NUMERICAL LINEAR ALGEBRA WITH APPLICATIONS
Numer. Linear Algebra Appl. 2004; **11**:795–811 (DOI: 10.1002/nla.347)

Preconditioning a mixed discontinuous finite element method for radiation diffusion‡

James S. Warsa[1,*,†], Michele Benzi[2], Todd A. Wareing[1] and Jim E. Morel[1]

[1]*Transport Methods Group, Los Alamos National Laboratory, Los Alamos, NM 87545, U.S.A*
[2]*Department of Mathematics and Computer Science, Emory University, Atlanta, GA 30322, U.S.A.*

SUMMARY

We propose a multilevel preconditioning strategy for the iterative solution of large sparse linear systems arising from a finite element discretization of the radiation diffusion equations. In particular, these equations are solved using a mixed finite element scheme in order to make the discretization discontinuous, which is imposed by the application in which the diffusion equation will be embedded. The essence of the preconditioner is to use a continuous finite element discretization of the original, elliptic diffusion equation for preconditioning the discontinuous equations. We have found that this preconditioner is very effective and makes the iterative solution of the discontinuous diffusion equations practical for large problems. This approach should be applicable to discontinuous discretizations of other elliptic equations. We show how our preconditioner is developed and applied to radiation diffusion problems on unstructured, tetrahedral meshes and show numerical results that illustrate its effectiveness. Published in 2004 by John Wiley & Sons, Ltd.

KEY WORDS: radiation diffusion; mixed discontinuous finite element method; indefinite matrices; two-level preconditioning; preconditioned Krylov subspace methods, inner–outer iteration

1. INTRODUCTION

In this paper we describe a preconditioning method for a mixed, discontinuous finite element discretization of the equations describing radiation diffusion. We are interested in solving these equations in order to accelerate the convergence of an outer radiation transport iteration. The discretized transport equation results in linear systems that are far too large to be solved by direct methods. A widely used solution approach in the transport community is source iteration, or transport sweep method, which is a stationary iterative scheme based on a simple splitting of the transport operator; see for example References [1–3]. This iteration can converge quite slowly for certain important classes of problems. In such cases it is impractical to solve the transport problem without some kind of acceleration. The most effective and well-known technique is based on a solution of the radiation diffusion equations. This is known in the neutron transport community as diffusion synthetic acceleration (DSA); see Reference

*Correspondence to: James S. Warsa, CCS-4-MS D409, Transport Methods Group, Los Alamos National Laboratory, Los Alamos, NM 87545, U.S.A.
†E-mail: warsa@lanl.gov

Received 17 August 2001
Revised 19 March 2002
Accepted 20 March 2002

Published online 17 May 2004
Published in 2004 by John Wiley & Sons, Ltd.

[4]. Therefore, an efficient and robust solution of the radiation diffusion equations can make solution to the radiation transport equation feasible in those types of problems.

It is well known that the discretization of the radiation diffusion equations must be *consistent* with the discretization of the transport equations for the acceleration method to be robust, effective and unconditionally stable [5]. In our applications, we use a linear discontinuous finite element spatial discretization of the transport equation on unstructured meshes. Discontinuous transport discretizations are often used for difficult problems because of their robustness and accuracy; see for example Reference [6] and the references therein. Consistency requires us to use a similar discontinuous discretization of the radiation diffusion equations. The second-order operator that describes the steady-state diffusion of radiation is written as a coupled system of first-order equations, referred to as the P_1 equations. The P_1 equations are discretized using an 'upwind' mixed, linear discontinuous Galerkin method, denoted by dG(1), similar to the local discontinuous Galerkin (LDG) method of Cockburn and Shu [7].

The discontinuous P_1 equations form a linear system that can be quite large, even for modestly sized meshes. We found direct solution methods to be impractical for these large linear systems so we use preconditioned Krylov iterative methods instead. Our multilevel preconditioner is based on a linear continuous Galerkin finite element discretization, denoted by cG(1), of the second-order form of the radiation diffusion equation. This lies at the lowest level of the preconditioning technique. A relaxation-based smoother is used before projecting the residual from the discontinuous representation to the lower-order continuous finite element representation which is used as the right-hand side for the approximate iterative solution of the cG(1) linear system with the method of conjugate gradients (CG). The result is then interpolated back onto the discontinuous representation, followed by another relaxation. We have found that our preconditioning technique, while not quite optimal, is very effective. Furthermore, we found that our multilevel preconditioner performs best in just the kind of problems for which acceleration of the transport solution is needed.

