

# Parametric Study of Parallel Block Jacobi / Source Iteration Hybrid Methods in 2-D Cartesian Geometry and Construction of the Integral Transport Matrix Method Matrices via Green's Functions

Dylan S. Hoagland<sup>1</sup> and Yousry Y. Azmy<sup>1</sup>

<sup>1</sup>NC State University, Department of Nuclear Engineering  
2500 Stinson Drive, Raleigh, NC, 27607, United States

dshoagla@ncsu.edu, yyazmy@ncsu.edu

## ABSTRACT

*Parallel Block Jacobi (PBJ)* [1] is an asynchronous spatial domain decomposition with application in solving the neutron transport equation due to its extendibility to massively parallel solution in unstructured spatial meshes (grids) without the use of the computationally complex and expensive sweeps required by the *Source Iteration (SI)* method in these applications. [2] However, *PBJ* iterative methods suffer a lack of iterative robustness in problems with optically thin cells, [1] which we have previously demonstrated to be a consequence of *PBJ*'s asynchronicity. To mitigate this effect, we have developed multiple *PBJ / SI* hybrid methods which employ a *PBJ* method (*Parallel Block Jacobi - Integral Transport Matrix Method (PBJ-ITMM)*) or *Inexact Parallel Block Jacobi (IPBJ)*) along with *SI*. [3,4] In this work, we perform a parametric study to determine performance of numerous *PBJ / SI* hybrid methods as a function of multiple problem parameters. This parametric study reached 5 main conclusions: 1) our hybrid approach is more effective with *PBJ-ITMM* than with *IPBJ*, 2) for *PBJ-ITMM*, there is a hybrid method that mitigates the aforementioned iterative slowdown in optically thin cells without diminishing the method's potential parallelism in unstructured grids, 3) this hybrid method is most effective in problems with large, continuous regions of very thin cells, 4) the best performing hybrid method consistently executes within a factor of ten slower than current state-of-the-art acceleration methods that are not efficiently extendable to the massively parallel regime, and 5) both *PBJ-ITMM* and *IPBJ* are observed to be viable approaches for our desired applications. In the pursuit of implementing *PBJ-ITMM* in unstructured grids, we conclude with a description of the *Green's Function ITMM Construction (GFIC)* algorithm, which allows for the *ITMM* matrices to be constructed using the pre-existing *SI* sweep algorithm already present in unstructured grid  $S_N$  transport codes.

KEYWORDS: Neutron Transport, Parallel, Parallel Block Jacobi, Integral Transport Matrix Method

## 1. INTRODUCTION

*Parallel Block Jacobi (PBJ)* [1] is a spatial domain decomposition that partitions a spatial mesh into multiple asynchronous sub-domains; asynchronous meaning that the incoming angular fluxes on sub-domain interfaces are lagged by an iteration. Iterative methods that invoke the *PBJ* decomposition are of interest for solving the neutron transport equation because domain decoupling allows for massively parallel solution on unstructured grids without the use of computationally complex and expensive sweep algorithms, as required in the *SI* iterative method. [2] The resulting asynchronicity comes at the cost, however, of a

degradation of iterative performance. Specifically, in an infinite, homogeneous medium, as the optical thickness of a problem's cells tends towards zero, the spectral radii of *PBJ* type methods tend towards unity. [1] Consequently, the number of required iterations grows without bound as the optical thickness of cells decreases, implying lack of iterative robustness. This lack of robustness is a consequence of *PBJ*'s asynchronicity, as sub-domains only exchange information with adjacent sub-domains after each iteration, while distant sub-domains become increasingly coupled with thinning cells.

To mitigate this issue, we have developed numerous *PBJ / SI* hybrid methods [3,4] which utilize the synchronicity of *SI* to improve the convergence rate of a *PBJ* method. For these hybrid methods, all *SI* sweeps must be assumed to be serial when extended to unstructured grids, since avoiding the complexity of synchronous algorithms associated with *SI* in parallel sweeps over unstructured grids is our motivation for studying *PBJ* methods. We may assume parallelism over angle, however, as this does not invoke such complications in unstructured grids. There are two specific *PBJ* methods of interest in our study, *PBJ-ITMM*, and *IPBJ*. [1,3,4] Mathematically, the only difference between these two methods is that *IPBJ* lags the scattering source, while *PBJ-ITMM* does not. Consequently, *IPBJ*'s iterative solution is simply comprised of a local mesh sweep, while *PBJ-ITMM*'s requires a more complex matrix solution. Conceptually, the difference is that *PBJ-ITMM* is able to resolve local scattering within a sub-domain, whereas *IPBJ* cannot resolve this over the course of a single iteration. Both methods are unable to resolve streaming across sub-domain interfaces within a single iteration. There are two varieties of hybrid implementation we have studied, *preconditioning (P)*, and *asynchronous hybrid (AH)*. The former is the execution of a *PBJ* iteration followed by an *SI* iteration, both over the entire spatial domain. The latter requires the execution of a *PBJ* iteration only in regions with cells exceeding a prespecified optical thickness and an *SI* iteration only in regions with cells below this optical thickness. The *AH* method was motivated by the observation that in almost all homogeneous problems, in the preconditioning approach, one method was primarily responsible for the convergence of the problem, with little contribution from the other.

