Preconditioning a Parallel, Inexact Block-Jacobi Splitting of the S_N Algorithm

J.S. Warsa,* K. Thompson,* J. Morel,*
J. Chang,* K. Budge,* M. Benzi†

*Los Alamos National Laboratory, Los Alamos, NM, 87544,

† Emory University, Atlanta, GA 30322

We explored effectiveness of traditional "β" and modified "β" transport synthetic accelerations (TSA) as preconditioners for a parallel, inexact block-Jacobi algorithm for optically thin problems of Richardson iteration and GMRES iterative solvers. We observed that restarted GMRES stagnates when the restart value is less than the number of processors for thin problems. However, with TSA as a preconditioner, restarted GMRES with the restart value less than the number of processors does not stagnate. We observed that the modified TSA is not only effective, but more efficient than source iteration for large number of processors.

Introduction

Previously, we reported on the effectiveness of Krylov iterative methods for solving the spatially-decomposed S_N equations using a parallel, inexact block-Jacobi splitting of the transport operator (Warsa et al, 2003). It was shown that using the diffusion synthetic acceleration (DSA) to precondition the problem was efficient and scalable. However, DSA was no longer an effective preconditioner for problems that are optically thin on large ($O(10^2)$) numbers of processors. In such cases, the eigenvalue spectrum degrades and convergence of the Krylov method stagnates or slows down to such an extent that it is no longer useful.

Here, we describe how Transport Synthetic Acceleration (TSA) (Ramone et al., 1997) can be used to efficiently precondition an iterative method in the optically thin case. We modified the traditional TSA to make it simpler to implement and to improve performance when TSA is used in conjunction with DSA. We will see that with this preconditioner, simply the parallel, block-Jacobi iteration itself can outperform Krylov acceleration of the split operator.

Parallel, Inexact Block-Jacobi Splitting

The time-independent S_N equations with isotropic scattering for the angular flux $\psi_m(r)$ in the direction $\hat{\Omega}_m$ are

$$\left(\hat{\Omega}_m \bullet \nabla + \sigma_t(r)\right)\psi_m(r) = \frac{1}{4\pi}\sigma_s(r)\phi(r) + q_m(r), \text{ for } m = 1,\ldots,N_A, \tag{1a}$$

where scalar flux $\phi(r)$ is computed using a specified N_A-point quadrature over the unit sphere:

$$\phi(r) = \sum_{m=1}^{N_A} w_m \psi_m(r). \tag{1b}$$

Boundary conditions have been ignored and q_m is an inhomogeneous source term. Equations (1a) and (1b) are discretized in space (centered on a total of N_C spatial locations in the mesh), leading to the following matrix equation for the vector of angular fluxes $\Psi \in \mathbb{R}^{N_A N_C}$

$$L\Psi = MS\Phi + Q, \tag{2a}$$

$$\Phi = D\Psi. \tag{2b}$$

The meanings of the operators in Eqs. (2a) and (2b) are deduced by comparing with Eqs. (1a) and (1b). As is usually the case, the problem can be cast in terms of the scalar fluxes $\Phi \in \mathbb{R}^{N_C}$ by inverting L and operating from the left with D:

$$\Phi = DL^{-1}MS\Phi + DL^{-1}Q, \tag{3a}$$

or

$$\left(I - DL^{-1}MS\right)\Phi = DL^{-1}Q. \tag{3b}$$

The action of the operator L^{-1} is computed with a traditional space-angle sweep. In a spatially decomposed parallel implementation, finding a sweep schedule that is close-to-optimal and scalable (as well as identifying and breaking cycles in the graph) is difficult and often inefficient (Pautz, 2002).

A parallel, inexact block-Jacobi splitting of Eq. (3b) corresponding to a spatially-decomposed problem requires not only the scalar fluxes on the interior of the sub-domains as iterative unknowns in the problem, but the angular fluxes on sub-domain boundaries as well. Only the angular fluxes exiting from mesh cells and spatial centering locations that border the sub-domain boundary, i.e., ψ_m for which $\hat{n} \bullet \hat{\Omega}_m < 0$, where $\hat{n}$ is the outward normal of the sub-domain boundary, need to be considered. These are referred to as the "boundary angular fluxes." The need for parallel sweep schedule is obviated by inverting L on every sub-domain p independently, rather than over the entire domain with a full parallel sweep. Incoming boundary angular fluxes are transferred from neighboring processors to begin the local sweeps and waiting for data is no longer necessary.