We believe that the approach of using a cG(k) method to precondition a mixed dG(k) method can be generalized to $k>1$. Recently, there has been quite a bit of interest in mixed discontinuous Galerkin methods for the solution of fluid flow and other problems (see Reference [8] for a survey) and we would like to emphasize the possibility of using our approach to create efficient preconditioners for these applications.

The work reported here extends our initial implementation in two-dimensional Cartesian geometry [9] to three-dimensional, unstructured tetrahedral meshes using the host transport code AttilaV2 [6]. Application of our preconditioned solution of the discontinuous P_1 equations to the acceleration of the transport iterations in AttilaV2, including analysis and numerical results, is found in Reference [10].

The paper is organized as follows. In the next section, we will present the discretized P_1 equations on tetrahedra. This is followed by a brief discussion of the Krylov iterative methods and other issues for solving our linear systems. The next section presents the details of our multilevel preconditioning approach. Numerical experiments and results follow that and the paper concludes with some summary remarks.

2. RADIATION DIFFUSION EQUATIONS

The radiation diffusion equations, or P_1 equations, are a system of linear first-order equations for the scalar flux and current vector, which are the first two angular moments, respectively,

of the transport angular flux. In steady state, they are equivalent to a second-order diffusion equation, but we must work with the first-order system because the discontinuous discretization cannot be applied directly to the diffusion equation.

The time-independent P_1 equations are

$$\nabla\mathbf{\Phi}(\mathbf{r}) + 3\sigma_t(\mathbf{r})\mathbf{J}(\mathbf{r}) = 3\mathbf{Q}_1(\mathbf{r}) \tag{1a}$$

$$\nabla\cdot\mathbf{J}(\mathbf{r}) + \sigma_a(\mathbf{r})\mathbf{\Phi}(\mathbf{r}) = Q_0(\mathbf{r}) \tag{1b}$$

on some domain $\mathbf{r}\in\Omega$, where $\mathbf{\Phi}(\mathbf{r})$ is the scalar flux and the current vector is $\mathbf{J}(\mathbf{r})$ and $\sigma_t(\mathbf{r})$ and $\sigma_a(\mathbf{r})$ are non-negative functions.

We employ a dG(1) discretization of the P_1 equations on tetrahedral meshes to be consistent with the linear discontinuous discretization used in the host transport code AttilaV2. The trial functions on a mesh cell $T_k\subset\Omega$ are $u_h\in U_h$ and $\mathbf{w}_h\in\mathbf{W}_h$, where U_h and $\mathbf{W}_h$ are both linear finite element spaces (or P_1-P_1). Using Green's Theorem, the discrete variational problem reads:

> find an approximation to the scalar flux and current on a cell T_k, $\mathbf{\Phi}_h\in U_h$ and $\mathbf{J}_h\in\mathbf{W}_h$, respectively, satisfying
>
> $$\int_{\partial T_k}(\hat{n}\mathbf{\Phi}_h^{\mathrm{b}})\cdot\mathbf{w}_h\,\mathrm{d}S - \int_{T_k}\mathbf{\Phi}_h(\nabla\cdot\mathbf{w}_h)\,\mathrm{d}V + 3\sigma_{t,k}\int_{T_k}\mathbf{J}_h\cdot\mathbf{w}_h\,\mathrm{d}V = 3\int_{T_k}\mathbf{Q}_1\cdot\mathbf{w}_h\,\mathrm{d}V \tag{2a}$$
>
> $$\int_{\partial T_k}(\hat{n}\cdot\mathbf{J}_h^{\mathrm{b}})u_h\,\mathrm{d}S - \int_{T_k}\mathbf{J}_h\cdot\nabla u_h\,\mathrm{d}V + \sigma_{a,k}\int_{T_k}\mathbf{\Phi}_h u_h\,\mathrm{d}V = \int_{T_k}Q_0u_h\,\mathrm{d}V \tag{2b}$$
>
> for all $u_h\in U_h$ and $\mathbf{w}_h\in\mathbf{W}_h$.

The parameters $\sigma_{t,k}$ and $\sigma_{a,k}$ are assumed constant on a cell. The linear approximations u_h and $\mathbf{w}_h$ have degrees of freedom at the vertices of each cell; they are defined as the limiting values of the scalar fluxes and current vectors as the vertex is approached from within the cell.

