Use of Generalized Davidson Eigenvalue Solver for Coarse Mesh Finite Difference Acceleration

Benjamin Collins, Steven Hamilton, Shane Stimpson

Oak Ridge National Laboratory, 1 Bethel Valley Rd., PO Box 2008 MS-6172, Oak Ridge, TN 37831
collinsbs@ornl.gov, hamiltonsp@ornl.gov, stimpsonsg@ornl.gov

Abstract - This paper summarizes improvements in the coarse mesh finite difference (CMFD) performance by focusing on the eigenvalue solution methodology. The current solution technique uses a shifted power iteration with the PETSc solver package. A generalized Davidson methodology has been implemented into MPACT using the Trilinos solver package. The new generalized Davidson solver shows significant improvements in the overall runtime for the CMFD acceleration scheme and the overall MPACT performance. The performance improves between 5 and 27 times for the CMFD runtime. This results in a speedup of between 2.5 to 5 times on the overall solution time for MPACT.

I. INTRODUCTION

In recent years, development of VERA-CS—the core simulator for the Consortium for the Advanced Simulation of Light Water Reactors (CASL) [1]—has focused on simulating multiple cycles of operating commercial nuclear power plants. Now that the VERA-CS capabilities have advanced to the point where they are being deployed to users, attention is now directed on improving the computational performance of various components in VERA-CS. The focus of this work is the coarse mesh finite difference [2,3] (CMFD) solution in MPACT [4]. CMFD serves multiple purposes in the 2D/1D solution methodology. First, it is a natural mechanism to tie together the 2-D radial method of characteristics (MOC) transport and the 1D axial NEM-P_3 solution. Because the CMFD system solves the multigroup three-dimensional core in one system, it pulls together the global response of the system. In addition, the CMFD solution provides a framework to accelerate the convergence of the eigenvalue problem.

The CMFD methodology is based on a finite-difference approximation of the zeroth angular-moment of the neutron transport equation homogenized onto a 3D Cartesian grid:

$$\sum_{\substack{f\in n,s,e,\\ w,t,b}} J_{g,f}\bullet A_f + \bar{\Sigma}_{t,g,i}\bar{\phi}_{g,i}V_i = \sum_{g'}\bar{\Sigma}_{s,g'\to g,i}\bar{\phi}_{g',i}V_i + \frac{\bar{\chi}_{g,i}}{k_{eff}}\sum_{g'}\nu\bar{\Sigma}_{f,g',i}\bar{\phi}_{g',i}V_i \qquad (1)$$

The CMFD methodology creates a finite-difference relationship between the surface current and flux as follows:

$$J_s = -\frac{2D^+D^-}{\Delta\left(D^+ + D^-\right)}\left(\bar{\phi}^+ - \bar{\phi}^-\right) + \hat{D}_s\left(\bar{\phi}^+ + \bar{\phi}^-\right), \qquad (2)$$

where + and - denote the cells on each side of any given surface s and Δ defines the distance between the two cells. This equation defines the relationship between the current and the cell average flux using a coarse mesh diffusion approximation; however, an additional term is added to correct errors in the approximation. The solution marching scheme used will approximate this correction coefficient using the transport solution (high-order MOC in the radial direction and the NEM-P_3 in the axial direction) for the current iteration:

$$\hat{D}_s = \frac{J_s^{trans} + \frac{2D^+D^-}{\Delta\left(D^+ + D^-\right)}\left(\bar{\phi}^+ - \bar{\phi}^-\right)}{\left(\bar{\phi}^+ + \bar{\phi}^-\right)}\,. \qquad (3)$$

The iteration scheme in MPACT first assumes $\hat{D}_s$ is zero and solves the full-core CMFD equations. Once the coarse mesh fluxes are obtained, they are projected onto the fine MOC mesh and a single MOC sweep is performed. The MOC equations solve for the current on the boundaries of the coarse mesh, which are used to estimate a new value for $\hat{D}_s$ in the radial direction. The coarse cells are then homogenized and the 1D P_3 solution is used to compute the axial $\hat{D}_s$. This process is repeated until both the coarse and fine mesh solutions are converged.

