A Study of Various Thermal Upscattering Acceleration Schemes for Massively Parallel Transport Sweeps

Milan Hanuš[*], Jean C. Ragusa[*], Michael Hackemack[†]

[*]*Dept. of Nuclear Engineering, Texas A&M University, College Station, TX, 77843-3133*
[†]*Naval Nuclear Laboratory, KAPL, Schenectady, NY 12301-1072*
mhanus@tamu.edu, jean.ragusa@tamu.edu, michael.hackemack@unnpp.gov

Abstract - *In this work, we investigate various thermal upscattering iterative schemes for solving the multigroup transport equations. Our goal is to efficiently solve massively parallel problems with significant upscattering in the thermal energy range, for which the traditional iterative methods are converging too slowly. The presented methods are based on the two-grid diffusion synthetic acceleration (DSA), originally developed for the Gauss-Seidel scheme, which we extend to the more parallelizable Jacobi iteration. Furthermore, we propose a new approach which "boosts" the convergence of the within-group iterations by performing additional group-by-group DSA solves. This removes the need for fully converged within-group iterations, further reducing the overall number of transport sweeps. We perform Fourier analyses for the proposed schemes and apply them to several practical problems using the parallel deterministic S_N transport solver PDT. We also discuss how to cope with the increased number of diffusion solves in our "boosted" scheme and their effect on parallel scalability.*

I. INTRODUCTION

The multigroup neutron transport equations form a coupled system of single-group transport equations with scattering (and fission, if present) events coupling the various groups together. Oftentimes, iterative techniques are employed to solve the system of multigroup equations by looping over the energy groups, one group at a time. At high enough energies, neutrons only downscatter; the scattering matrix is lower triangular and only one pass through these fast neutron groups is necessary to yield the fast flux solution. In the thermal range, upscattering can be significant and thermal iterations are required. This situation is exacerbated in low-leakage configurations containing Graphite and Heavy Water. Indeed, these materials have very low neutron absorption cross sections and standard iterative schemes, such as Gauss-Seidel over the thermal groups, are slowly convergent, with a spectral radius approaching unity.

In [1], Morel et al. proposed a two-grid acceleration for the Gauss-Seidel iterative scheme. The two-grid aspect of their scheme is a reference to (spatial) multigrid techniques; in their work, they collapsed the "fine" energy grid (the thermal group structure) onto a "coarse" energy grid (a single macro-group over the entire thermal range), then performed a single diffusion synthetic acceleration (DSA) on the coarse energy grid in order to estimate a scalar flux correction which was then added to the previous Gauss-Seidel iterate. We note that, in this approach, the within-group scattering iterations must be converged reasonably well. In Graphite and Heavy Water, the within-group scattering ratio $c_g = \frac{\sigma_{s,gg}}{\sigma_{t,g}}$ can be significant, with several groups for which $c_g > 0.96$, for instance, and thus many one-group transport solves are required to iteratively solve the within-group problem. Evans et al. [2] proposed a modification of Morel's two-grid scheme whereby (a) a transport synthetic accelerator is employed in lieu of a diffusion solve and (b) the inner (within-group) iterations need not to be fully converged. They state that employing a transport solver avoids some of the difficulties associated with diffusion synthetic accelerators (as in the case of highly heterogeneous material configurations); however, the most slowly converging modes of the Gauss-Seidel thermal iterative scheme are the diffusion modes. In addition, diffusion solves are significantly less expensive than transport solves/sweeps. Hence, in this work, we only consider synthetic accelerators based on diffusion theory.

In this work, we present an investigation of various iterative techniques to solve the multigroup equations. In addition to a Gauss-Seidel method (with or without converging the within-group iterations), we also analyze a Jacobi approach (with and without convergence of the inners). A Jacobi method allows for greater flexibility in parallel computations (Gauss-Seidel is essentially a sequential approach). The two-grid technique is applied to both Jacobi and Gauss-Seidel solves. Converging the within-group iterations may be important to ensure fast convergence of the thermal upscattering equations; however, this is costly in terms of transport solves. We analyze an alternate approach whereby a single one-group transport solve is performed by group, followed by a within-group DSA to "boost" the convergence of the inner iterations. Fourier analyses are carried out for each scheme and are compared against numerical results using the PDT code, a massively parallel deterministic S_n transport solver. PDT uses transport sweeps and has been scaled efficiently up to 768,000 processors [3].

II. ITERATIVE SCHEMES FOR THERMAL UPSCATTERING

1. Formalism

The multigroup transport equations are written as follows, using an operator notation:

$$\boldsymbol{L}_g \Psi_g = \boldsymbol{M} \sum_{g'} \boldsymbol{S}_{gg'} \Phi_{g'} + \boldsymbol{Q}_g, \qquad 1 \le g \le G\,, \tag{1a}$$

$$\Phi_g = \boldsymbol{D} \Psi_g\,, \tag{1b}$$

where $\boldsymbol{L}_g$ is the streaming+interaction operator, $\boldsymbol{M}$ is the moment-to-discrete operator, $\boldsymbol{S}$ is the transfer operator ($\boldsymbol{S}_{gg'}$

represents scattering from group g' to group g), $\boldsymbol{D}$ is the discrete-to-moment operator, $\boldsymbol{Q}$ is the external source (which may contain the previous fission source iterate in the case of eigenvalue problems). $\boldsymbol{L}$ is a block diagonal operator on groups. We employ a DFEM S_n technique, so each $\boldsymbol{L}_g$ operator is "inverted" using transport sweeps. Ψ is the multigroup angular flux vector and Φ is the multigroup flux moment vector. G is the total number of energy groups.

