 \psi(x, \mu) d\mu, \quad (9)$$

$$J(x) = \int_{-1}^1 \mu \psi(x, \mu) d\mu. \quad (10)$$

All nonlinear acceleration methods utilize the zeroth moment of the transport equation in angle which is a simple balance statement,

$$\frac{dJ}{dx} + (\Sigma_t - \Sigma_s) \phi = Q. \quad (11)$$

The acceleration methods vary in their closure for J and whether they are solving for a correction to the scalar flux or a new solution to the scalar flux. The successful nonlinear acceleration used in the CASMO code [4] uses an intuitive combination of standard diffusion plus a drift term to express current,

$$J = -\frac{1}{3\Sigma_t} \frac{d\phi}{dx} + \hat{D}\phi. \quad (12)$$

In this case the drift term is a function of the high-order (HO) solution ($\hat{D} = f(\psi)$) and the acceleration is nonlinear. Thus in CASMO,

$$\frac{d}{dx} \left[-\frac{1}{3\Sigma_t} \frac{d\phi}{dx} + \hat{D}\phi \right] + (\Sigma_t - \Sigma_s) \phi = Q, \quad (13)$$

is the low-order (LO) problem (with an appropriately defined \hat{D}), and this solution is used to accelerate the solution of the HO problem (typically on a finer mesh). We refer to this accelerator as NDA. The boundary conditions for this approach are constructed to ensure consistency between the scalar flux from the LO problem and the scalar flux from the HO problem.

The standard use of nonlinear acceleration is cast as a Picard (or fixed point) iteration. What is meant here is that the LO problem is solved for the scalar flux, while holding fixed the coefficients which are a function of the HO problem. The HO problem is solved separately for the angular flux, while holding the scattering source fixed from the LO problem. The iteration process (without spatial homogenization) is:

- One transport sweep with known scalar flux (ϕ^k) to compute $\psi^{k+\frac{1}{2}}$

- Compute $\hat{D}^{k+\frac{1}{2}}$ and boundary conditions using HO solution
- Solve LO problem for ϕ^{k+1} , using Eq. 12 with $\hat{D}^{k+\frac{1}{2}}$
- Back to transport sweep with ϕ^{k+1} (updated source)

This fixed point iteration will result in linear convergence rates. By removing this fixed point iteration and replacing it with a Newton based iteration we may produce a more robust accelerator. To achieve this goal we use the NDA LO operator as the nonlinear function with \hat{D} at the same iteration level as ϕ ,

$$F(\phi^{n+1}) = \frac{d}{dx} \left[-\frac{1}{3\Sigma_t} \frac{d\phi^{n+1}}{dx} + \hat{D}^{n+1}\phi^{n+1} \right] + (\Sigma_t - \Sigma_s)\phi^{n+1} - Q. \quad (14)$$

The algorithm achieves a consistent iteration level between \hat{D} and ϕ through executing a transport sweep within the function evaluation. We will refer to this method as JFNK-NDA. It is important to note that each Krylov iteration will require a transport sweep, and there will be a few Krylov iterations per Newton iteration.

This nonlinear method has a direct connection to the linear method discussed in [5]. In [5] a Krylov iterative method is used to treat the scattering implicitly and thus remove the source iteration. The scalar flux is the dependent variable in the Krylov iteration. To achieve this the transformation between angular flux and scalar flux and the transport sweep become a part of a sophisticated matrix-vector multiply required in the Krylov algorithm. We achieve this same result by having a transport sweep inside of our nonlinear function evaluation.

As with any JFNK implementation, the key to efficiency is effective preconditioning. In this application we will employ the NDA operator with \hat{D} evaluated at the previous nonlinear iteration as our preconditioning operator, M . For this simple one dimensional problem we will use a direct inverse for M^{-1} . As we move to multidimensional problems we will use multigrid methods.

The Quasi-diffusion method [6] uses the first moment of the transport equation,

$$\frac{dE\phi}{dx} + \Sigma_t J = 0, \quad (15)$$

with the Eddington tensor, E , defined as

$$E(x) = \frac{\int_{-1}^1 \mu^2 \psi(x, \mu) d\mu}{\int_{-1}^1 \psi(x, \mu) d\mu}. \quad (16)$$

Then the LO problem becomes,

$$\frac{d}{dx} \left(-\frac{1}{\Sigma_t} \frac{dE\phi}{dx} \right) + (\Sigma_t - \Sigma_s)\phi = Q. \quad (17)$$

The boundary conditions for this problem are formulated via integration of the discrete boundary conditions for the HO problem. An identical fixed point iteration is used between the LO problem

and the HO problem. As with NDA. Also, as with NDA, one can use the JFNK method to remove the fixed point iteration in QD. We will not do that in this study.