In the block-Jacobi splitting, the scalar flux vector is augmented by the boundary angular fluxes and will be symbolized with $\tilde{\Phi}$. Note that the operators M, S and D are block-diagonal and the only part of Eqs. (2a) and (2b) that have terms coupling processor sub-domain is L, which we split into a local part L_0, and a part that couples to the boundary angular fluxes on neighboring processors, L_b. Then the parallel, inexact block-Jacobi splitting can be written as

$$L_0\Psi = \tilde{M}\tilde{S}\tilde{\Phi} + L_b\Psi + Q. \tag{4}$$

Writing (4) in terms of the augmented scalar flux vector and rearranging, this becomes

$$\left(I - \tilde{D}L_0^{-1}\left[\tilde{M}\tilde{S} + L_b W\right]\right)\tilde{\Phi}\,. \tag{5}$$

Notice the addition of the injection operator W and that some of the operators in these last two expressions are augmented. This accounts for the boundary angular fluxes contained in $\tilde{\Phi}$.

The advantage of the parallel, inexact block-Jacobi approach is that reliable and readily-available software for domain decompositions that minimize the surface-to-volume ratios of sub-domains can be used for this parallel algorithm, in contrast to specialized decompositions used with some other parallel sweep scheduling approaches (Pautz, 2002). The disadvantage is that the convergence rates of iterative methods slow with increasing number of processors (Warsa et al, 2003; Yavuz and Larsen, 1989; Yavuz and Larsen, 1992).

A MATLAB implementation of the parallel, block-Jacobi splitting for one dimensional problem has shown quantitatively why the GMRES convergence rate stagnates or degrades to such a point as to be impractical in optically thin problems. As the number of processors increases, the eigen-spectrum spreads over the complex plane and there appears to be a correspondence to the radius of the spectrum and convergence rate, even though the spatial discretization of the transport equation results in a non-normal operator, which is somewhat contrary to the ideas presented in (Campbell et al., 1996).

In Figure 1, we plot the eigenvalue spectrum for source iteration for 2 processors where the width (H) is 128 cm, scattering ratio (c) is 0.999 and the total cross section is 0.0625 cm^{-1}. In Figure 2, we plot the eigenvalue spectrum for the same problem with 128 processors with other properties as Figure 1. We can easily observe that the eigenvalue spectrum spreads out as we increased the number of processors from 2 to 128. This spreading of the eigenvalues will decrease the effectiveness of a GMRES iterative solver. Therefore, our goal is to find a preconditioner that restores convergence of GMRES in such problems, where the resulting iteration scheme clusters the eigenvalues.

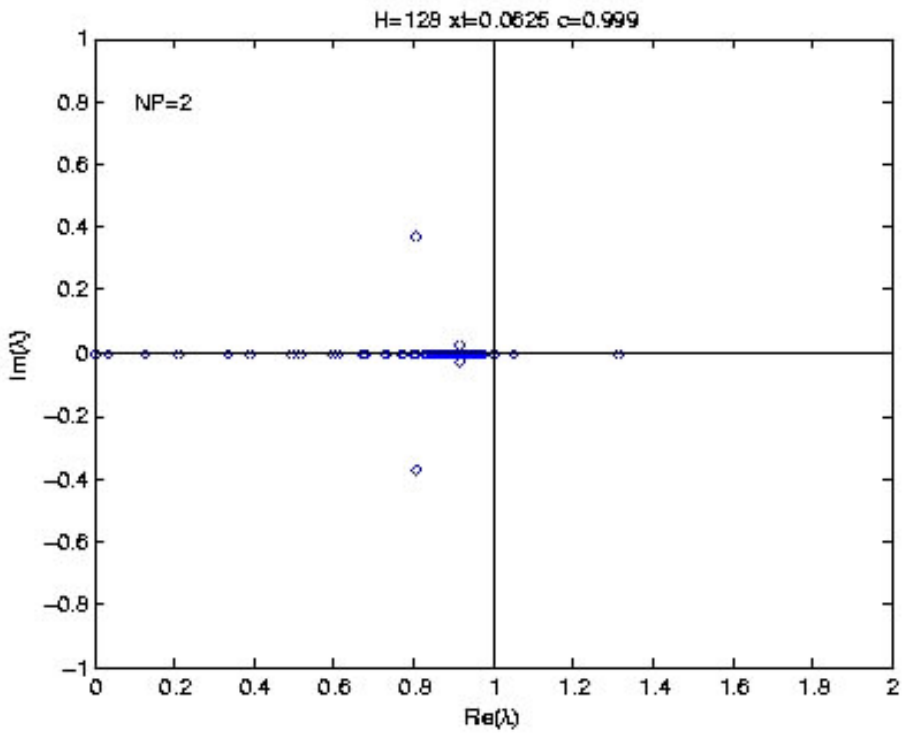

Figure 1. Eigenvalue spectrum for 2 processors.