Another key purpose of CMFD is to accelerate the solution to the eigenvalue problem. To understand this, it is easier to consider the matrix notation of the CMFD system. There are four major components to the matrix notation: the diffusion operator $\mathbf{D}$, the collision operator $\mathbf{T}$, the scattering operator $\mathbf{S}$, and the fission operator $\mathbf{F}$. These four terms are combined into the generalized CMFD eigenvalue problem

$$\left(\mathbf{D}+\mathbf{T}-\mathbf{S}\right)\phi=\frac{1}{k_{eff}}\mathbf{F}\phi \tag{4}$$

and is simplified to

$$\mathbf{M}\phi=\frac{1}{k_{eff}}\mathbf{F}\phi \ , \tag{5}$$

where $\mathbf{M}$ is the migration matrix.

The migration matrix for this case represents the movement of neutrons through space and energy. In MPACT, it is ordered such that there is a dense block matrix that represents the transfer of neutrons between energy groups caused by scatter, as seen in Fig. 1.

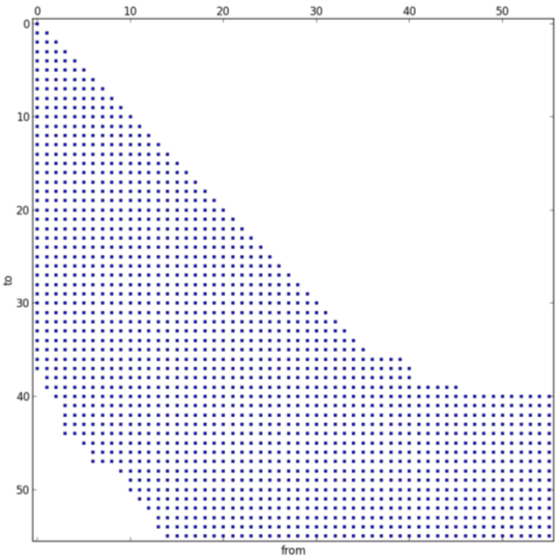


Figure 1: 56 Energy Group Migration Matrix

The scattering matrix as well as the nonlinear correction term creates a structural asymmetry in the migration matrix. Along with the scattering matrix, the nonlinear correction term introduces an asymmetry in the matrix. It should also be noted that the fission matrix is both asymmetric and rank deficient.

The main focus of this work is to accelerate the convergence of this eigenvalue problem by looking at advanced solution schemes to solve for the dominant eigenvalue of this linear system.

II. EIGENVALUE SOLUTION METHODOLOGIES

Several methods have been proposed over the years to solve this generalized eigenvalue problem. This section highlights many of the different approaches but focuses on the current methodology (power iteration with a fixed shift) and the methodology used in this work (generalized Davidson).

One of the most common methods for solving the CMFD eigenvalue system is to cast the generalized eigenvalue problem into a standard eigenvalue problem:

$$\mathbf{M}^{-1}\mathbf{F}\phi=k_{eff}\phi \ . \tag{6}$$

Since the migration operator is nonsingular, it can be inverted and moved to the left hand side of the equation. Once we have a standard eigenvalue problem, power iteration can be applied to obtain the dominant eigenvalue. Typically storing the full factorization of the migration operator is impractical for most problems; so instead, a linear system is solved each iteration:

$$\mathbf{M}y=z\,, \tag{7}$$

where $z=\mathbf{F}f^{n}$.

Although this iteration scheme is straightforward, the convergence of power iteration is highly dependent on the dominance ratio

$$r=\frac{k_2}{k_1}\ , \tag{8}$$

where k_1 and k_2 are the largest and second-largest eigenvalues of system, respectively. This convergence rate can be improved by introducing a shift (μ) to the eigenvalue system

$$\left(\mathbf{M}-\mu\mathbf{F}\right)^{-1}\mathbf{F}\phi=\tilde{k}\phi\ , \tag{9}$$

where μ is an approximation to the inverse of the dominant eigenvalue. It is straightforward to show that the eigenvalue of the original system relates to the eigenvalue of the shifted system by

$$k_{eff}=\frac{\tilde{k}}{1+\mu\tilde{k}}\ , \tag{10}$$

and that the dominance ratio of the shifted system can be reduced significantly. However, shifted power iteration does not come without risk. First, an incorrect choice of shift could alter the convergence to an eigenvalue that does not reflect the largest eigenvalue of the original system. Second, as the shift becomes closer to the true solution of the original system, the linear system becomes more and more difficult to solve. The obvious extreme is if the shift is exactly the inverse of the true eigenvalue of the system; $\left(\mathbf{M}-\mu\mathbf{F}\right)$ is a singular matrix, so care must be taken when trying to solve shifted systems.