We denote these hybrid methods with 3 sets of letters separated by hyphens. The first letterset indicates the manner in which the methods are combined, *P* or *AH*. The second and third sets indicate the primary and secondary methods respectively; *PI* (*PBJ-ITMM*), *IP* (*IPBJ*), or *SI*. For example, the *asynchronous hybrid* combination of *PBJ-ITMM* and *SI* is abbreviated to *AH-PI-SI*. In addition to the four hybrid methods discussed, preliminary results from the ensuing parametric study motivated the development of a fifth, *AH-PI-IP*. This method uses the linear relationship between *IPBJ* sub-domain size and iterative solution time to increase the sub-domain size in thin cells to be larger than the *PBJ-ITMM* sub-domains, but smaller than the entire optically thin region. To compare the performance of these iterative methods against each other, their individual iterative-method components, and state-of-the-art acceleration techniques, we conduct a parametric study, as discussed in the following section. Then, we present a new algorithm for the construction of the matrix operators required for *ITMM*'s solution that is extendable to unstructured grids.

## 2. PARAMETRIC STUDY OF *PBJ / SI* HYBRID METHODS

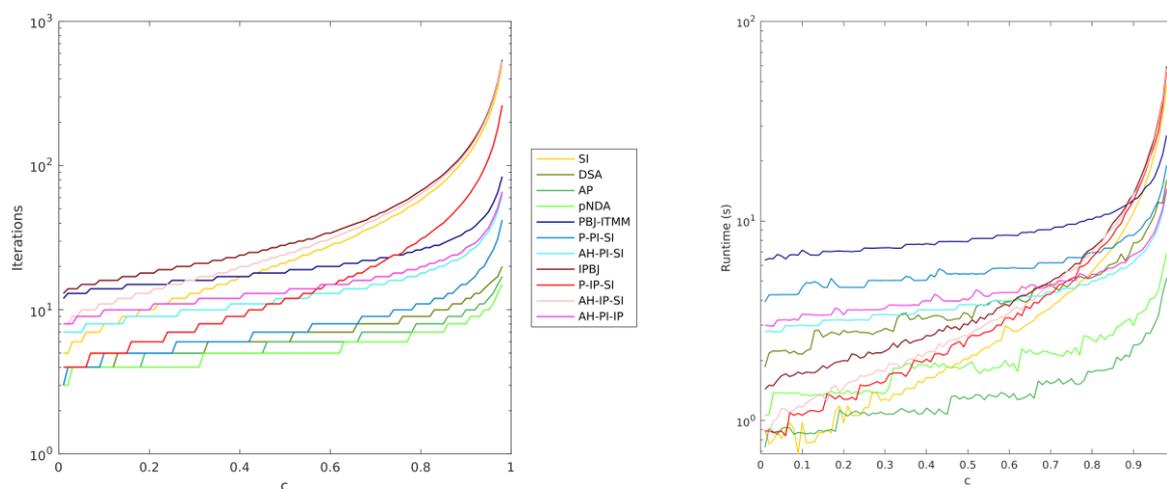
### 2.1. Parametric Study Setup and Scope

To assess the iterative performance of *PBJ / SI* hybrid methods, we conducted a parametric study using a periodic vertical interface problem with the objective of using the required number of iterations and serial execution runtime on simple, 2-D Cartesian grids to inform future implementation in parallel 3-D unstructured grid codes. This problem setup [5] facilitates the study of iterative methods' performances because its heterogeneity allows conclusions from the results to be extended to realistic applications, while employing a simple configuration that allows for the underlying causes of observed trends to be identified. The test problem is a  $128 \times 128$  mesh with uniform square cells of unit side length that are homogeneous in the *y* direction and periodically alternate between optically thin and thick sections in the *x* direction. We