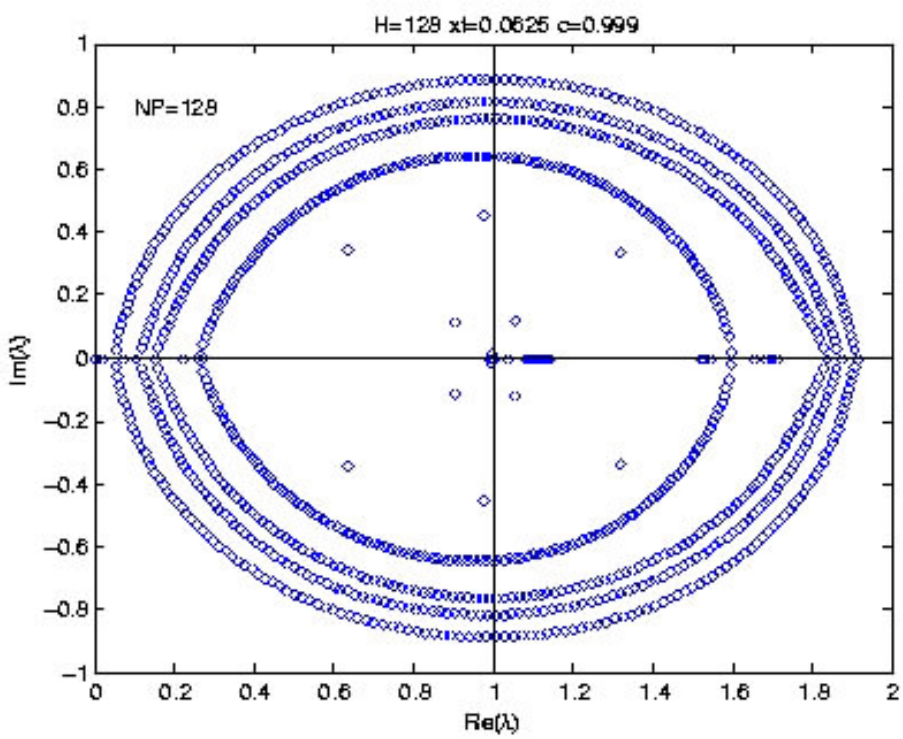

Figure 2. Eigenvalue spectrum for 128 processors.

Modified Transport Synthetic Acceleration

For optically thick problems, a global DSA scheme that includes residuals from the boundary angular fluxes and makes corrections to those angular fluxes, works well when used to precondition a Krylov iterative method (Warsa et al., 2003). However, we observed that when the problem becomes optically thin, convergence of the Krylov method stagnates or slows to such an extent that it is no longer useful. In order to improve convergence, we constructed a preconditioner for Eq. (5) of the form

$$\left(I + QPR\tilde{S}\right) \text{ where } P \approx \left(L - MSD\right)^{-1}, \tag{6}$$

and the operators Q and R represents appropriate projections and interpolations. The preconditioner takes this form because it represents synthetic acceleration (Adams and Larsen, 2002; Kopp, 1963; Warsa et al., 2004). In the case of DSA, P is calculated by solving a diffusion equation. For TSA, P is simply a low-order transport equation. Traditional β-TSA is implemented by letting $\sigma_s' \to (1-\beta)\sigma_s$ and $\sigma_t' \to (\sigma_t - \beta\sigma_s)$ in the low-order operator, where $\beta \in [0,1]$ (Ramone et al., 1997). Both L and S are modified in the low-order equations. Reducing the amount of scattering improves convergence in calculating the inverse in Eq. (6). Modifying L ensures the low-order equations satisfy the same conservation properties as the original system. Typically, a lower order quadrature is used as well to improve efficiency, although this reduces effectiveness to some extent.

In our implementation of TSA, we simply reduce the scattering in the low-order equations by some factor $\beta \in [0,1]$, i.e., $\sigma_s' = \beta\sigma_s$. We have found this to be simpler for implementation purposes, particularly in time and energy dependent calculations. It appears to be just as efficient in most cases. In some cases, this modified β-TSA approach loses effectiveness when the inverse of the low-order equations is not computed to high tolerance, as was done in the original TSA implementation described in (Ramone et al., 1997). However, we observed this is not a serious problem in optically thin problems that is the focus of our study. The modified β-TSA approach has two distinct advantages. First, the traditional method can be unstable when used to precondition source iteration, whereas the modified approach appears to be stable. Second, in problems with scattering ratio ($c = \frac{\sigma_s}{\sigma_t}$) approaching 1, the effective scattering ratio of the low-order problem in the traditional TSA method is a slowly varying function of the parameter β, which means the low-order problem converges slowly unless $\beta \to 1$ and which, in turn, means the low-order equations do not approximate the high-order equations as closely. The modified β-TSA approach has an effective scattering ratio of βc, which means the low-order equations can more closely approximate the high-order problem while converging quickly.