Traditionally in MPACT, a fixed shift of 2/3 is used. This was determined to be an acceptable balance between performance of the solver and ensuring that the system is not overshifted, which is unstable. Another approach that has been used in many other places is to have a variable shift based on the current eigenvalue iteration and potentially previous iteration. PARCS [5] uses a variable shift based on iteration history:

$$\mu^{(n)}=\mu_P^{(n)}=\max\left\{\frac{1}{k^{(n-1)}}-C_1\left|\frac{1}{k^{(n-1)}}-\frac{1}{k^{(n-2)}}\right|-C_0,\mu_{\min}\right\}, \quad (11)$$

where C_0, C_1, and $\mu_{\min}$ are constants (0.02, 10, and 1/3, respectively). This method tries to keep enough margin from the actual eigenvalue to ensure stability of the method.

Another common iteration-dependent approach is the Rayleigh Quotient Iteration (RQI) [6], which uses the current eigenvalue estimate as the shift. The RQI iteration strategy becomes

$$\begin{aligned}\phi^{(n+1)}&=\left(\mathbf{M}-\mu^{(n)}\mathbf{F}\right)^{-1}\mathbf{F}\phi^{(n)},\\ \mu^{(n+1)}&=\frac{\left\langle\phi^{(n+1)},\mathbf{M}\phi^{(n+1)}\right\rangle}{\left\langle\phi^{(n+1)},\mathbf{F}\phi^{(n+1)}\right\rangle},\end{aligned} \quad , \quad (12)$$

where $\langle\bullet,\bullet\rangle$ denotes the inner product of two vectors. The main advantage of RQI is that the convergence is quadratic; thus very few iterations are required for convergence. However, the linear system becomes increasingly more difficult to solve because the shifted matrix approaches a singularity. RQI is also not guaranteed to converge to the dominant eigenvalue.

Thus far, all of the methods discussed here have been fixed-point methods; that is, the next estimate of the solution depends only on the estimate immediately preceding it. A set of alternatives to fixed-point iterations are subspace eigenvalue solvers in which information from several previous vectors is used to generate the next approximate solution. One such method that has been used successfully for the solution of both the discrete ordinates and SP_N forms of the transport equation is the generalized Davidson method [7,8]. The advantage of this method relative to other approaches (such as Arnoldi's method) is that it can be applied directly to a generalized eigenvalue problem without requiring conversion to a standard eigenvalue problem. It accomplishes this by using a Rayleigh-Ritz procedure, which solves the projected eigenvalue problem

$$\mathbf{V}^T\mathbf{M}\mathbf{V}y=\lambda\mathbf{V}^T\mathbf{F}\mathbf{V}y\ , \quad (13)$$

where $\mathbf{V}$ contains a set of (typically orthogonal) basis vectors for the current subspace. For an appropriate selection of the subspace, the eigenvalues of the projection problem will closely approximate the original system and the vectors of $\mathbf{V}y$ will approximate the eigenvector. New vectors are added to the subspace by applying a preconditioner to the current eigenvalue residual:

$$\mathbf{P}v=-r^{(n)}. \quad (14)$$

Further details about the generalized Davidson solver can be found in [7,8]. The Davidson method is extremely attractive because unlike the methods discussed previously, a linear system solve is not required. Instead, only a preconditioner application that approximates the solution of a linear system is needed. The choice of preconditioner is important for the Davidson solver to quickly converge. To avoid singularities in the preconditioner system, the preconditioner is chosen based on the migration operator $\mathbf{M}$ rather than the shifted operator $\left(\mathbf{M}-\lambda^{(n)}\mathbf{F}\right)$.

III. MPACT ITERATION SCHEME

The iteration scheme used in MPACT consists of three major components: (1) the radial MOC solver, which captures the radial heterogeneity; (2) the axial P_N solver, which captures axial streaming, and (3) the CMFD solver, which consists of the mechanics to set up the migration and fission matrices and the actual eigenvalue solve. Figure 2 shows the flow chart that MPACT uses to iterate between the three components.