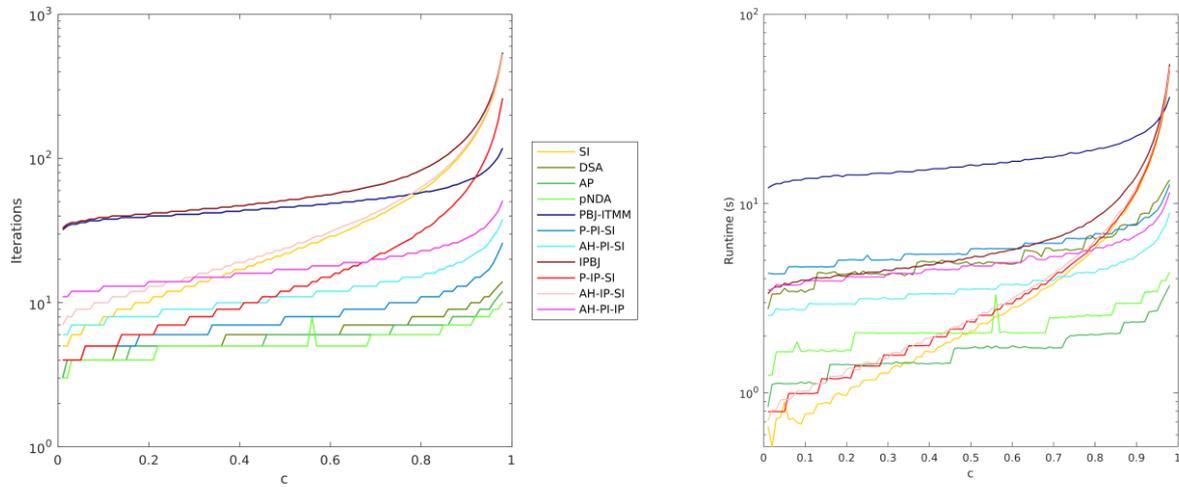
refer to one section of homogeneous cells as a stripe, with  $N_s$  indicating the number of cells in the  $x$  direction per stripe. We then observe the number of iterations and computational runtime required to converge this problem with an  $S_6$  angular quadrature using a variety of methods to a relative stopping criterion of  $10^{-6}$  for varying problem parameters to assess the effect of these parameters on iterative performance of the tested methods. The parameters varied in this study are scattering ratio ( $c$ ), optical thickness of thin ( $\Sigma_{t,thin}$ ) and thick ( $\Sigma_{t,thick}$ ) unit-square cells, and number of cells per stripe ( $N_s$ ). The methods tested include *SI*, *IPBJ*, *PBJ-ITMM*, the 5 previously discussed hybrid methods (*P-PI-SI*, *P-IP-SI*, *AH-PI-SI*, *AH-IP-SI*, and *AH-PI-IP*), and 3 traditional acceleration methods, *Diffusion Synthetic Acceleration (DSA)* [6], *Adjacent-cell Preconditioning (AP)* [7], and *partial-current Nonlinear Diffusion Acceleration (pNDA)* [8]. The HAT-2C test code used to produce the reported results is a 2-D Cartesian geometry, steady-state, one-speed, isotropic scattering, fixed-source  $S_N$  transport solver that uses the *AHOT-N0* [9] spatial discretization method. For *PBJ-ITMM* and *IPBJ* or any hybrid method utilizing one of these as the primary method, we use a  $4 \times 4$  sub-domain size. For *AH-PI-IP*, our objective is to increase the size of the *IPBJ* sub-domains such that the execution time to sweep the *IPBJ* sub-domain for a single angle (since all angles can be executed in parallel) is not longer than the time required to solve one *PBJ-ITMM* sub-domain, thus eliminating any penalty to the potential degree of parallelism incurred by the participant methods in the hybrid scheme. On the employed computational platform we found the maximum *IPBJ* sub-domain size for HAT-2C to satisfy this constraint is  $21 \times 21$ .