Typical use of NDA in LWR calculation involves homogenization [4]. Here the HO problem is solved on a fine spatial mesh resolving structure inside a pin cell, while the LO problem is solved on a mesh defined by the pin cell. In order to achieve this homogenization one requires a prescription for transferring information between meshes. Many options are available and we will only consider the standard approach in the current study [4].

3.2. k -eigenvalue iteration

Consider the one group, diffusion, 1-D criticality problem,

$$\frac{d}{dx} \left[-\frac{1}{3\Sigma_t} \frac{d\phi}{dx} \right] + \Sigma_a \phi = \frac{\nu}{k} \Sigma_f \phi. \quad (18)$$

The standard power iteration (PI) solves a fixed source problem at iteration $n+1$

$$\frac{d}{dx} \left[-\frac{1}{3\Sigma_t} \frac{d\phi^{n+1}}{dx} \right] + \Sigma_a \phi^{n+1} = \frac{\nu}{k^n} \Sigma_f \phi^n, \quad (19)$$

and then uses the updated solution to approximate the new value for k .

$$k^{n+1} = \frac{k^n \sum_{grid} (\nu \Sigma_f \phi^{n+1})}{\sum_{grid} (\nu \Sigma_f \phi^n)}. \quad (20)$$

This iteration is slow to converge in the limit of high dominance ratio. We will demonstrate the acceleration of the PI by using JFNK.

In our fully implicit JFNK implementation we will solve the nonlinear system $[F_\phi, F_k]^T = 0$, with:

$$F_\phi = \frac{d}{dx} \left[-\frac{1}{3\Sigma_t} \frac{d\phi^{n+1}}{dx} \right] + \Sigma_a \phi^{n+1} - \frac{\nu}{k^{n+1}} \Sigma_f \phi^{n+1}, \quad (21)$$

and

$$F_k = k^{n+1} - \sum (\nu \Sigma_f \phi^{n+1}). \quad (22)$$

which has used the normalization

$$\frac{k^n}{\sum (\nu \Sigma_f \phi^n)} = 1. \quad (23)$$

The unknowns in the Krylov vector will be the fluxes at each cell, ϕ_i , and the eigenvalue, k . The Newton iteration system is

$$\begin{bmatrix} J_{\phi,\phi} & J_{\phi,k} \\ J_{k,\phi} & J_{k,k} \end{bmatrix} \begin{bmatrix} \delta\phi \\ \delta k \end{bmatrix} = - \begin{bmatrix} F_\phi \\ F_k \end{bmatrix}. \quad (24)$$

The dimension of $\delta\phi$ is the dimension of the grid (times number of groups for MG), and δk is a scalar.

We will use PI as preconditioner. The preconditioner is lower block triangular

$$M = \begin{bmatrix} M_{\phi,\phi} & 0 \\ M_{k,\phi} & M_{k,k} \end{bmatrix} \quad (25)$$

with $M_{\phi,\phi} = \frac{d}{dx} \left[-\frac{1}{3\Sigma_t} \frac{d}{dx} \right] + \Sigma_a$, $M_{k,\phi} = -\sum \nu \Sigma_f$, $M_{k,k} = 1$. In our problem the preconditioning process is equivalent to one iteration of PI in "delta" or residual form. The preconditioner inverse then has two steps

1. $\delta\phi = M_{\phi,\phi}^{-1}(-F_\phi)$ (flux solve)
2. $\delta k = M_{k,k}^{-1}(-F_k - M_{k,\phi}\delta\phi)$ (eigenvalue update)

A proof of principle of the role of JFNK for this problem can be found in [7]. Here we bring in the important aspect of using the PI as the preconditioner. Using Newton's method to find the dominant eigenvalue of a steady-state problem does have some drawbacks. However a few PI to provide an excellent initial guess. In this work, we use a few PIs to seed our JFNK solver.

4. Numerical Results

4.1. Nonlinear Acceleration

We consider a problem with $L = 1, 5, 10, 20$, $\Sigma_s = 9.9$, $\Sigma_t = 10$, $Q = 0.5$. For the first set of results the LO and the HO problem are solved on a grid of 50 cells, with no spatial homogenization. The HO problem is solved with S_8 considering both diamond differencing (DD) and step characteristics (SC). For these problems the optical depth (OD) = $\Delta x \Sigma_t = 0.2L$. We consider three methods, NDA, QD, and JFNK-NDA. It can be shown that for either DD or SC, NDA will stall with increasing OD. QD will find an efficient solution, but the scalar flux from the LO problem is not the same as the scalar flux from the HO problem. The JFNK-NDA method will also find an efficient solution and the scalar flux from the LO and HO problems will be equivalent.