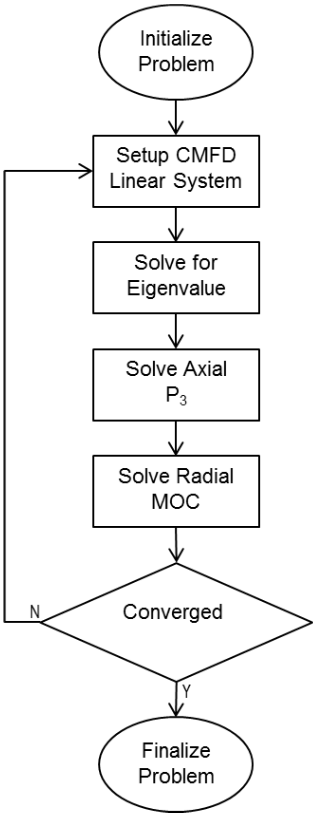


Figure 2: Eigenvalue Iteration Strategy in MPACT

First, the CMFD system is set up with the nonlinear correction factor set to zero, and the eigenvalue system is then solved. The axial solvers and then the radial solvers are used to compute fine mesh fluxes and currents on the coarse mesh. If the solution is converged, the simulation is complete; but if the solution has not met the convergence criteria, the nonlinear correction terms are computed and the CMFD system is reconstructed. This “outer” iteration is continued until the system is complete.

The “Solve for Eigenvalue” box in Fig. 2 is the main point of interest for this paper. The following three sections describe the methods compared in this work.

1. Fixed Shift Power Iteration

Historically, MPACT has used a fixed-shift power iteration by constructing the shifted matrix with full spatial and energy detail in parallel and using the GMRES solver in PETSc [9] with a block-ILU preconditioner. Figure 3 shows the structure of the full migration matrix with fixed shift.

The remaining fission source is computed with each power iteration, and the previous flux solution is provided as an initial guess to the Krylov solver with each power iteration.

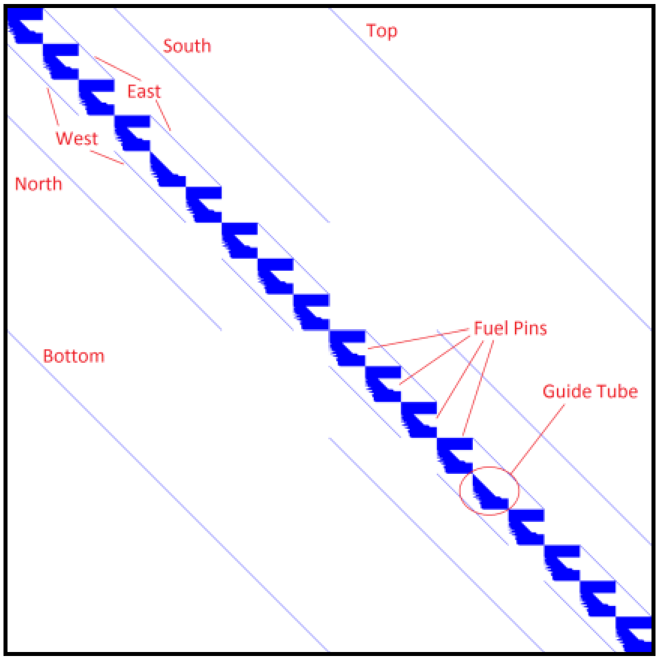
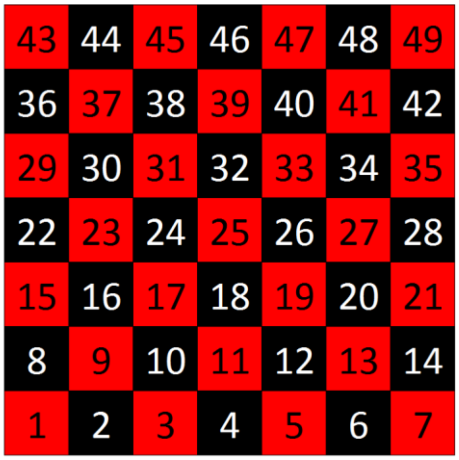


Figure 3: Migration Matrix Structure

It can be observed in this matrix that a component of the fission matrix appears to account for the fixed shift, but this term goes to zero as the shift approaches zero.

2. Red-Black Successive Over-Relaxation

In an attempt to achieve a more efficient solution of the CMFD eigenvalue problem, a red-black Gauss-Siedel scheme was implemented into MPACT [10]. The red-black scheme tags each coarse cell as red or black to produce a checkerboard pattern (as in Fig. 4 for a simple 7 × 7 problem).