## 2.2. Parametric Study Results

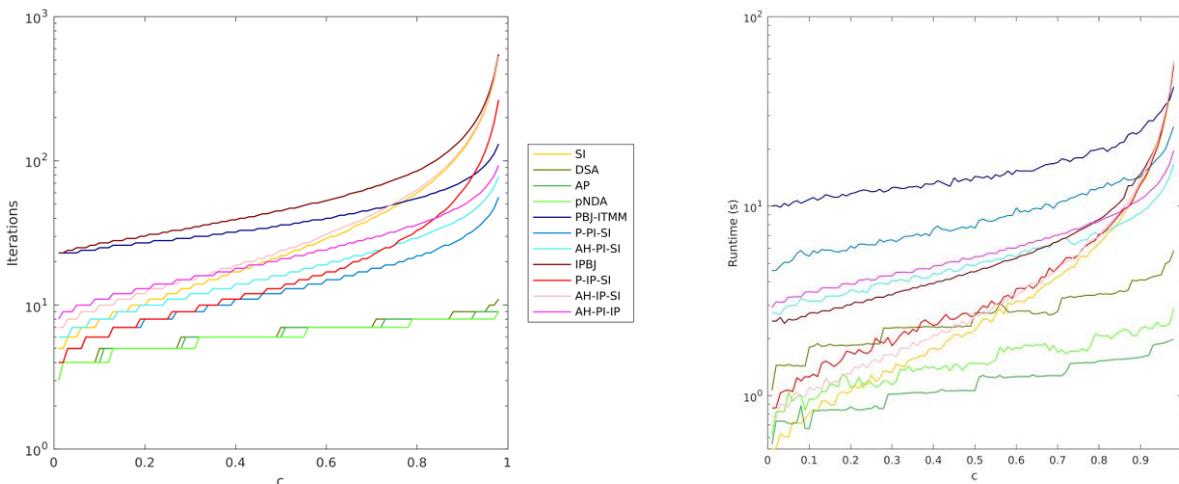
From the parametric study results, Figs. 1 - 6, we are able to draw many conclusions with regards to the iterative performance of *PBJ-ITMM*, *IPBJ*, and our developed hybrid methods deploying one or both of these methods. First, we offer a tip for reading the results. For a given graph, all methods containing *PBJ-ITMM* or *IPBJ* (except *AH-PI-IP*) are in shades of blue or red respectively, with the *PBJ* method alone being the darkest shade, the *SI* preconditioned method a lighter shade, and the *AH* method with *SI* a lighter shade still. The acceleration methods to which we compare our methods' performances (*DSA*, *AP*, and *pNDA*) are in shades of green. Note that a total of 80 cases were analyzed in this parametric study, each combination of the data sets:  $N_s = \{4, 8, 16, 32, 64\}$ ,  $\Sigma_{t,thin} = \{0.001, 0.01, 0.1, 0.5\}$ , and  $\Sigma_{t,thick} = \{1.0, 2.5, 5.0, 10.0\}$ . While, for brevity, only select cases are presented, the discussed conclusions are consistent across the full data set.



**Figures 1 & 2.  $N_s = 4$ ,  $\Sigma_{t,thin} = 0.01$ ,  $\Sigma_{t,thick} = 5.0$**



**Figures 3 & 4.  $N_s = 16, \Sigma_{t,thin} = 0.01, \Sigma_{t,thick} = 10.0$**



**Figures 5 & 6.  $N_s = 16, \Sigma_{t,thin} = 0.1, \Sigma_{t,thick} = 10.0$**

From the parametric study results, the first observation emerges from contrasting the iterative performances of *PBJ-ITMM* and *IPBJ*. In general, we see that *IPBJ* requires shorter runtime than *PBJ-ITMM* until very high scattering ratios, at which point, for some cases, the ability of *PBJ-ITMM* to resolve local scattering compensates for the cheaper cost of an *IPBJ* iteration. While *IPBJ*, *PBJ-ITMM*, nor any hybrid implementation studied are iteratively robust as the scattering ratio approaches unity, it is clear that *PBJ-ITMM*, or hybrid methods containing it have slower increases in the number of iterations as scattering ratio increases compared to their *IPBJ* counterparts. Thus, our results establish *IPBJ* as the faster method in general, but *PBJ-ITMM* is the method less sensitive to scattering ratio. It must also be noted that these methods in serial operation neglect communication costs that would be incurred in our desired parallel applications where both methods have identical communication requirements each iteration: sending and receiving sub-domain interface angular fluxes. Since *IPBJ* is guaranteed to equal or (almost always) exceed the number of iterations required by *PBJ-ITMM*, [1] *IPBJ* will clearly incur greater total communication costs, on average. Additionally, the runtimes for *PBJ-ITMM* include the initial construction and factorization of the *ITMM* matrix operators. In a multi-group problem, this construction and factorization would only occur once per group, during the first outer iteration. In a problem with up-scattering, each

subsequent outer iteration would not require this construction and factorization, thus amortizing their computational cost.