We observed that while either approach can be used to effectively precondition the Krylov iterative solution of the parallel, inexact block-Jacobi splitting in optically thin problems, efficiency of the overall computation is less than when the TSA preconditioner is used to accelerate the block-Jacobi iteration itself in such problems. We found that

choosing $\beta = 1$ in the original TSA method or $\beta = 0$ in the modified approach works extremely well in optically thin problems. This is significant because in the absence of scattering, the number of iterations needed to invert the low-order equations on N_p processors scales roughly as $O\left(N_p\right)$, $O\left(N_p^{1/2}\right)$, and $O\left(N_p^{1/3}\right)$ in one, two and three dimensions, respectively. Furthermore, the number of iterations could be less than that if all or some of the domain is optically thick. This observation has some intriguing implications for multi-dimensional applications of the parallel, block-Jacobi splitting because performance should actually improve on large numbers of processors.

In Figure 3, we plot the eigenvalues for source iteration for 32 processors, where H = 2 cm, $\sigma_t = 1/16$ cm^{-1} and $c = 0.999$. As we observed in the previous figure, the eigenvalue spectrum is not well clustered. Therefore, GMRES will have difficulties solving this problem. In Figure 4, we plot the eigenvalues for the modified TSA for three different values of β(0.0, 0.1, 0.5) and with the same properties as Figure 3. There is a dramatic difference between Figures 3 and 4. The eigenvalues from the modified TSA are well clustered and away from the origin. This indicates GMRES will be very efficient in solving this problem. As β increases, the clustering is tighter. However, this requires more iterations in the low-order solution sweeps.

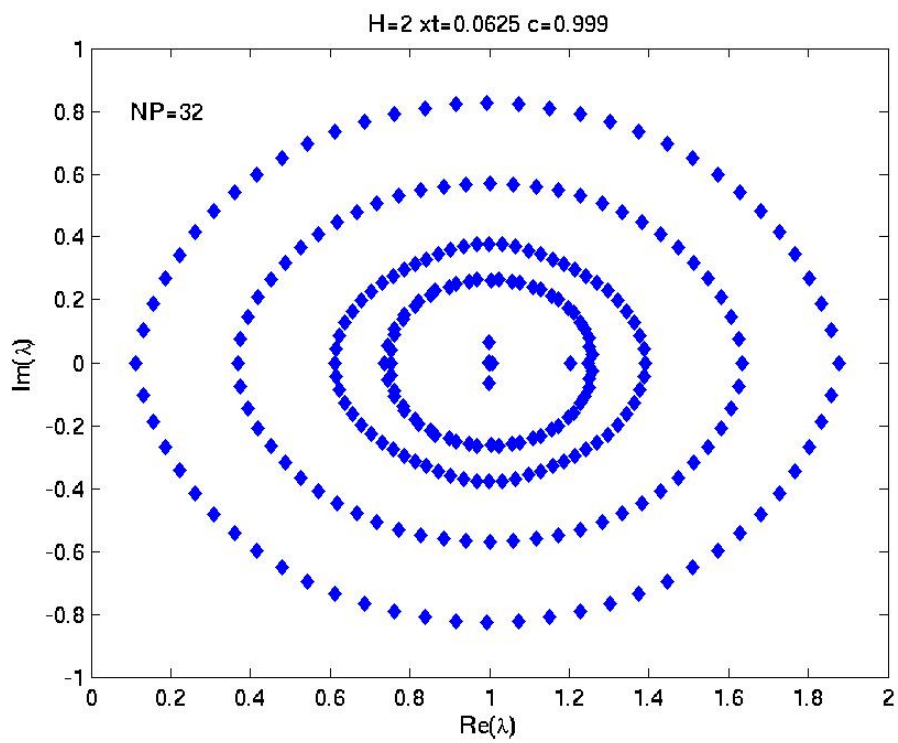

Figure 3. Eigenvalues for source iteration and 32 processors.