Figures 1 - 3 are results using diamond differencing, with flux-ratio being a relative error. Iteration convergence is plotted in terms of transport sweeps to put all methods on an equal footing. We can see that the QD method is not sensitive to OD here, but does produce a LO scalar flux which is not the same as the HO scalar flux. JFNK-NDA can improve the convergence of NDA with increasing OD, while maintaining a consistency between the LO and HO scalar flux. Moving to SC discretization can be shown to produce the same basic trends although at a somewhat higher OD. Figure 4 demonstrates this behavior.

Figures 5 and 6 demonstrate some results using homogenization. Here the coarse grid contains $nx_c = 50$ cells, and the fine grid, nx_f , can range from 50 to 500 cells. Two things are clear, there is a small price to pay for homogenization, which may be removed with a different grid interpolation, and the coarse grid LO scalar flux is identical to the HO scalar flux even with the grid mismatch.

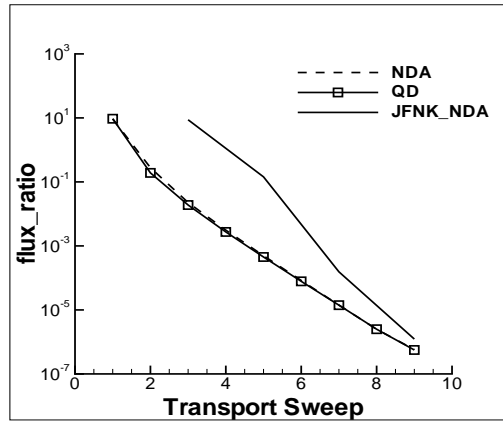


Figure 1. Convergence at OD = 0.2

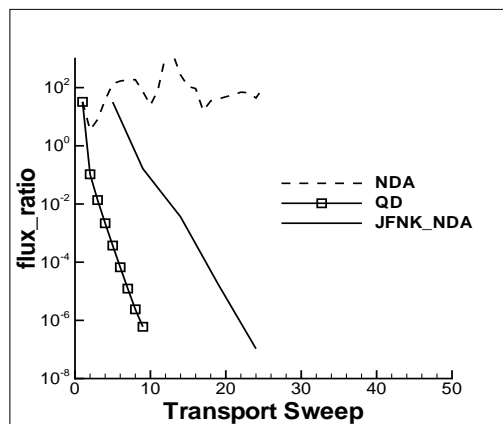


Figure 2. Convergence at OD = 2.0

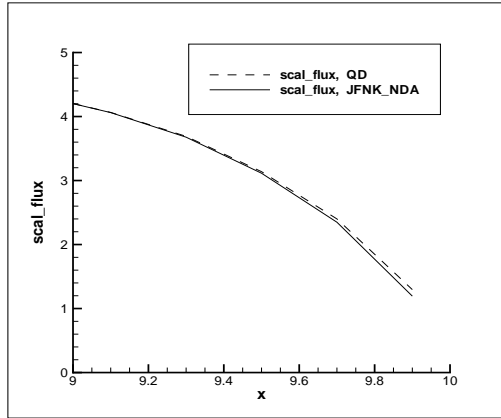


Figure 3. LO scalar Flux for QD and JFNK-NDA

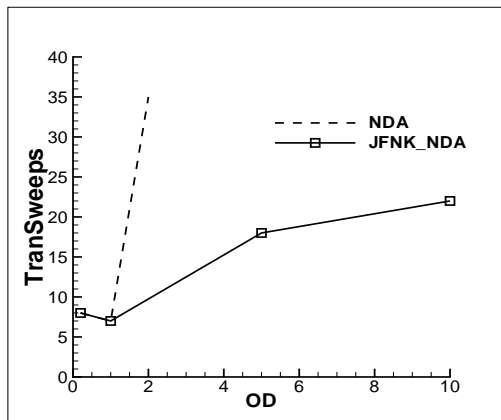


Figure 4. Convergence as a function of OD for SC discretization, NDA and JFNK-NDA