In the case of a Gauss-Seidel solution technique, where the inner iterations are fully converged, the iteration is written as follows:

$$\boldsymbol{L}_g\Psi_g^{(k+1)} = \boldsymbol{M}\sum_{g'=1}^{g'=g}\boldsymbol{S}_{gg'}\Phi_{g'}^{(k+1)} + \boldsymbol{M}\sum_{g'=g+1}^{g'=G}\boldsymbol{S}_{gg'}\Phi_{g'}^{(k)} + \boldsymbol{Q}_g\,. \tag{2}$$

Introducing $\boldsymbol{L} = \text{diag}(\boldsymbol{L}_1, \ldots, \boldsymbol{L}_G)$, $\boldsymbol{S}_L = \text{lower}(\boldsymbol{S})$, $\boldsymbol{S}_D = \text{diag}(\boldsymbol{S})$, $\boldsymbol{S}_U = \text{upper}(\boldsymbol{S})$, Eq. (1a) can be succinctly written as

$$[\boldsymbol{I} - \boldsymbol{TS}]\,\Phi = \boldsymbol{DL}^{-1}\boldsymbol{Q}\,, \tag{3}$$

and Eq. (2) as

$$[\boldsymbol{I} - \boldsymbol{T}(\boldsymbol{S}_L + \boldsymbol{S}_D)]\,\Phi^{(k+1)} = \boldsymbol{TS}_U\Phi^{(k)} + \boldsymbol{DL}^{-1}\boldsymbol{Q}\,, \tag{4}$$

where we have introduced the transport operator $\boldsymbol{T} = \boldsymbol{DL}^{-1}\boldsymbol{M}$. Different iterative schemes are being analyzed here:

1. Gauss-Seidel with full convergence of the within-group iterations, (GSfull),
2. Gauss-Seidel with partial convergence of the within-group iterations using n transport sweeps, (GSpn). Note that one special case of interest will consists in situations where only 1 transport sweep is performed (i.e., GSp1).
3. Jacobi with full convergence of the within-group iterations, (Jfull),
4. Jacobi with partial convergence of the within-group iterations using n transport sweeps, (Jpn), including the special case Jp1.

All of these schemes can be formulated as

$$\Phi^{(k+1)} = \mathcal{T}\Phi^{(k)} + \boldsymbol{q}\,, \tag{5}$$

where the various expressions for $\mathcal{T}$ are given in Table I (and $\boldsymbol{q} = \boldsymbol{DL}^{-1}\boldsymbol{Q}$).

TABLE I. Thermal Upscattering Operator $\mathcal{T} = (\boldsymbol{I} - \mathcal{A})^{-1}\mathcal{B}$ for various iterative schemes

Scheme	operator $\mathcal{A}$	operator $\mathcal{B}$
GSfull	$\boldsymbol{T}(\boldsymbol{S}_L + \boldsymbol{S}_D)$	$\boldsymbol{TS}_U$
GSp1	$\boldsymbol{TS}_L$	$\boldsymbol{T}(\boldsymbol{S}_D + \boldsymbol{S}_U)$
Jfull	$\boldsymbol{TS}_D$	$\boldsymbol{T}(\boldsymbol{S}_L + \boldsymbol{S}_U)$
Jp1	$\boldsymbol{0}$	$\boldsymbol{T}(\boldsymbol{S}_L + \boldsymbol{S}_D + \boldsymbol{S}_U)$

For the partially converged Gauss-Seidel, GSpn, and Jacobi, Jpn, schemes, a general expression for $\mathcal{T}$ is

$$\mathcal{T}^{(n)} = \mathcal{A}^n + \sum_{i=0}^{n-1}\mathcal{A}^i\mathcal{B}\,.$$

For GSpn, $\mathcal{A} = [\boldsymbol{I} - \boldsymbol{TS}_L]^{-1}\,\boldsymbol{TS}_D$ and $\mathcal{B} = [\boldsymbol{I} - \boldsymbol{TS}_L]^{-1}\,\boldsymbol{TS}_U$; for Jp$n$, $\mathcal{A} = \boldsymbol{TS}_D$ and $\mathcal{B} = \boldsymbol{T}(\boldsymbol{S}_L + \boldsymbol{S}_U)$. For n large, the above operator $\mathcal{T}^{(n)}$ tends to $\mathcal{T}$. For instance, in the case of the GSpn scheme, this becomes

$$\begin{aligned}(\boldsymbol{I} - \mathcal{A})^{-1}\mathcal{B} &= \left[\boldsymbol{I} - [\boldsymbol{I} - \boldsymbol{TS}_L]^{-1}\,\boldsymbol{TS}_D\right]^{-1}[\boldsymbol{I} - \boldsymbol{TS}_L]^{-1}\,\boldsymbol{TS}_U \\ &= [\boldsymbol{I} - \boldsymbol{T}(\boldsymbol{S}_L + \boldsymbol{S}_D)]^{-1}\,\boldsymbol{TS}_U\end{aligned} \tag{6}$$

which is the operator $\mathcal{T}$ for GSfull.

2. The two-grid method

Introducing the multigroup error $\epsilon_g^{(k+1)} = \Phi_g - \Phi_g^{(k+1)}$, an error equation can be derived:

$$[\boldsymbol{I} - \boldsymbol{TS}]\,\epsilon^{(k+1)} = \mathcal{B}(\Phi^{(k+1)} - \Phi^{(k)})\,.$$