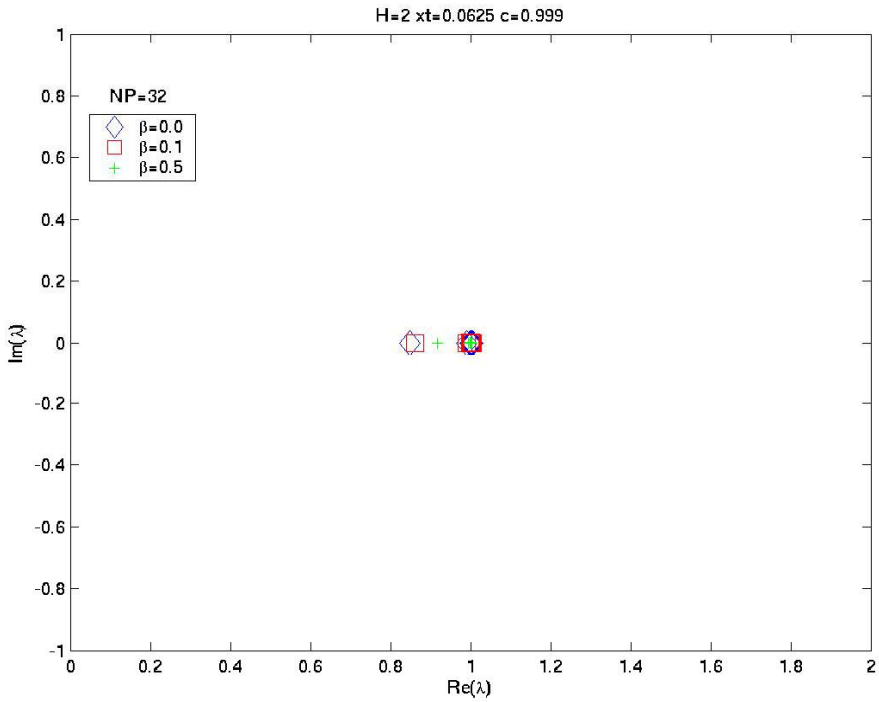

Figure 4. Eigenvalues for modified TSA and 32 processors.

It should be noted that our modified TSA is not effective for serial applications. The Fourier mode analysis of the continuous modified TSA reveals that "flat" mode is not attenuated like the traditional TSA (Ramone et al., 1997). Instead, the eigenvalue for the flat mode is just the scattering ratio squared for the modified TSA when $\beta=0$. This is expected since this is equivalent to two applications of the transport operator or two source iteration sweeps. However, the modified TSA does not diverge, which is the case for the traditional TSA when β is sufficiently large, for any value of β.

Numerical Results

We present a limited set of results that illustrate performance of traditional and modified TSA in one-dimensional slab geometry. The results are necessarily limited because we can explore only a small subset of the large number of possible TSA parameter combinations and problem types.

A code implementation of the two TSA algorithms is used to measure actual computational effort. Measurements are shown for Krylov and source iteration methods, with and without TSA preconditioning, and for a range of values of β. We use the same quadratures for the low and high order problems, in contrasts to what is usually done in a TSA implementation. This eliminates the projection and interpolations that are otherwise needed to accelerate the boundary angular fluxes of the block-Jacobi splitting. The commensurate increase in effectiveness compensates for the cost of using full quadrature in the low-order TSA problem, particularly when there is no scattering in the TSA

operator. The number of iterations and CPU times were measured on 128 processors for two homogeneous problems with 2048 mesh cells per processor and a scattering ratio of $c = 0.999$. Table 1 shows the results for a problem with a total width of 2 cm and $\sigma_t = 1/16$ cm. Table 2 shows the results for a problem with a total width of 128 cm and $\sigma_t = 16$ cm. The sub-domain optical thickness for these two problems are $H = 1/1024$ and $H = 1/16$, respectively. Convergence for the outer iteration is achieved when the residual satisfied $\|r_{outer}\|_2 / \|b\|_2 \le 10^{-5}$. The inner iteration converged when that residual satisfied $\|r_{inner}\|_2 / \|b\|_2 \le 10^{-7}$.

Table 1. Computational measurements for subdomain optical thickness *H = 1/1024*.

TSA Preconditioner			Outer			
			SI		GMRES	
Method	β	Inner	Iterations	CPU (s)	Iterations	CPU (s)
None	n/a	n/a	550	3.3	n/c	n/a
Traditional	1	SI	6	5.9	4	5.1
		GMRES	n/c	n/a	n/c	n/a
	0.98	SI	6	16	4	14.4
		GMRES	n/c	n/a	n/c	n/a
Modified	0	SI	7	6.5	2	3
		GMRES	n/c	n/a	n/c	n/a
	0.5	SI	5	18.7	2	12.6
		GMRES	n/c	n/a	n/c	n/a
	0.95	SI	3	15.1	1	11.2
		GMRES	n/c	n/a	n/c	n/a

Table 2. Computational measurements for subdomain optical thickness *H = 1/16*.

TSA Preconditioner			Outer			
			SI		GMRES	
Method	β	Inner	Iterations	CPU (s)	Iterations	CPU (s)
None	n/a	n/a	3954	36.6	n/c	n/a
Traditional	1	SI	n/c	n/a	19	16.9
		GMRES	n/c	n/a	n/c	n/a
	0.98	SI	n/c	n/a	19	90.8
		GMRES	n/c	n/a	n/c	n/a
Modified	0	SI	369	298	9	10.5
		GMRES	n/c	n/a	n/c	n/a
	0.5	SI	187	366	7	23
		GMRES	n/c	n/a	n/c	n/a
	0.95	SI	22	305	4	75.4
		GMRES	n/c	n/a	n/c	n/a

For thin problems such as these, restarted GMRES stagnates. So, whether GMRES as the inner iterative method (note that the inner iterations are not preconditioned) the problem method does not converge. Of course, this is what originally motivated this

research. As was alluded to earlier, we find the best performance when no scattering is present in low-order TSA operator. In this case, the low-order problem must be solved with source iteration because restarted GMRES stagnates. Nonetheless, it leads to an efficient algorithm because source iteration needs exactly N_p iterations to converge in the absence of scattering. While the timings for source iteration and GMRES outer iterations are similar for different values of β, GMRES performed slightly better. The modified TSA method always performed better than the original TSA method. Finally, we have observed that the optical thickness of the sub-domains is the primary factor affecting convergence.