For our hybrid methods, it was previously suggested that *PBJ-ITMM* would benefit more from the *SI* hybrid approach than *IPBJ*. [4] This is due to the physical interpretation of the hybrid approach, namely that *PBJ-ITMM* effectively models localized scattering while *SI* effectively models long distance streaming, the dominant transport phenomena in optically thick and thin cells respectively. *IPBJ*, due to the lagging of the scattering source, does not effectively resolve localized collision, leaving this phenomenon without a method to efficiently converge it. This prediction is confirmed by our parametric study. While *P-IP-SI* and *AH-IP-SI* show significant improvement over *IPBJ* when the scattering ratio is small, these problems are already known to be rapidly convergent. As the scattering ratio becomes large, the inability of *IPBJ* to resolve local collisions becomes more impactful than its asynchronicity and there is very little improvement obtained from the hybrid approach. *PBJ-ITMM* though, is observed to yield, in many cases, a dramatic improvement when the hybrid approach is used. With *PBJ-ITMM*, the performance of these hybrid methods is also seen to be problem dependent though. When the number of cells per stripe becomes small or the optical thickness of thin cells becomes larger, hybrid methods yield only small improvements. This observation initially led to the development of *AH-PI-IP*, with the realization that the convergence rate of *PBJ-ITMM* is ultimately dependent on sub-domain size rather than cell size and that *IPBJ* could, for many cases, accommodate sub-domains large enough to mitigate the iterative slowdown without incurring any penalty to the potential degree of parallelism. This is confirmed by our results, with *AH-PI-IP* requiring only slightly longer runtimes than *AH-PI-SI* in serial execution. Lastly in the comparison of hybrid methods, *AH* methods are consistently observed to converge in shorter runtimes than their preconditioning counterparts, while also maintaining a higher degree of potential parallelism.

The final comparison to be made with the parametric study data concerns the performance of our hybrid methods *versus* current state-of-the-art acceleration methods. We observe that *AP* and *pNDA* converge faster than other methods as the scattering ratio becomes large. However, our best performing hybrid method consistently maintains runtimes within a factor of ten of the best performing acceleration method. This is an encouraging result, as we are interested in *PBJ* and hybrid methods for their application in massively parallel solution on unstructured grids, which traditional acceleration methods are not easily extendable to.

### 3. Green's Function ITMM Construction Algorithm

Based on our parametric study's generated data, we are pursuing the implementation of both *IPBJ* and *PBJ-ITMM* in THOR, an  $S_N$  transport code for unstructured tetrahedral grids. [10] Implementing *PBJ-ITMM* requires an effective algorithm for constructing the *ITMM* matrices on unstructured grids. For this purpose, we have developed the *Green's Function ITMM Construction (GFIC)* algorithm, formulation of which is the subject of this section. To begin this development, we consider a single sub-domain as a full transport problem, since *PBJ-ITMM*'s iterative solution is the fully converged transport solution over a single sub-domain for the current iterate's incoming angular fluxes on the sub-domain boundary. In matrix/vector notation, the transport equation with lagged scattering source is, i.e. source iteration, [11]

$$\boldsymbol{\phi}^{(s+1)} = \mathbf{J}_{\boldsymbol{\phi}}(\boldsymbol{\phi}^{(s)} + \boldsymbol{\Sigma}_s^{-1} \mathbf{q}) + \mathbf{K}_{\boldsymbol{\phi}} \boldsymbol{\psi}_{in}, \quad (1)$$

where  $\boldsymbol{\phi}$  is a vector of cell-averaged scalar fluxes,  $\boldsymbol{\Sigma}_s^{-1}$  is a diagonal matrix of scattering cross section reciprocals,  $\mathbf{q}$  is a vector of cell-averaged external sources,  $\boldsymbol{\psi}_{in}$  is the vector of sub-domain boundary incoming angular fluxes, and  $s$  is the iteration index.  $\mathbf{J}_{\boldsymbol{\phi}}$  and  $\mathbf{K}_{\boldsymbol{\phi}}$  are *ITMM* response matrices that are comprised of elements coupling the current iterate's cell-averaged scalar fluxes to the previous iterate's cell-averaged scalar fluxes and sub-domain boundary incoming angular fluxes, respectively. We begin with

the lagged scattering source in Eq. (1) because it will subsequently allow us to construct the elements of  $\mathbf{J}_\phi$  using mesh sweeps. For the *ITMM* solution however, we take the iterative limit of Eq. (1), yielding, [11]

$$\boldsymbol{\phi} = (\mathbf{I} - \mathbf{J}_\phi)^{-1} (\mathbf{J}_\phi \boldsymbol{\Sigma}_s^{-1} \mathbf{q} + \mathbf{K}_\phi \boldsymbol{\psi}_{in}). \quad (2)$$