This equation for the error is just as difficult to solve as the original transport problem, Eq. (3). Its right-hand-side contains the lagged thermal upscattering source terms and depends on the chosen iterative scheme. A low-order synthetic operator will be employed to obtain an approximate correction $\epsilon^{(k+1)}$. Rather than using a multigroup synthetic operator, Morel et al. [1] noted that for the Gauss-Seidel approach (with inners converged), the most poorly attenuated error modes were nearly constant in space, angle, and energy. The main implications of these results are that (1) a diffusion operator will likely perform well as a low-order synthetic accelerator and (2) the multigroup error can be separated into a spectral shape $(\xi_g)_{1\leq g\leq G}$ and a spatial component $\theta(\boldsymbol{r})$. The spectral shape is obtained as the dominant eigenmode of the iteration matrix for the constant-in-space (flat) mode, $\mathcal{T}_\infty\xi = \rho\xi$, or, equivalently,

$$[\boldsymbol{I} - \mathcal{A}_\infty]\,\xi = \rho\mathcal{B}\xi$$

where ρ is the spectral radius and the subscript ∞ denotes the homogeneous infinite medium operator (i.e., without spatial derivatives present; that is to say that the streaming+interaction transport operator $\boldsymbol{L}$ only contains the total cross section). Without neutron absorption, the spectral shape would be exactly the Maxwellian distribution. The spectral shape ξ is computed once for all for each material present in the configuration. Then, the low-order error multigroup equations are summed up to yield a one-group diffusion equation for the spatial component of the error

$$-\boldsymbol{\nabla}\cdot\langle D\rangle\boldsymbol{\nabla}\theta + \langle\sigma_a\rangle\theta = \boldsymbol{1}^T\mathcal{B}(\Phi^{(k+1)} - \Phi^{(k)})\,, \tag{7}$$

where $\boldsymbol{1}$ is a unit column vector of length G, and the spectrally averaged coefficients are

$$\langle D\rangle = \sum_g(3\sigma_{tr,g})^{-1}\xi_g \text{ and } \langle\sigma_a\rangle = \sum_g\left(\sigma_{t,g}\xi_g - \sum_{g'}\sigma_{s0,gg'}\xi_{g'}\right)\,.$$

Finally, the iterative process is:
Loop over thermal groups ($g \in \mathbb{G}_{th}$) and obtain a new flux iterate from transport:

$$\Phi^{(k+1/2)} = \mathcal{T}\Phi^{(k)} + \boldsymbol{q}\,, \tag{8a}$$

Perform a two-grid acceleration:

$$-\boldsymbol{\nabla}\cdot\langle D\rangle\boldsymbol{\nabla}\theta + \langle\sigma_a\rangle\theta = \boldsymbol{1}^T\mathcal{B}(\Phi^{(k+1/2)} - \Phi^{(k)})\,, \tag{8b}$$

Update the transport iterate:

$$\Phi_g^{(k+1)} = \Phi_g^{(k+1/2)} + \theta\xi_g \qquad \forall g \in \mathbb{G}_{th}\,. \tag{8c}$$

3. Within-group acceleration

The success of the two-grid method hinges upon the dominant error mode being distinct from the other persistent error modes. When this is the case, employing a two-grid approach (as opposed to a fully multigrid technique) is sufficient for practical purposes. However, as will be shown in the Results section, not converging the inner (within-group) scattering iterations can seriously degrade the effectiveness of the two-grid method. Requiring reasonable convergence of the inner iterations is costly in terms of transport sweeps. Rather, we propose to perform *only* one transport sweep per group (for computational efficiency) but to accelerate the new flux iterate in each group with one DSA step per group. Then, the iterative process is changed as follows:

Transport sweeps:

$$\Phi^{(k+1/3)} = \mathcal{T}\Phi^{(k)} + \boldsymbol{q}\,, \tag{9a}$$

Within-group DSA error solve ($\forall g \in \mathbb{G}_{th}$):

$$-\boldsymbol{\nabla}\cdot D_g\boldsymbol{\nabla}\delta\Phi_g + \sigma_{a,g}\delta\Phi_g = \sigma_{s0,gg}(\Phi_g^{(k+1/3)} - \Phi_g^{(k)})\,, \tag{9b}$$

Within-group flux update:

$$\Phi_g^{(k+2/3)} = \Phi_g^{(k+1/3)} + \delta\Phi_g \qquad \forall g \in \mathbb{G}_{th}\,. \tag{9c}$$

Two-grid DSA error solve:

$$-\boldsymbol{\nabla}\cdot\langle D\rangle\boldsymbol{\nabla}\theta + \langle\sigma_a\rangle\theta = \boldsymbol{1}^T\mathcal{B}(\Phi^{(k+2/3)} - \Phi^{(k)})\,, \tag{9d}$$

Transport update:

$$\Phi_g^{(k+1)} = \Phi_g^{(k+2/3)} + \theta\xi_g \qquad \forall g \in \mathbb{G}_{th}\,. \tag{9e}$$

4. GMRes formulation

Instead of employing an accelerated Richardson technique, one can recast Eq. (9) as a preconditioned GMRes solve. Recalling the operator notation for transport sweeps,

$$\Phi^{(k+1/3)} = \mathcal{T}\Phi^{(k)} + \boldsymbol{q}\,, \tag{10}$$

and introducing the following for the DSA error solve and the two-grid DSA:

$$\Phi^{(k+2/3)} = \Phi^{(k+1/3)} + \mathcal{D}^{-1}\boldsymbol{S}_D(\Phi^{(k+1/3)} - \Phi^{(k)}) \tag{11}$$

$$\Phi^{(k+1)} = \Phi^{(k+2/3)} + \mathcal{G}^{-1}\left(\Phi^{(k+2/3)} - \Phi^{(k)}\right) \tag{12}$$

where $\mathcal{D}$ is the DSA error operator and $\mathcal{G}$ the two-grid operator, we can finally obtain an expression for the entire iteration:

$$\Phi^{(k+1)} = (\boldsymbol{I} - \mathcal{W})\Phi^{(k)} + \boldsymbol{w} \tag{13}$$

where

$$\mathcal{W} = (\boldsymbol{I} + \mathcal{G}^{-1})(\boldsymbol{I} + \mathcal{D}^{-1}\boldsymbol{S}_D)(\boldsymbol{I} - \mathcal{T}) \tag{14a}$$

$$\boldsymbol{w} = (\boldsymbol{I} + \mathcal{G}^{-1})(\boldsymbol{I} + \mathcal{D}^{-1}\boldsymbol{S}_D)\boldsymbol{D}\boldsymbol{L}^{-1}\boldsymbol{Q}\,. \tag{14b}$$

The preconditoned GMRes equivalent for Eq. (13) is

$$\mathcal{W}\Phi = \boldsymbol{w}\,. \tag{15}$$

This equation is solved using GMRes. The action of the DSA and two-grid preconditioners is apparent in the expression for $\mathcal{W}$.