Next, we present results from parallel scaling tests. The test problem has the following characteristics:

- S_8 quadrature
- Fixed problem width H = 2 cm (equal-width mesh spacing)
- Constant isotropic distributed source q = 1
- Constant material properties: $\sigma_t = 1/16$ cm^{-1}, $c = 0.999$
- Vacuum boundary conditions
- Relative convergence criteria: 10^{-5} (high order) and 10^{-7} (low order)

We compared source iteration, GMRES(15) and GMRES with traditional TSA (β=1.0 and β=0.9) and modified TSA (β=0.0, β=0.5 and β=0.95) as preconditioners.

The scaling tests were conducted on the Q machine at Los Alamos National Laboratory using 1 processor to 1024 processors. First we show the results of weak scaling tests, where the number of cells per processor remained fixed at 65,536. Figure 5 shows the number of iterations as a function of number of processors for each type of solution methods. For both source iteration and GMRES, the number of iterations increased as the number of processors increased. However, for GMRES the problem stagnates when the number of processors was greater than 16. This is expected since the restart value is 15. For all TSA methods, the iteration count remains constant as the number of processors was increased. Figure 6 shows the CPU time as a function of the number of processors, which reveals parallel scaling issues for all methods. Ideally, we would like to have a constant CPU times. However, this is impossible due to communication costs. It should be noted that modified TSA with β=0, was the most efficient out of all the methods when the number of processors was large.

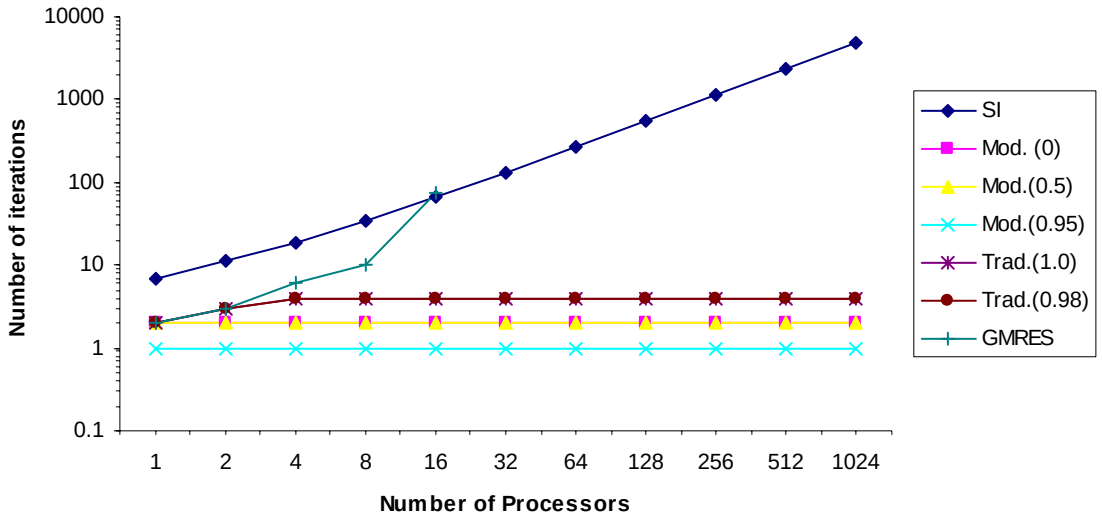

Figure 5. Number of iterations for weak scaling (65,536 cells per processor).