43	44	45	46	47	48	49
36	37	38	39	40	41	42
29	30	31	32	33	34	35
22	23	24	25	26	27	28
15	16	17	18	19	20	21
8	9	10	11	12	13	14
1	2	3	4	5	6	7

Figure 4: Red-Black Indexing

Because the inner iteration convergence for Gauss-Seidel solvers tends to be slower than that for many other iterative solvers, each CMFD eigenvalue update iteration is restricted to performing a user-specified maximum number of iterations.

To increase the applicability of the solver, both message passing interface (MPI) and open multiprocessing (OpenMP) are available to provide parallelization. When MPACT is executed, the user provides inputs specifying the number of processors used for spatial decomposition (MPI), MOC angle decomposition (MPI), and MOC ray decomposition (OpenMP). The current solver takes advantage of the processors used for spatial decomposition, as well as the OpenMP threads, which are repurposed from the application to MOC to provide additional spatial decomposition during the sweeps over the red and black indexes. One advantage of this approach is that PETSc is restricted to using MPI, though a hybrid MPI/OpenMP approach is being considered by the PETSc development team.

Gauss-Seidel is a special case of the successive over-relaxation (SOR) solver, where the relaxation factor equals unity. However, SOR typically applies a relaxation factor larger than unity to accelerate the convergence of the system, as in Eq. 15 [11]:

$$\phi_{n,g}^{i} = \omega\phi_{n,g}^{i} + (1-\omega)\phi_{n,g}^{i-1} \ , \tag{15}$$

where *i* denotes the iteration index, *n* denotes the cell index, and *g* denotes the group index.

The determination of optimal relaxation factors has been studied extensively. This implementation uses adaptive relaxation factors based on the Cyclic Chebyshev Semi-Iterative (CCSI) method [12] where the red and black fluxes use different relaxation factors (Eqs. 16–17), which eventually converge to the same value. Equation 16 shows the initial relaxation factors, and Eq. 17 shows the relaxation factor for subsequent iterations:

$$\begin{aligned} &\omega_R = 1, \\ &\omega_B = \frac{1}{1-\frac{1}{2}\rho_J^2}, \end{aligned} \tag{16}$$

$$\begin{aligned} &\omega_R^{i+1} = \frac{1}{1-\frac{1}{4}\rho_J^2\omega_B^i}, i \geq 1, \\ &\omega_B^{i+1} = \frac{1}{1-\frac{1}{4}\rho_J^2\omega_R^{i+1}}, i \geq 1. \end{aligned} \tag{17}$$

These equations make use of the Jacobi spectral radius, which, in this work, is estimated during the first 10 inner iterations, applying no relaxation, to obtain the Gauss-Seidel spectral radius. The Gauss-Seidel spectral radius is then converted to the Jacobi spectral radius.

3. Generalized Davidson Method

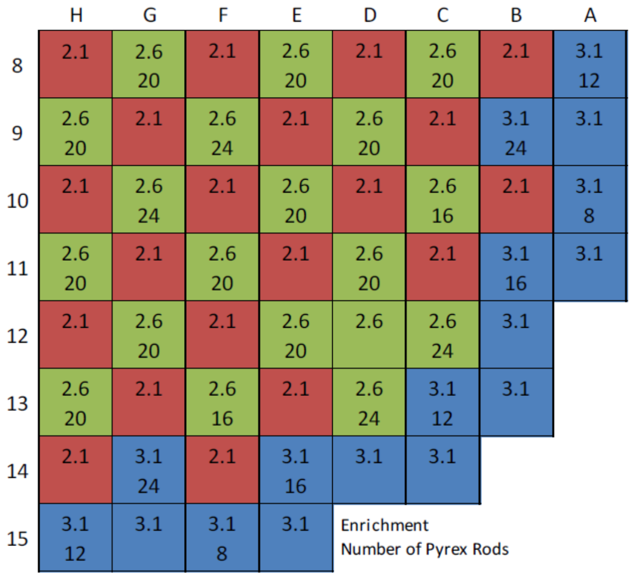
The previous two implementations use the same underlying shifted power-iteration methodology but alter the solution methodology for the linear system. The generalized Davidson solver alters this mechanism. The Anasazi package in Trilinos is used primarily because of the availability of a wide range of preconditioners. MPACT directly provides the migration and fission matrices to Anasazi at every outer iteration.