5. PDT's Handling of Energy Groups

PDT uses transport sweeps to solve the transport equations and has been scaled efficiently up to 768,000 processors (and 1.536 million processes by over-committing the cores on ANL's leadership class supercomputer Mira). To maintain efficiency at such core counts, the transport sweep tasks are aggregated in space (cells), in angle, and energy groups, and are scheduled in a provably optimal fashion (on logically Cartesian grids). In PDT, the concept of group sets (a set of several energy groups) is employed to enhance data locality. Energy groups that are solved concurrently during transport sweeps form the within-group set (WGS). The number of transport sweeps and/or the point-wise relative error can be set to exit a given WGS and tackle the next WGS. For instance, solving a WGS with one transport sweep was described as Jp1 in Table I.

III. RESULTS

In this section, we discuss the main results from a Fourier analysis performed for the various iterative techniques (Gauss-Seidel/Jacobi), with or without two-grid acceleration, with fully converged inner iterations/no inner iteration convergence (n transport sweeps)/one transport sweep followed by a within-group DSA acceleration boost. We then compare the Fourier spectral radius results with the numerical spectral radii obtained using the PDT code which implements the various methods described above. PDT solves the transport equation using an S_n technique in angle and the Discontinuous Finite Element Method (DFEM) in space [4]. The diffusion acceleration schemes also employ a DFEM discretization in space, based on the Symmetric Interior Penalty technique for elliptic operators [5, 6, 7]. We use 99-group cross sections for graphite (57 thermal groups).

1. Fourier analyses

Fourier analyses are carried out in 3-D to determine the spectral radius of the various proposed schemes. The standard Fourier ansatz $\exp(\jmath\boldsymbol{\Lambda}\cdot\boldsymbol{r})$ is used in the governing equations to remove the spatial operators. $\jmath^2 = -1$. $\boldsymbol{\Lambda} = [\lambda_x, \lambda_y, \lambda_z] \in (-\infty, +\infty)^3$ is the Fourier wave number (the flat mode corresponds to $\boldsymbol{\Lambda} = \boldsymbol{0}$). In Fig. 1, the eigenvalues of the iteration matrix for the GSfull, Jfull, GSp1, and Jp1 schemes, with and without two-grid (TG) acceleration, are plotted in the complex plane for the flat (diffusive) mode. We clearly note that: (1) when the inner iterations are converged, the slowest converging mode (eigenvalue close to 1) is well separated from the other modes and that the TG acceleration is effective (with TG, the spectral radius for GSfull is about 0.4 and for Jfull, it is about 0.65); (2) when the inner iterations are not converged, the effectiveness of the TG acceleration is severely reduced.

The eigenvalues for various Jacobi-type schemes are shown in Fig. 2. Without converging the inner iterations (Jp1), the spectral radius (largest eigenvalue) tends to 0.999 (without TG) or 0.961 (with TG). When the within-group are fully converged (Jfull), the spectral radii are 0.994 (without TG)

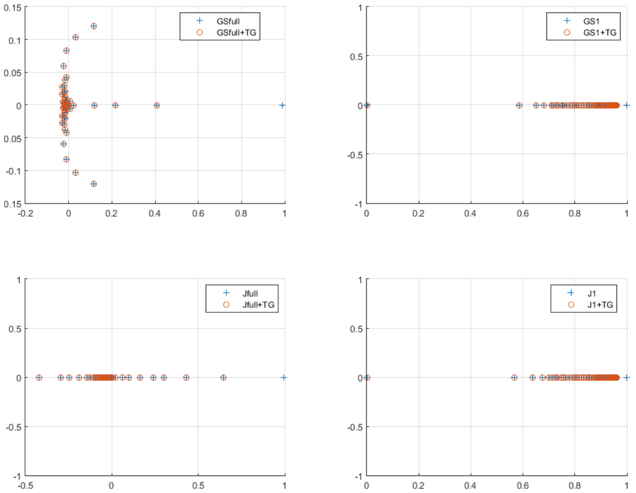

Fig. 1. Plot of the eigenvalues in the complex plane for the GSfull, Jfull, GSp1, and Jp1 schemes, with and without two-grid (TG) acceleration.

and 0.646 (with TG). If the inners are not converged (only one transport sweep) but accelerated with a within-group DSA, the spectral radii are 0.994 (without TG) and 0.646 (with TG), that is identical to the fully converged case. This represents a tremendous savings in terms of transport sweeps. Table II provides the spectral radii for various schemes (TG=two-grid acceleration, wgDSA=within-group DSA). The Fourier and PDT results are in excellent agreement.

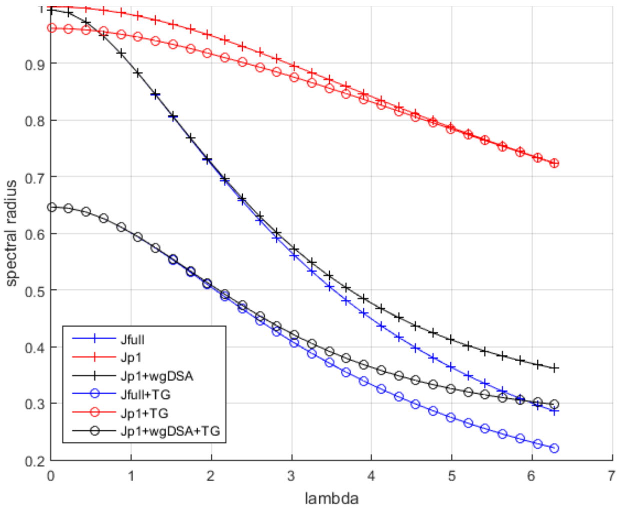

Fig. 2. Eigenvalue for the Jfull, Jp1, Jp1 with within-group DSA (wgDSA) schemes, with or without two-grid (TG) acceleration.