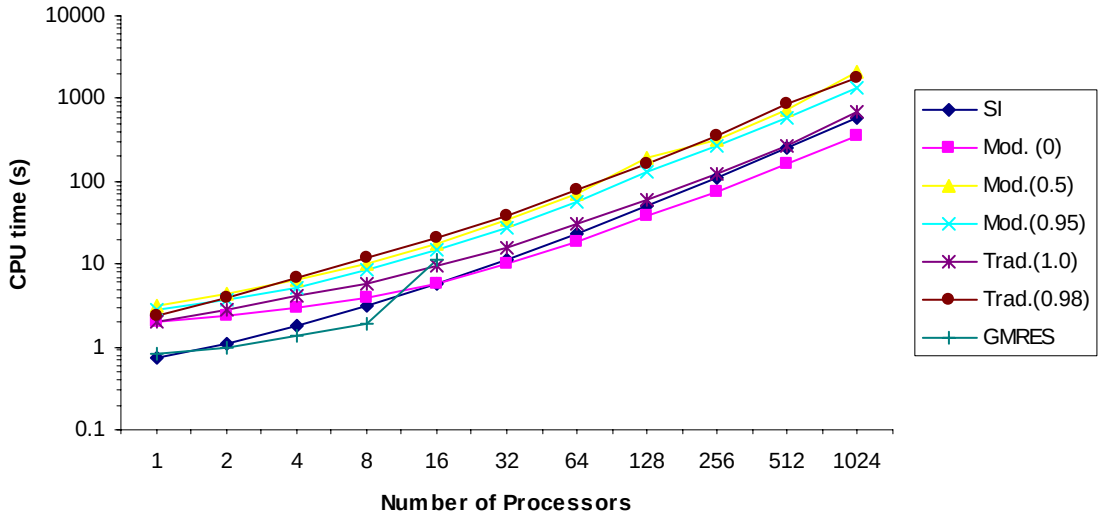

Figure 6. CPU times for weak scaling (65,536 cells per processor).

Figures 7 and 8 are plots of the number of iterations and CPU times for strong scaling tests, where the total number of cells were constant at 2,097,152. The number of iterations (Figure 7) is identical to the results from the weak scaling tests. The CPU time to solve this problem decreased from 1 processor to 128 processors. The CPU time increased as number of processors increased above 128 processors. This is most likely due to communication time between different computer nodes. Once again, modified TSA with β=0, was the most efficient for large number of processors.

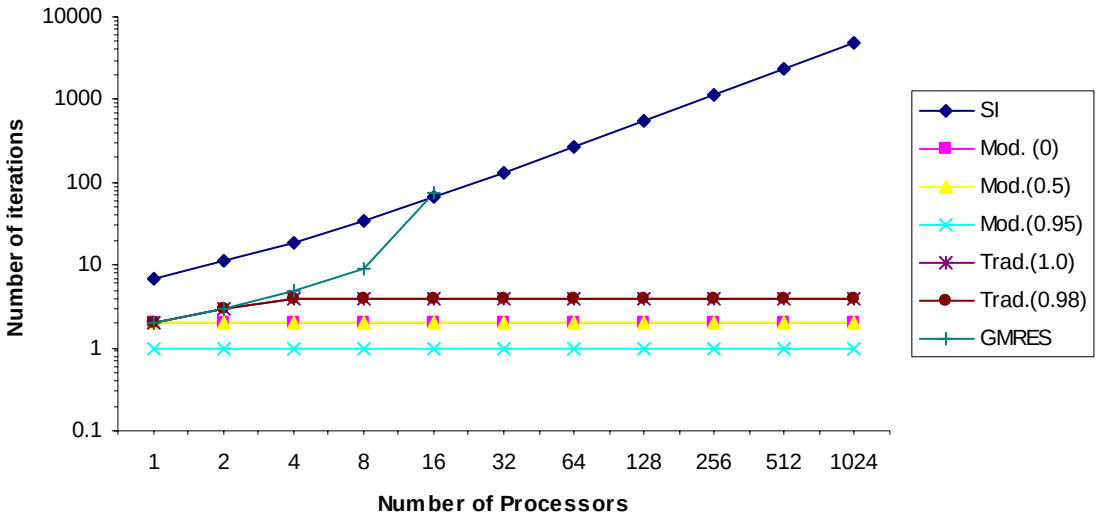

Figure 7. Number of iterations for strong scaling (2^{21} cells).

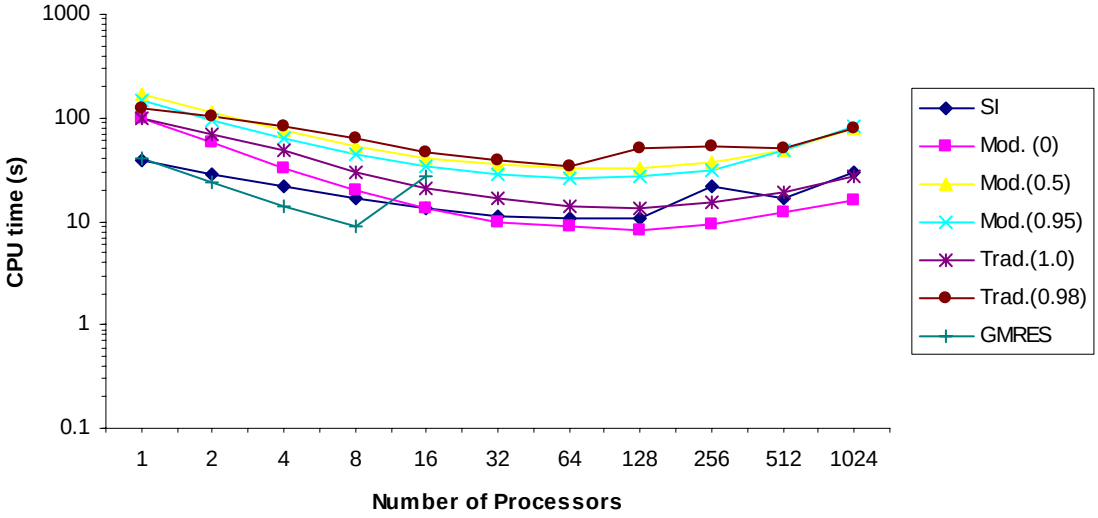

Figure 8. CPU times for strong scaling (2^{21} cells).