To advance the *PBJ-ITMM* iterative process, the outgoing angular fluxes on the sub-domain boundary are needed, as these are exchanged between neighboring sub-domains; solved for using, [11]

$$\boldsymbol{\psi}_{out} = \mathbf{J}_\psi (\boldsymbol{\phi} + \boldsymbol{\Sigma}_s^{-1} \mathbf{q}) + \mathbf{K}_\psi \boldsymbol{\psi}_{in}, \quad (3)$$

where  $\boldsymbol{\psi}_{out}$  is a vector of the sub-domain boundary outgoing angular fluxes, and  $\mathbf{J}_\psi$  and  $\mathbf{K}_\psi$  are *ITMM* response matrices whose elements couple the sub-domain boundary outgoing angular fluxes to the converged cell-averaged scalar fluxes and sub-domain boundary incoming angular fluxes, respectively. One *PBJ-ITMM* iteration is executed by solving Eq. (2) followed by Eq. (3). The vector  $\boldsymbol{\psi}_{in}$  is then updated using  $\boldsymbol{\psi}_{out}$  from neighboring upstream sub-domains and the process is repeated until the stopping criterion on cell-averaged scalar fluxes is satisfied. Execution of this iterative process is contingent upon constructing the four *ITMM* matrices,  $\mathbf{J}_\phi$ ,  $\mathbf{K}_\phi$ ,  $\mathbf{J}_\psi$ , and  $\mathbf{K}_\psi$ . Differentiating Eqs. (1) and (2) yields the following relationships, thus providing prescriptions for the elements of these matrices. [11]

$$\frac{\partial \boldsymbol{\phi}^{(s+1)}}{\partial \boldsymbol{\phi}^{(s)}} = \mathbf{J}_\phi, \quad \frac{\partial \boldsymbol{\psi}_{out}^{(s+1)}}{\partial \boldsymbol{\phi}^{(s)}} = \mathbf{J}_\psi, \quad \frac{\partial \boldsymbol{\phi}^{(s+1)}}{\partial \boldsymbol{\psi}_{in}} = \mathbf{K}_\phi, \quad \frac{\partial \boldsymbol{\psi}_{out}^{(s+1)}}{\partial \boldsymbol{\psi}_{in}} = \mathbf{K}_\psi \quad (4)$$

These matrices were previously constructed using the *Differential Mesh Sweep (DMS)* algorithm, [11] which is a sweep of recursive differentiation. This algorithm, however, requires inverting the transport matrix over a single cell. While trivial in Cartesian grids, this becomes complicated in unstructured grids. As an alternative, *GFIC* utilizes the linearity of the transport equation to cast derivatives as a quotient of discrete differences. With this in mind, we cast a single element of the  $\mathbf{J}_\phi$  matrix as,

$$j_{\phi,(k,j),(k',j')} = \frac{\Delta \phi_{k,j}^{(s+1)}}{\Delta \phi_{k',j'}^{(s)}} = \frac{\left( \phi_{k,j}^{(s+1)} \right)_{GFIC} - \left( \phi_{k,j}^{(s+1)} \right)_0}{\left( \phi_{k',j'}^{(s)} \right)_{GFIC} - \left( \phi_{k',j'}^{(s)} \right)_0}, \quad (5)$$

where  $(k, j)$  and  $(k', j')$  are indices of two spatial cells in 2-D, and the subscripts 0 and *GFIC* represent initial and final states, respectively. Prescribing a “0” state with no external source, vacuum boundary conditions, and a zero initial guess, the two 0-subscript terms vanish from Eq. (5). Then for the *GFIC* state, again with no external source and vacuum boundary conditions, setting  $\left( \phi_{k',j'}^{(s)} \right)_{GFIC} = 1$ , produces,