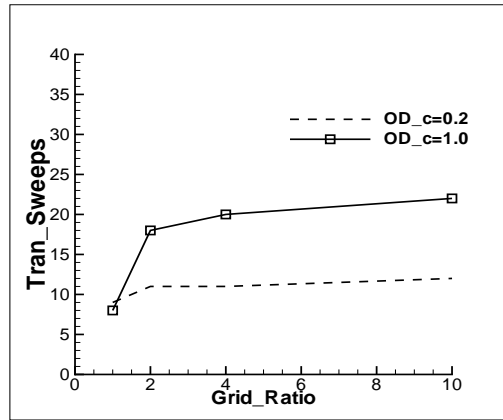


Figure 5. Convergence at OD = 0.2 and 1.0 as a function of grid ratio

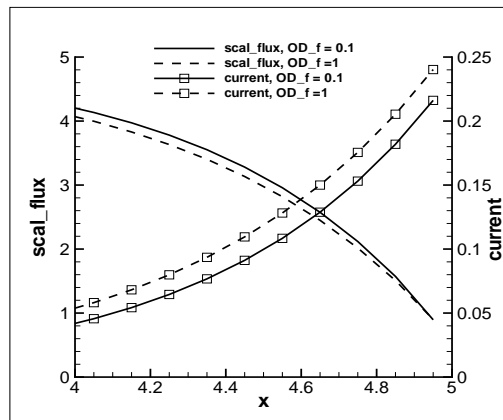


Figure 6. LO scalar flux and current with two different HO grids

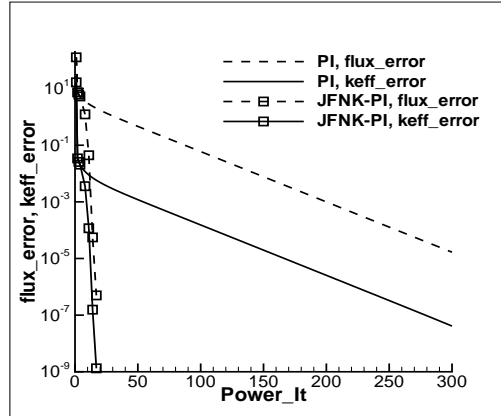


Figure 7. Convergence at $L = 15$

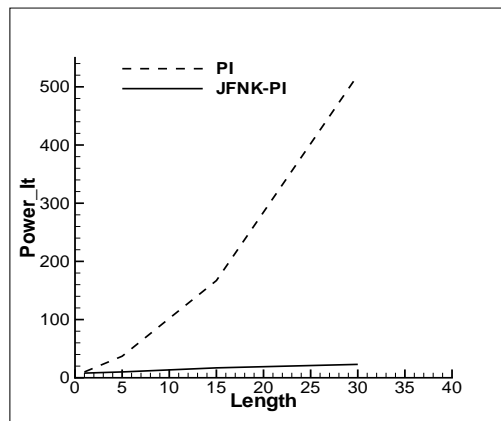


Figure 8. Convergence as a function of L

4.2. k -eigenvalue iteration

For this study we used a fixed set of parameters and then varied the domain size in order to vary the dominance ratio. The following parameters were used for a test problem.

$\Sigma_t = 1.67$, $\Sigma_a = 1.6$, $\nu\Sigma_f = 3.2$ with the dominance ratio set by the domain length,

$L = 1, 5, 15, 30$. The dominance ratio will increase with L , and thus more PIs will be required. 4 Power iterations were used to seed JFNK-PI, and the Newton updates are damped so as to not allow negative flux. Figure 7 presents detailed convergence data for $L = 15$, and Figure 8 plots PIs required for convergence as a function of L . The superiority of JFNK-PI is clear. The key here to fair comparison is to compare methods in terms of required PIs not to compare Newton iterations to PIs.

5. Conclusion and Future Work

We have presented preliminary results from the application of the JFNK method to two important problems in computational reactor physics, nonlinear diffusion acceleration of source iteration and k -eigenvalue iteration. The positive results provide motivation to continue this line of investigation on increasingly more complex problems.

Acknowledgement

This work has been carried out for the U.S. Department of Energy Office of Nuclear Energy Under DOE Idaho Operations Office Contract DE-AC07-05ID14517 (INL/CN-08-14429).

REFERENCES

- [1] D. A. Knoll and D. E. Keyes. Jacobian-Free Newton Krylov methods: A survey of approaches and applications. *J. Comput. Phys.*, 193:357–397, 2004.
- [2] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving non-symmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7:856, 1986.
- [3] S. C. Eisenstat and H. F. Walker. Choosing the forcing terms in an inexact Newton method. *SIAM J. Sci. Comput.*, 17:16–32, 1996.
- [4] K.S. Smith and III J.D. Rhodes. Full-core, 2-D LWR core calculations with CASMO-4E. *PHYSOR 2002, Seoul, Korea, 2002*.
- [5] James S. Warsa, Todd Wareing, and Jim Morel. Krylov iterative methods and the degraded effectiveness of diffusion synthetic acceleration for multidimensional S_n calculations in problems with material discontinuities. *Nuc. Sci. Eng.*, 147:218–248, 2004.
- [6] V. Ya. Gol'din. A quasi-diffusion method for solving the kinetic equation. *USSR Comp. Math. and Math. Phys.*, 4:136, 1967.
- [7] Vijay Mahadevan and Jean Regusa. Novel hybrid scheme to compute several dominant eigenmodes for reactor analysis problems. *PHYSOR 2008, Interlaken, Switzerland, 2008*.