TABLE II. Spectral Radii for various iterative schemes

Scheme	Fourier	PDT
GSfull	0.988336	0.988292
GSfull+TG	0.408389	0.408380
GSp1+TG	0.960420	0.960416
GSp1+TG+wgDSA	0.408389	0.408379
Jfull	0.994144	0.994121
Jfull+TG	0.646230	0.646221
Jp1+TG	0.961350	0.961346
Jp1+TG+wgDSA	0.646230	0.646221

2. PDT results

The last column of Table II shows the numerical spectral radii computed using PDT when solving the following problem: A graphite block of width 10^6 cm with vacuum boundary conditions, zero source, and an initial flux sampled randomly in $(0, 1)$, and the same 99-group cross-section data as in the Fourier analysis. The exact solution for this problem is a zero flux. Hence, we can compare the various iterative methods by their ability to drive the initial random guess to zero and numerically evaluate the spectral radius as

$$\rho^{(k+1)} = \frac{\left\|\Phi^{(k+1)}\right\|_\infty}{\left\|\Phi^{(k)}\right\|_\infty}, \quad \text{where} \quad \|\Phi\|_\infty = \max_{g,i} \left|\Phi_{g,i}\right|$$

and i denotes the spatial index.

Table III summarizes the amount of work each method required to obtain $\Phi^{(k)}$ such that $\left\|\Phi^{(k)}\right\|_\infty < 10^{-100}$. The amount of work is measured in terms of the number of one-group transport sweeps needed in the thermal energy range and the number of iterations of the one-group diffusion solver (we use the PCG solver from the Hypre library [8] with the BoomerAMG preconditioner [9]). Note that the number of sweeps in the fast range was the same for all methods (69,608), such that the fast flux is fully resolved in that range to within machine underflow precision ($\sim 10^{-308}$).

TABLE III. PDT work cost for various iterative schemes († = not converged to 10^{-100})

Scheme	# sweeps	# diff. sln.	# upscat. it.
GSfull	6,703,426	0	$400^{\dagger}$
GSfull+TG	5,306,465	1,036	259
GSp1+TG	323,589	22,708	5,677
GSp1+TG+wgDSA	14,763	60,965	259
Jfull	29,192,607	0	$800^{\dagger}$
Jfull+TG	33,679,590	2,132	533
Jp1+TG	330,600	23,200	5,800
Jp1+TG+wgDSA	30,381	126,398	533

The unaccelerated methods (GSfull, Jfull) were only able to reduce the initial norm of the solution to only $\sim 10^{-3}$ – the large number of sweeps required for that emphasizes the well-known need for a thermal up-scattering acceleration scheme for such kinds of problems. While the two-grid acceleration reduces the norm of the flux much faster using a comparable number of sweeps (GSfull+TG, Jfull+TG), we can see that we can obtain the solution of same quality using significantly fewer transport sweeps by limiting the number of within-group iterations to one (GSp1+TG, Jp1+TG). Of course, this increases the number of up-scattering iterations, and, hence, the number of two-grid DSA solves, as is clear from comparing the values in the third and fourth column of Table III. However, rows GSp1+TG+wgDSA and Jp1+TG+wgDSA confirm our Fourier analysis, i.e., we can nearly recover the spectral radius of a particular iterative method with fully converged inner iterations by performing the within-group DSA solve for each thermal group. This leads to the smallest overall number of transport sweeps, but at the same time to the largest number of diffusion solves. Nevertheless, the total number of sweeps and diffusion solver iterations is still much lower for the methods involving the within-group DSA boost than for those without. Balancing the cost of transport sweeps and diffusion solves is therefore crucial to make the proposed schemes meaningful to use for practical calculations and we will focus on that matter in the following sections.

3. Reducing the cost of DSA

The purpose of this section is to assess the effect of the acceleration on overall solution time. To this end, we consider a homogeneous graphite block using the same 99-group cross-section data as in the previous tests, vacuum boundary conditions and distributed unit isotropic source. As we are ultimately interested in solving increasingly larger problems using increasingly larger computing power available, we assess parallelism using a weak scaling study, in which the size of the problem grows as the CPU core count increase. The largest number of CPU cores used is $128,000$. The cell width is kept fixed at 2 cm (approximately 10 mean free paths of an average thermal neutron).

When designing the scaling study, we need to decide on the 3D partitioning of the available processors. PDT typically uses the KBA-like decomposition for transport sweeps [10], in which one dimension is assigned to two processors and the other two dimensions to the remaining processors in a square fashion. However, this decomposition is likely less suitable for the parallel AMG-PCG solver used in the (diffusion) acceleration phase, which would favor a volumetric domain decomposition. For the numerical results presented in this paper, we have chosen a partitioning that is closer to a volumetric decomposition, see in Table IV; as such, we should expect transport sweeps to scale in a slightly less than optimal manner. All parallel calculations were done on the Vulcan supercomputer at the Lawrence Livermore National Laboratory.

TABLE IV. Parallel partitioning.

#cores	1	8	64	512	4k	32k	128k
P_x	1	1	2	4	8	16	32
P_y	1	2	4	8	16	32	64
P_z	1	4	8	16	32	64	64

The piecewise linear discontinuous finite element approximation is used to discretize both the transport and the diffusion operators in space. With a fixed number of 128 cells per core, this results in 1024 spatial degrees of freedom per core. The Gauss-Legendre-Chebyshev product quadrature is used to discretize the angular domain with 1024 total angles, aggregated into 256 angle sets.