$$j_{\phi,(k,j),(k',j')} = \left( \phi_{k,j}^{(s+1)} \right)_{GFIC}. \quad (6)$$

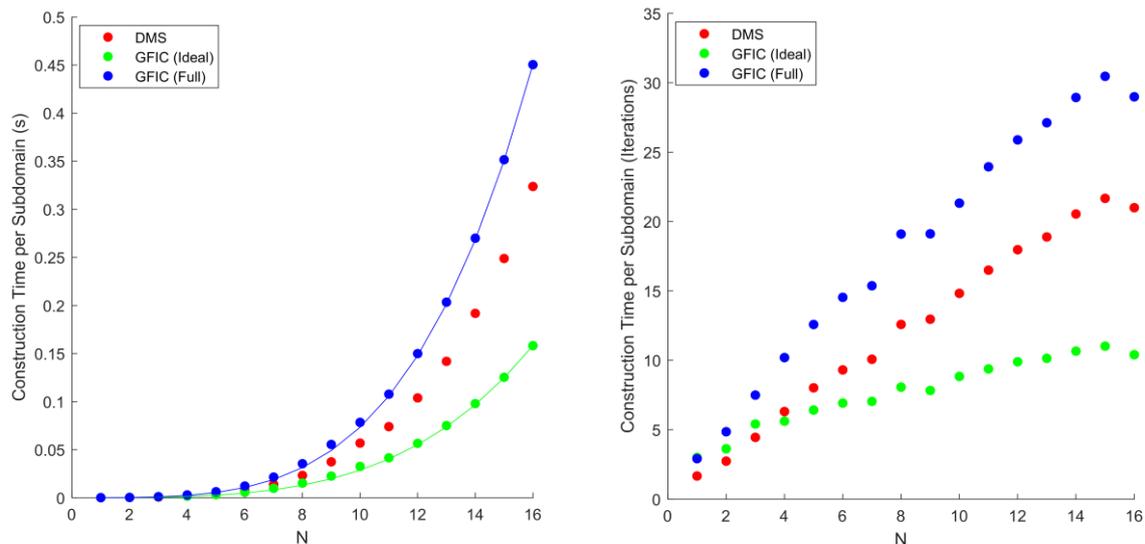
For brevity, we only show this process for  $\mathbf{J}_\phi$ , however, it is analogous for the other response matrices in Eqs. (2) and (3). The resulting *GFIC* algorithm is thus quite simple, even on unstructured grids; for a sub-domain, prescribe a zero external source, vacuum boundary conditions, and a zero initial guess. Then, prescribe a unit initial guess for one cell and execute one *SI*. The resulting cell-averaged scalar fluxes are the elements of one column of  $\mathbf{J}_\phi$  and the resulting sub-domain boundary outgoing angular fluxes are the elements of one column of  $\mathbf{J}_\psi$ . For construction of  $\mathbf{K}_\phi$  and  $\mathbf{K}_\psi$  the process is identical, except unit incoming angular fluxes on the sub-domain boundary are prescribed, rather than the initial guess. Separately

executing this process for each cell and sub-domain boundary face thus constructs all columns of the *ITMM* matrix operators. The *GFIC* algorithm is useful for constructing the *ITMM* matrices on unstructured grids because it utilizes the pre-existing *SI* sweep routine, requiring no new kernel calculation or sweep algorithm.

There are two different implementations of *GFIC* we consider, termed “ideal sweep” and “full sweep”. If we consider a cell  $(k', j')$  with a prescribed unit flux (either cell-averaged scalar or incoming angular), only cells downstream of  $(k', j')$  in the current sweep direction must be solved, as the solution in upstream cells will clearly be zero. This process of only sweeping over upstream cells is termed the ideal sweep, as it avoids solving cells that are not algorithmically necessary. Implementation of *GFIC* using the ideal sweep, however, requires modification of the standard *SI* sweep algorithm, which *GFIC* seeks to avoid. The alternative, the full sweep, therefore sweeps over all cells within the sub-domain, simply calculating values of zero for all quantities in cells upstream from  $(k', j')$ . The simplicity of the full sweep clearly comes at the cost of increased *ITMM* matrix construction time, with the construction times for an  $N \times N$  sub-domain with *GFIC* using the ideal and full sweeps in 2-D Cartesian grids being proportional to,

$$T_{ideal,2D} \propto \frac{N^2(N+1)(N+5)}{4} \quad \text{and} \quad T_{full,2D} \propto N^4 + 2N^3 \quad (7)$$

respectively, with the coefficient of proportionality being dependent on the *SI* grid time. In Figs. 7 and 8, we present the time required to construct the *ITMM* matrices using *DMS* and *GFIC* with both the ideal and full sweeps. Theoretical trendlines produced by Eq. (7) are included, to which the observed construction times show excellent agreement. In addition to the construction time in seconds, we present these results in terms of the number of *PBJ-ITMM* iterations that could be conducted in the time required to construct and factor the *ITMM* matrices. This quantifies the amount by which runtime required for each outer iteration after the first would be reduced in a multigroup problem. While *GFIC* takes longer to construct the *ITMM* matrices with the full sweep than with the ideal sweep, *GFIC* with the full sweep still executes in under 0.5 seconds for all presented sub-domain sizes. *GFIC* with the full sweep was therefore chosen as our desired algorithm for constructing the *ITMM* matrices in unstructured grids, and was consequently used as the construction algorithm for the parametric study presented in this paper. Implementation in THOR for unstructured grids is currently in progress.