The generalized Davidson solver in Anasazi is used with the algebraic multigrid preconditioner package ML [13], which is a smoothed aggregate preconditioner. The preconditioner is allowed up to eight total levels and thee smoothing sweeps per level. An ILU smoother with a parallel domain overlap of one is used on each level. Although ML allows the user to define sets of coupled equations, each spatial and energy matrix entry is considered uncoupled from the others. Attempts were made to define the matrix as having 47 coupled equations to capture the energy dependency of the system, but the uncoupled preconditioner significantly outperformed the same cases when all groups were coupled.

IV. RESULTS

Results from this work demonstrate the performance of MPACT and the CMFD solver on a range of cases and core counts on two different machines. The majority of the cases for this work are run on the Titan computer maintained by the Oak Ridge Leadership Computing Facility at Oak Ridge National Laboratory.

The first case considered is the VERA Progression Problem [14] 5-2D. A 2D slice of a quarter-core pressurized-water reactor (PWR) is shown in Fig. 5.



	H	G	F	E	D	C	B	A
8	2.1	2.6 20	2.1	2.6 20	2.1	2.6 20	2.1	3.1 12
9	2.6 20	2.1	2.6 24	2.1	2.6 20	2.1	3.1 24	3.1
10	2.1	2.6 24	2.1	2.6 20	2.1	2.6 16	2.1	3.1 8
11	2.6 20	2.1	2.6 20	2.1	2.6 20	2.1	3.1 16	3.1
12	2.1	2.6 20	2.1	2.6 20	2.6	2.6 24	3.1	
13	2.6 20	2.1	2.6 16	2.1	2.6 24	3.1 12	3.1	
14	2.1	3.1 24	2.1	3.1 16	3.1	3.1		
15	3.1 12	3.1	3.1 8	3.1	Enrichment Number of Pyrex Rods			

Figure 5: Core Layout for Problem 5 Cases

This case is run with a 15-core, 73-core (assembly based), and 257-core (fully decomposed) core count. The 15-core partition is not necessarily optimal, but it is as close as possible to obtaining a ~1,000-core solution for a 3D quarter core problem. Table I shows the number of iterations, the total run time for MPACT, and the CMFD solve time. Three cases are shown: (1) a shifted power iteration with a constant shift of 1.5 using the GMRES solver from PETSc for a linear solver with a block ILU preconditioner; (2) a shifted power iteration with a constant shift of 1.5 using a red-black successive over-relaxation (RBSOR) linear solver; and (3) the generalized Davidson solver from Trilinos, preconditioned with algebraic multigrid (AMG) from ML.

TABLE I. Problem 5-2D Performance on Titan

	15 procs		
	Its.	Total	CMFD
Shift = 1.5, PETSc GMRES	12	46:51	35:58
Shift = 1.5, RBSOR	12	22:56	12:13
Gen. Davidson, AMG	9	09:54	01:20
	73 procs		
Shift = 1.5, PETSc GMRES	11	10:15	07:26
Shift = 1.5, RBSOR	11	06:51	04:09
Gen. Davidson, AMG	11	02:16	00:18
	257 procs		
Shift = 1.5, PETSc GMRES	15	05:13	04:08
Shift = 1.5, RBSOR	15	02:14	01:134
Gen. Davidson, AMG	17	01:14	00:13

It can be seen that across all problems, the CMFD run time is significantly decreased, which results in run times reduced by 4 to 5 times.

The second case considered is the full-height, quarter-core VERA Progression Problem 5a. This case is run on 870 cores with 15 radial partition and 4,234 core with assembly radial partition). This is the target capability for MPACT to solve. The 15-core radial partition with 58 axial planes yields a less than 1,000-core solution to quarter-core PWR geometry, which is the target platform for VERA-CS. The 1,000-core solution is important because it is a computational resource that CASL believes will be available to the nuclear industry in coming years.

Table II shows the problem 5a performance on Titan. Again, the three solution methods are evaluated at the ~1,000-core solution and the current 4,234-core solution. The RBSOR result for the 4,234-core solution was not run to reduce computational resources. As with the 2D cases, significant decreases in CMFD and total run times are observed in all cases.

TABLE II. Problem 5a Performance on Titan

	870 procs		
	Its.	**Total**	**CMFD**
Shift = 1.5, PETSc GMRES	13	1:05:25	47:24
Shift = 1.5, RBSOR	14	37:55	19:57
Gen. Davidson, AMG	12	19:28	03:48
	4234 procs		
Shift = 1.5, PETSc GMRES	12	19:58	12:19
Shift = 1.5, RBSOR	--	--	--
Gen. Davidson, AMG	12	08:55	02:11

An extension to this problem is to run in full core to determine the difference in performance as the problem size is increased. These cases used 4,408 processors on Titan [15]. Only the default method and the generalized Davidson method were used for this case. As seen in Table III, the performance is consistent with the previous cases.