Although Fourier results show the superior convergence rate of the two-grid method with a Gauss-Seidel splitting, it is inherently sequential and lacks the scaling potential of the block Jacobi splitting. Moreover, it does not fit well into the aggregation framework of PDT (which has been shown to be essential for efficient parallel execution, [11]), because it forces each groupset to consist of just a single group. For these reasons, we will henceforth consider only the block Jacobi schemes, in particular the one with the inner iteration boost (Jp1+TG+wgDSA).

A. Effect of the diffusion solver parameters

The first set of calculations was performed with the Hypre/BoomerAMG settings that are generally recommended for 3D diffusion problems (e.g., [12, 13]). We summarize the parameters that are different from Hypre defaults below (see [12] for more details):

- AMG strength threshold 0.8,
- HMIS coarsening,
- 1 level of aggressive coarsening,
- hybrid symmetric Gauss-Seidel relaxation,
- extended+i interpolation, multipass interpolation on levels of aggressive coarsening,
- maximal number of elements per row for interpolation 5.

We used the same parameters for both diffusion solvers (two-grid and within-group DSA), as well as the same stopping criterion (relative PCG residual threshold 10^{-6}). To monitor convergence of the transport solver, we used the maximum relative pointwise change:

$$\tau_p = \left\| \frac{\Phi^{(k+1)} - \Phi^{(k)}}{\max\{\Phi^{(k+1)}, \Phi^{(k)}\}} \right\|_\infty$$

with the stopping criterion $\tau_p = 10^{-6}$ for the fast group sets and $\tau_p = 10^{-4}$ for the thermal group set.

The left bars in Figure 3 show the total solve time for this parameter setup, decomposed by its main components.We can see that both the DSA setup phase (construction of the AMG preconditioner) and the solve phase (PCG iterations) scale poorly relatively to the transport sweeps [1]. The last column of Table V contains the ratios of the acceleration time to the total solve time for this parameter setup – at higher core counts, the acceleration time becomes dominant and destroys any benefits from the reduced number of transport sweeps.

[1] The Hypre times cover both the two-grid and the within-group DSA boost phases, but we note that the small number of the two-grid solver invocations as compared to the within-group DSA solver makes the contribution of the former to the total acceleration time negligible.

TABLE V. DSA performance – default param. ($X_Y = X \times 10^Y$).

#cores	PCG Setup	PCG Solve	PCG it	DSA/Solve
1	12613.2	33186.0	5557	0.0510273
8	68450.3	86654.0	8047	0.0794613
64	666937.0	895490.0	11092	0.293499
512	2.33174_6	2.21697_6	12311	0.496209
4k	4.61688_6	2.85578_6	12555	0.587829
32k	5.80373_6	3.1316_6	12658	0.596342
128k	6.05512_6	3.093_6	12708	0.569176

Before discussing the approaches to improve the performance of the AMG solver through variation of selected Hypre parameters, let us note the relative improvement of the performance from 32k cores to 128k. This is the consequence of the more volumetric decomposition of the 128k processors (see Table IV) and the results corroborate the remark made at the beginning of Section 3. Investigation of how the optimal parallel partitioning for sweeps with DSA preconditioning differs from that for the sweeps alone is one of the directions of our future work.

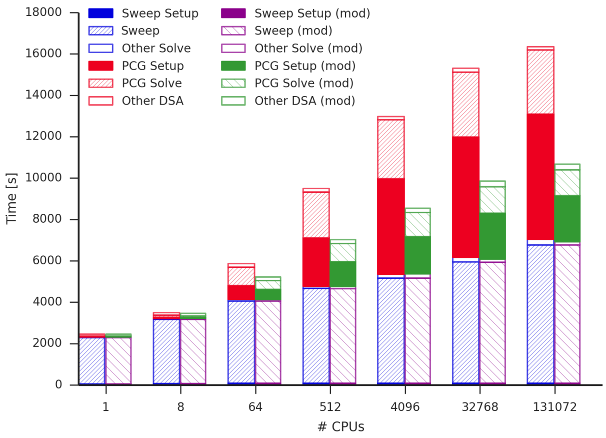

Fig. 3. Comparison of the weak scaling performance with standard and modified Hypre parameter sets.

Poor scalability of AMG with Galerkin coarsening has been recently addressed in several papers (e.g., [14, 15, 16]). Based on the results and recommendations from these papers, we performed the same calculations with the following changes

- Non-Galerkin drop tolerances [0.0, 0.1],
- 10 levels of aggressive coarsening.

The first item specifies a non-zero drop tolerance for sparsifying the standard Galerkin coarse grid matrix on the second and higher levels of the multigrid hierarchy. As the problem size and the number of coarse grid levels increase, the stencil size (number of non-zeros) of the Galerkin coarse matrix grows, leading to increased communication and decreased parallel performance. Replacing the Galerkin coarse grid matrix by its sparser approximation on higher levels should mitigate this problem. We refer to papers [15, 14] for more details about coarse grid sparsification in AMG.

Increasing the number of levels of aggressive coarsening is another way of reducing the stencil size and communication and we use the setting that was shown to be the most robust for diffusion problems in [13].

TABLE VI. DSA performance – adjusted parameters.

#cores	PCG Setup	PCG Solve	PCG it	DSA/Solve
1	12312.9	32771.0	5616	0.0506192
8	60242.6	71042.0	8083	0.0727121
64	491451.0	432728.0	11361	0.208914
512	1.21184_6	877960.0	12890	0.323851
4k	1.79822_6	1.16355_6	12977	0.370976
32k	2.21353_6	1.28137_6	12928	0.381719
128k	2.2346_6	1.23418_6	12923	0.350893

The right bars in Figure 3 and Table VI show a significant improvement in the parallel performance of the DSA, comparing to the original parameter setup. Note that while the total number of PCG iterations increased slightly as a consequence of less powerful AMG preconditioner, the PCG solve times decreased since applying the preconditioner has become cheaper.