Conclusions

We have identified a preconditioner that can restore convergence of an iterative solution of the parallel, inexact block-Jacobi splitting of the transport operator in problems where the operator (without preconditioning) would stagnate. We observed that this preconditioned oeprator is robust and efficient, at least in the homogeneous, one-dimensional problems we have considered so far.

An issue raised by this investigation is one has been remained since TSA was originally introduced. That is, how can the various parameters be chosen reliably so that

TSA remains efficient and effective under a wide variety of circumstances, including the value of β and the quadrature order for the low-order TSA operator. There is an interplay between efficiency and effectiveness in the choice of these parameters that is not well understood, which is particularly true for a Krylov iterative method solution of the parallel, block-Jacobi splitting. Furthermore, we have found that a parallel implementation of restarted GMRES entails enough overhead that multiple applications of the transport operator per GMRES iteration reduce the overall cost of the solution. This is in contrast to serial implementations for which the reduction in iterations does not compensate for the additional applications of the operator at each iteration (Guthrie et al., 1999; Warsa et al., 2004). The number of times that the transport operator should be applied per GMRES iteration is a new free parameter whose effect on efficiency and convergence rate is also not well understood.

We are currently investigating how effective TSA is at preconditioning a multi-dimensional implementation of the parallel, block-Jacobi splitting. A related question is whether the modified β-TSA is effective for full parallel sweeps in multi-dimensions. We hope to explore whether multiple application of the transport operator per iteration reduces computational cost for both the block-Jacobi splitting and full parallel sweeps in multi-dimensions when solving with a Krylov iterative method on parallel platforms. We are currently pursuing an approach whereby the modified TSA method is used to precondition the fully-coupled, energy dependent transport operator, solved in parallel with restarted GMRES. This is in contrast to the usual approach of the classic inner-outer (Gauss-Seidel) iteration in energy.

What is clear from these considerations is that we have raised more questions than we have answered concerning solving and preconditioning the parallel, inexact block-Jacobi splitting of the transport operator. We hope to answer some of these questions in the near future.

Acknowledgements

This work was performed under the auspices of the U.S. Department of Energy at the Los Alamos National Laboratory.

References

Warsa, J.S., Thompson, K, and Morel, J.E., "Improving the Efficiency of Simple Parallel S_N Algorithms with Krylov Iterative Methods," *Trans. of the Am. Nucl. Soc.*, **90**, 217 (2003).

Ramone, G.L., Adams, M.L., and Nowak, P.F., "A Transport Synthetic Acceleration Method for Transport Iterations," *Nucl. Sci. Engr.*, **125**, 257-283, (1997).

Pautz, S., "An Algorithm for Parallel S_N Sweeps on Unstructured Meshes," *Nucl. Sci. Engr.*, **140**, 111-136, 2002.

Yavuz, M. and Larsen, E.W., "Spatial Domain Decomposition for Neutron Transport Problems," *Trans. Theory and Stat. Phys.*, **18**, 205-219, (1989).

Yavuz, M. and Larsen, E.W., "Iterative Methods for Solving X-Y Geometry S_N Problems on Parallel Architecture Computers," *Nucl. Sci. Engr.*, **112**, 32-42, (1992).

Campbell, S.L., Ipsen, I.C.F., Kelley, C.T., and Meyer, C.D., "GMRES and Minimal Polynomial," *BIT*, **36**, 664-675, (1996).

Adams, M.L., and Larsen, E.W., "Fast Iterative Methods for Discrete-Ordinates Particle Transport Calculations," *Prog. Nucl. Energy*, **40**, 3-159, (2002).

Kopp, H.J., "Synthetic Method Solution of the Transport Equation," *Nucl. Sci. Engr.*, **17**, 65-74, (1963).

Warsa, J.S., Wareing, T.A., and Morel, J.E., "Krylov Iterative Methods and the Degraded Effectiveness of Diffusion Synthetic Acceleration for Multidimensional S_N Calculations in Problems with Material Discontinuities," *Nucl. Sci. Engr.*, **147**, 218-248, (2004).

Guthrie, B., Holloway, J.P., and Patton, B.W., "GMRES as a Multi-Step Transport Sweep Accelerator," *Trans. Theory and Stat. Phys.*, **28**, 83-102, (1999).