**Figures 7 & 8. Construction time of *ITMM* matrices with theoretical trendlines (left) and construction plus factorization time in *PBJ-ITMM* iterations (right) for  $N \times N$  sub-domain**

#### 4. CONCLUSIONS

This work was performed in pursuit of obtaining algorithms for solving the transport equation on unstructured grids that are suitable for massively parallel computers. To alleviate the iterative slowdown of *PBJ* class methods in optically thin cells, we have developed hybrid methods on which we conducted a parametric study. This study concluded that the hybrid methods are effective when used with *PBJ-ITMM*. Based on the reported results, we have concluded that both *PBJ-ITMM* and *IPBJ* are viable options for our desired application, and will hence both be implemented in THOR for massively parallel solution. For this implementation, we have provided *GFIC*, an algorithm designed for constructing the *ITMM* matrices in unstructured grids utilizing only pre-existing capabilities. With this algorithm implemented in THOR, our future work lies in parallelization of THOR to utilize *PBJ-ITMM* and *IPBJ* on a massively parallel scale. This work is currently in the late development stages, using MPI for processor communication. Results from this development will be available in the near future.

#### ACKNOWLEDGMENTS

This material is based upon work by the two authors (DSH and YYA) supported by the Department of Energy National Nuclear Security Administration under Award Number DE-NA0002576.

#### REFERENCES

1. Azmy, Y. Y., Anistratov, D., & Zerr, R. J. (2015). Numerical and Analytical Studies of the Spectrum of Parallel Block Jacobi Iterations for Solving the Weighted Diamond Difference Form of the Sn Equations. Joint Int. Conf. Math & Comp. (M&C), Supercomputing in Nucl. App. (SNA) and the Monte Carlo (MC) method, Nashville, TN, USA.
2. Plimpton, S., Hendrickson, B., Burns, S., & McLendon, W. (2000). Parallel Algorithms for Radiation Transport on Unstructured Grids. Proc. of 2000 ACM/IEEE Conf. on Supercomputing, Dallas, TX, USA.
3. Hoagland, D. S., & Azmy, Y. Y. (2017). Iterative Properties of Parallel Block Jacobi - Integral Transport Matrix Method with Source Iteration Preconditioning. Int. Conf. Math & Comp. Meth. Applied to Nucl. Science & Engineering, Jeju, Korea.
4. Hoagland, D. S., & Azmy, Y. Y. (2019). Fourier Analysis of Inexact Parallel Block Jacobi with Source Iteration Preconditioning and Computational Experimentation with Parallel Block Jacobi / Source Iteration Hybrid Methods. Int. Conf. Math & Comp. Meth. Applied to Nuclear Science & Eng., Portland, Oregon.
5. Azmy, Y. Y. (2002). Unconditionally Stable and Robust Adjacent-Cell Diffusive Preconditioning of Weighted-Difference Particle Transport Methods is Impossible. J. Comp. Phys., 182, 213–233.
6. Azmy, Y. Y. (1985). Nodal Methods for Problems in Fluid Mechanic and Neutron Transport. PhD Dissertation: University of Illinois at Urbana-Champaign.
7. Azmy, Y. Y. (2000). Acceleration of Multidimensional Discrete Ordinates Methods Via Adjacent-Cell Preconditioners. Nucl. Sci. & Eng., 136, 202–226.
8. Cho, N. Z. (2005). Fundamentals and Recent Developments of Reactor Physics Methods. Nucl. Sci. & Eng., 37(1), 25–78.
9. Azmy, Y. Y. (1988). The Weighted Diamond-Difference Form of Nodal Transport Methods. Nucl. Sci. & Eng., 98(1), 29–40.
10. Ferrer, R. M., & Azmy, Y. Y. (2012). A Robust Arbitrarily High-Order Transport Method of the Characteristic Type for Unstructured Grids. Nucl. Sci. & Eng., 172(1), 33–51.
11. Zerr, R. J. (2011). Solution of the Within-Group Multidimensional Discrete Ordinates Transport Equations on Massively Parallel Architectures. PhD Dissertation: Penn State University.