TABLE III. Problem 5 Full-Core Performance on Titan

	Its.	**Total**	**CMFD**
Shift = 1.5, PETSc GMRES	13	1:04:55	43:46.4
Gen. Davidson, AMG	12	0:27:03	08:46.8

V. CONCLUSIONS

In this work, a new eigenvalue solution method was implemented into the CMFD solver in MPACT. Overall, the CMFD solve time was reduced from the default MPACT methods by approximately 13 times, which yields an overall reduction in MPACT solve times of approximately 3.5 times. This allows MPACT and VERA-CS production runs to require ~1,000 cores instead of the 4,000+ cores currently required without a significant change in run time. VERA-CS can also be solved significantly faster if the same core count is used.

Although significant improvements in run time are observed, additional work remains. Further testing is needed for full-core coupled depletion problems to ensure robustness and stability. Additionally, there are minor performance tuning activities that can be made in CMFD such as adaptive iteration control, optimization of how often the preconditioner is updated, and optimization of parameters in the preconditioners. This has become especially important in continuing assessment of the Davidson Method because during reactor depletion, there are observed cases where the preconditioner breaks down which can lead to slow or even non-convergence. There is also an optimization needed with depletion because power iteration converges very rapidly when the initial guess from previous time steps are used but the cost of the Davidson method remains approximately the same.

ACKNOWLEDGMENTS

This research was supported by the Consortium for Advanced Simulation of Light Water Reactors (www.casl.gov), an Energy Innovation Hub (http://www.energy.gov/hubs) for Modeling and Simulation of Nuclear Reactors under U.S. Department of Energy Contract No. DE-AC05-00OR22725.

This research used resources of the Oak Ridge Leadership Computing Facility at Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

REFERENCES

1. Consortium for the Advanced Simulation of Light Water Reactors, http://www.casl.gov.
2. K. SMITH, “Nodal method storage reduction by nonlinear iteration,” *Trans. Am. Nucl. Soc*. 44, 265 (1983).
3. T. SUTTON, B. AVILES, “Diffusion theory methods for spatial kinetics calculations,” *Prog. Nucl. Energy* 30, 119 (1996).
4. MPACT Team, *MPACT Theory Manual, Version 2.0.0*, Tech. Rep. CASL-U-2015-0078-000, Oak Ridge National Laboratory and University of Michigan (2015).
5. Downar, T. J. (2002). PARCS: Purdue Advanced Reactor Core Simulator. PHYSOR, Seoul, Korea.
6. R. SLAYBAUGH, et. al. “Rayleigh Quotient Iteration in 3D, Deterministic Neutron Transport,” *Proc. PHYSOR 2012*, Knoxville, Tennessee (2012).
7. S. HAMILTON, *Numerical Solutions to k-eigenvalue Problems*, PhD Thesis, Emory University (2011).
8. S. HAMILTON, T. EVANS, “Efficient Solution of the Simplified P_N Equations,” *Journal of Computational Physics*, 284, 155 (2014).
9. S. BALAY et al., “PETSc User Manual,” ANL-95/11 – Revision 3.5, Argonne National Laboratory, http://www.mcs.anl.gov/petsc (2014).
10. S. Stimpson, B. Collins, “Implementation of a Red-Black SOR CMFD Solver in MPACT”. *Transactions of the American Nuclear Society.* 115, (2016).
11. R.S. VARGA, *Matrix Iterative Analysis*, Prentice-Hall, Englewood Cliffs, N.J. (1962).
12. J. GEHIN, “A Quasi-Static Polynomial Nodal Method for Reactor Analysis,” Ph. D. Dissertation, Massachusetts Institute of Technology, September (1992).
13. M.W. Gee, et.al. “ML 5.0 Smoothed Aggregation User's Guide” . SAND2006-2649. Sandia National Laboratories. (2006).
14. A. T. GODFREY, “VERA Core Physics Benchmark Progression Problem Specifications,” CASL-U-2012-0131-004 (2014).
15. OAK RIDGE LEADERSHIP COMPUTING FACILITY, “Introducing Titan – The World’s #1 Open Science Supercomputer,” http://www.olcf.ornl.gov/titan/ (2013).