B. Reusing the preconditioner in the within-group boost

From the results of the previous section, we can see that the PCG setup time, taken up from the most part by the construction of the AMG preconditioner, dominates the PCG solve time. This is because the setup is repeated for every group at the beginning of each within-group DSA boost [2]. It is reasonable to assume that the material properties do not change significantly from group to group in the thermal range. Then it should be possible to re-use the AMG preconditioner for multiple groups.

Noting that the two-grid DSA operator, built using the spectrally averaged cross sections, encodes the information about all thermal groups, we first attempted to use the AMG preconditioner for the two-grid operator for all the within-group DSA solves. However, this turned out to be a too drastical change, resulting in a significant increase of the number of iterations needed to converge the within-group DSA problems and only a slight improvement of the relative total acceleration time. This is shown in the second row of Table VII, which contains the performance results from the 128k-core runs with the adjusted Hypre settings described in the previous section (Table VI).

A better performing alternative is to re-use the preconditioner from the first (fastest) group in the thermal group set for all the subsequent within-group DSA solves. The corresponding results in the third row of Table VII clearly show the benefits as compared to the original version (first row).

Figure 4 shows the increase of PCG iteration counts for

[2] As far as we know, there is no direct way to store the AMG operators constructed by Hypre, and doing that might also be prohibitive in terms of memory requirements for large thermal group sets.

TABLE VII. Performance of various preconditioner reuse schemes.

Reuse	PCG Setup	PCG Solve	PCG it	DSA/Sol
no reuse	2.2346_6	1.23418_6	12923	0.350893
two-grid	3272.6	3.12625_6	40315	0.328588
first	3239.0	1.55613_6	17671	0.209827
first+last	82129.0	1.45364_6	16407	0.207503

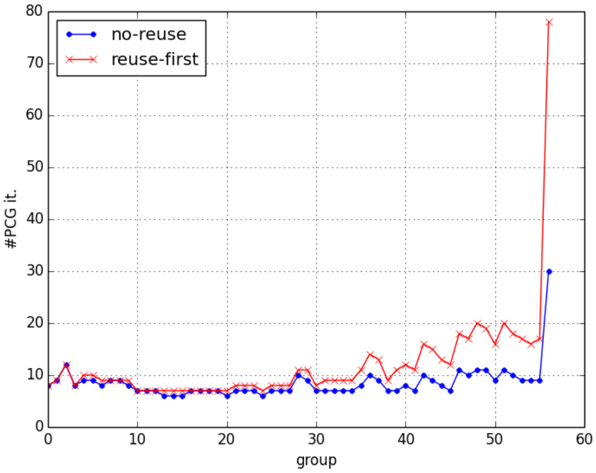

Fig. 4. Number of PCG iterations for each wgDSA solve.

each within-group DSA solve for the first up-scattering iteration (the wgDSA iteration counts almost don't change in subsequent up-scattering iterations). Note the sharp increase in the last thermal group, caused by the large change in material properties (the total cross-section changes by no more than 30% between the thermal groups, but decreases $\sim 15\times$ from the second-to-last to the last group). To capture this change, we used a scheme whereby the preconditioner from the first wgDSA solve is reused for all wgDSA solves until the last one, for which the preconditioner is re-built again (last row in Table VII).

From the results of this section, we conclude that while re-using the preconditioner can bring significant savings in the total acceleration time, the optimal reuse scheme is very problem-dependent and finding it may become more difficult in highly heterogeneous problems.

4. Using the acceleration as a preconditioner for GMRES.

In our final set of numerical experiments, we evaluate the efficiency of the two-grid acceleration with the within-group DSA boost as a preconditioner for the GMRES solver for the multigroup S_N equations. To this end, we solved the same problem as in Section 3., using $32,768$ cores. We prescribed both the residual convergence criterion: $\tau_r < 10^{-6}$, where $\tau_r = \|\tilde{r}\|_2 / \|\tilde{b}\|_2$ and $\tilde{r}$, $\tilde{b}$ are, respectively, the preconditioned residual and preconditioned right hand side of the linear system solved by the GMRES method, as well as the pointwise convergence criterion $\tau_p < 10^{-4}$. The latter was also used for the Jp1+TG+wgDSA run that we performed for comparison purposes. We used the adjusted Hypre parameters from Section 3. for both the two-grid and within-group DSA solves, and PCG residual threshold of 10^{-8} for the convergence of both diffusion solvers. We also re-used the AMG preconditioner from the first thermal group and re-built it in the last thermal group, as described in the previous section. The GMRES restart value was set to 5.

Table VIII shows the iteration numbers for the Jp1+TG+wgDSA iteration and for GMRES preconditioned with the same combination of operators (two-grid and within-group DSA)[3]. As expected, the preconditioning had a significant effect on lowering the GMRES iteration counts. However, when compared to the Jp1+TG+wgDSA iteration, the use of GMRES for this particular problem doesn't bring much benefit. In addition to the group set sweep per every GMRES iteration (when computing the matrix-vector product), there are additional sweeps involved in the restarted GMRES method as it is currently implemented in PDT (when computing the initial residual and right hand side, when checking convergence before restarting and when computing the final solution). Therefore, even though the number of GMRES iterations for the thermal group set is lower than the corresponding number for the Jp1+TG+wgDSA iteration (i.e., the number of up-scattering iterations), the number of thermal group set sweeps is actually higher and so is the number of two-grid and DSA solves. Note that the total number of sweeps is lower for the GMRES method thanks to the more efficient solution in the fast energy range, leading to the better overall solution time (see Table IX). Nonetheless, our current observations indicate that the preconditioned GMRES method is important for solving highly heterogeneous problems and hence the efficient application of the TG-wgDSA preconditioner for GMRES is the main topic of our ongoing work.

IV. CONCLUSIONS AND OUTLOOK

We have extended the two-grid thermal acceleration technique of Morel et al. based on a Gauss-Seidel approach [1] as follows:

1. Acceleration of Jacobi over thermal groups was carried out. The two-grid technique performs extremely well (spectral radius of ~ 0.65 for Jacobi over groups, versus ~ 0.40 for Gauss-Seidel).

2. We have analyzed the impact of not converging the inner iterations. The two-grid acceleration is severely degraded when the inners are not converged (for both the Gauss-Seidel and the Jacobi approaches). However, we proposed to limit the number of transport solve to a minimum (one) per group but accelerate each transport sweeps with a within-group DSA. The iterative properties of the schemes with the inners fully converged were recovered. This is a significant savings in terms of transport sweeps.

[3] The unpreconditioned GMRES didn't converge in the alloted time (8hrs.) – the final iteration number of the thermal-groupset GMRES was 186 and the corresponding residual 0.003.

TABLE VIII. Iteration counts for the preconditioned GMRES and Jp1+TG+wgDSA methods

method	# sweeps	# th. sweeps	# PCG it.	# GMRES/upscat. it.
GMRES+TG+wgDSA	235	28	25877	20
Jp1+TG+wgDSA	358	25	22863	25

TABLE IX. Timing of the preconditioned GMRES and Jp1+TG+wgDSA methods.

method	Overall	Sweep	PCG Setup	PCG Solve	DSA/Sol
GMRES+TG+wgDSA	8.898_6	5.765_6	91397.0	2.429_6	0.31528
Jp1+TG+wgDSA	9.622_6	6.755_6	81586.0	2.126_6	0.25668

3. For massively parallel transport simulations, enhancing parallel concurrency is necessary. A Jacobi approach over thermal group is preferred and we now have an efficient algorithm to accelerate the thermal upscattering iterations. The methods have been implemented in our massively parallel transport code, PDT. The diffusion solves are based on the DFEM MIP-DSA discretization.

REFERENCES

1. B. ADAMS and J. MOREL, “A two-grid acceleration scheme for the multigroup S_N equations with neutron upscattering,” *Nuclear Science and Engineering*, **115**, *3*, 253–264 (1993).
2. T. M. EVANS, K. T. CLARNO, and J. E. MOREL, “A transport acceleration scheme for multigroup discrete ordinates with upscattering,” *Nuclear Science and Engineering*, **165**, *3*, 292–304 (2010).
3. M. P. ADAMS, M. L. ADAMS, C. N. MCGRAW, , and A. T. TILL, “Provably Optimal Parallel Transport Sweeps with Non-Contiguous Partitions,” in “ANS MC2015-Joint International Conference on Mathematics and Computation (M&C), Supercomputing in Nuclear Applications (SNA) and the Monte Carlo (MC) Method,” (2015).
4. T. S. BAILEY, M. L. ADAMS, B. YANG, and M. R. ZIKA, “A Piecewise Linear Discontinuous Finite Element Spatial Discretization of the S_N Transport Equation for Polyhedral Grids in 3D Cartesian Geometry,” *Journal of Computational Physics*, **227**, 3738–3757 (2008).
5. Y. WANG and J. RAGUSA, “Diffusion Synthetic Acceleration for High-Order Discontinuous Finite Element S_n Transport Schemes and Application to Locally Refined Unstructured Meshes,” *Nuclear Science and Engineering*, **166**, 145–166 (2010).
6. J. C. RAGUSA, “Discontinuous finite element solution of the radiation diffusion equation on arbitrary polygonal meshes and locally adapted quadrilateral grids,” *Journal of Computational Physics*, **280**, 195–213 (2015).
7. B. TURCKSIN and J. C. RAGUSA, “Discontinuous diffusion synthetic acceleration for S_N transport on 2D arbitrary polygonal meshes,” *Journal of Computational Physics*, **274**, 356–369 (2014).
8. R. D. FALGOUT and U. M. YANG, *hypre: A Library of High Performance Preconditioners*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 632–641 (2002).
9. V. E. HENSON and U. M. YANG, “BoomerAMG: A parallel algebraic multigrid solver and preconditioner,” *Applied Numerical Mathematics*, **41**, *1*, 155 – 177 (2002).
10. W. D. HAWKINS, T. S. BAILEY, M. L. ADAMS, P. N. BROWN, A. J. KUNEN, M. P. ADAMS, T. SMITH, N. M. AMATO, and L. RAUCHWERGER, “Validation of Full-Domain Massively Parallel Transport Sweep Algorithms,” *Trans. Amer. Nucl. Soc*, **111**, 699–702 (2014).
11. M. P. ADAMS, M. L. ADAMS, W. D. HAWKINS, T. SMITH, L. RAUCHWERGER, N. M. AMATO, T. S. BAILEY, and R. D. FALGOUT, “Provably Optimal Parallel Transport Sweeps on Regular Grids,” in “International Conference on Mathematics and Computational Methods Applied to Nuclear Science & Engineering,” (2013).
12. Lawrence Livermore National Lab, *HYPRE Reference Manual*.
13. T. GEORGE, A. GUPTA, and V. SARIN, “An Empirical Analysis of the Performance of Preconditioners for SPD Systems,” *ACM Trans. Math. Softw.*, **38**, *4*, 24:1–24:30 (Aug. 2012).
14. A. BIENZ, R. D. FALGOUT, W. GROPP, L. N. OLSON, and J. B. SCHRODER, “Reducing Parallel Communication in Algebraic Multigrid through Sparsification,” *CoRR*, **abs/1512.04629** (2016).
15. R. D. FALGOUT and J. B. SCHRODER, “Non-Galerkin Coarse Grids for Algebraic Multigrid,” *SIAM J. Scientific Computing*, **36** (2014).
16. H. GAHVARI, A. H. BAKER, M. SCHULZ, U. M. YANG, K. E. JORDAN, and W. GROPP, “Modeling the Performance of an Algebraic Multigrid Cycle on HPC Platforms,” in “Proceedings of the International Conference on Supercomputing,” ACM, New York, NY, USA (2011), ICS ’11, pp. 172–181.