The 'boundary' terms are denoted by superscript b. They enable us to introduce discontinuities into the discretizations. We borrow upwinding techniques from finite volume discretizations used in fluid flow problems to establish expressions for the (particle) flows through the tetrahedral faces. Because such flows are not uniquely defined, we have chosen an upwinding that is based on physical principles [10]. The flows are used to define the boundary terms as follows. If $\hat{n}$ is the outwardly directed unit normal vector of an element face, we let

$$(\hat{n}\mathbf{\Phi}_h^{\mathrm{b}}) = 2\hat{n}((1+\xi)J^+ + (1-\xi)J^-) \tag{3a}$$

$$\hat{n}\cdot\mathbf{J}_h^{\mathrm{b}} = (1-\xi)(J^+ - J^-) \tag{3b}$$

with

$$J^+ = \tfrac{1}{4}\mathbf{\Phi}^+ + \tfrac{1}{2}\hat{n}\cdot\mathbf{J}^+ \tag{3c}$$

$$J^- = \tfrac{1}{4}\mathbf{\Phi}^- - \tfrac{1}{2}\hat{n}\cdot\mathbf{J}^- \tag{3d}$$

where '+' denotes quantities on the interior of the cell face and '−' denotes quantities on the cell adjacent to the cell face. The parameter ξ on a given element face is used to implement the boundary conditions:

$$\xi = \begin{cases} 0 & \text{if the face is internal or on a vacuum boundary} \\ 1 & \text{if face is on a reflective boundary} \end{cases}$$

The difference between our scheme and the LDG scheme of Cockburn and Shu [7] is in how we introduce the discontinuous approximation through the boundary terms of the variational formulation in (2). The definitions of the boundary terms are often referred to as 'numerical fluxes' and several different definitions have been recently put forth; the closest to ours is the LDG scheme. It has been determined recently that using equal order approximations ($k=1$ for both the scalar and vector unknowns) together with our numerical flux definitions gives a convergent and stable discretization [11]. This is in contrast to some LDG methods and other mixed finite element methods; see Reference [12] for an up-to-date, unified treatment of the various discontinuous Galerkin methods, including consistency, stability and convergence issues.

Equations (2) and (3) form an indefinite linear system $\mathbf{H}x = b$, which we can write in symmetric form

$$\begin{bmatrix} \mathbf{A} & \mathbf{B}^{\mathrm{T}} \\ \mathbf{B} & -\mathbf{C} \end{bmatrix} \begin{bmatrix} \mathbf{J} \\ \mathbf{\Phi} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ -g \end{bmatrix} \tag{4}$$

where in our transport acceleration application, $\mathbf{f} = \mathbf{0}$. The block submatrices $\mathbf{A}$ and $\mathbf{C}$ are symmetric positive definite (SPD) and of dimension n and m, respectively. Their size depends on the dimensionality of the system and the type of mesh cells. Here we consider three-dimensional tetrahedral meshes consisting of N cells, for which $n = 12N$ and $m = 4N$.

3. SOLUTION OF THE LINEAR SYSTEM

The solution of linear systems of form (4) has received considerable attention. Linear systems of this type arise in a number of applications including fluid flow, structures, electrical networks, optimization, and others.

We will now present properties of three different formulations of the linear system (4) and candidate solution strategies based on their properties.

3.1. The symmetric indefinite form

System (4) is uniquely solvable, because the coefficient matrix $\mathbf{H}$ is non-singular. Indeed,

$$\begin{bmatrix} \mathbf{A} & \mathbf{B}^{\mathrm{T}} \\ \mathbf{B} & -\mathbf{C} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_n & \mathbf{O} \\ \mathbf{B}\mathbf{A}^{-1} & \mathbf{I}_m \end{bmatrix} \begin{bmatrix} \mathbf{A} & \mathbf{O} \\ \mathbf{O} & -\mathbf{S} \end{bmatrix} \begin{bmatrix} \mathbf{I}_n & \mathbf{A}^{-1}\mathbf{B}^{\mathrm{T}} \\ \mathbf{O} & \mathbf{I}_m \end{bmatrix} \tag{5}$$

where $\mathbf{S} = \mathbf{B}\mathbf{A}^{-1}\mathbf{B}^{\mathrm{T}} + \mathbf{C}$ is the Schur complement of $\mathbf{A}$ in $\mathbf{H}$ (see below). Because $\mathbf{A}$ and $\mathbf{S}$ are SPD, it follows from Sylvester's law of inertia that $\mathbf{H}$ has n positive eigenvalues and m negative ones. Thus, $\mathbf{H}$ is symmetric, non-singular, and indefinite.

It should be pointed out that in most applications, the submatrix $\mathbf{C}$ is symmetric positive semidefinite (rather than SPD), and frequently $\mathbf{C} = \mathbf{O}$, the zero matrix. Nonetheless, most of the techniques and analysis that have been developed for the case of semidefinite $\mathbf{C}$ apply to our case as well.

3.2. The non-symmetric positive definite form

Another property of system (4) that is worth pointing out is the following. By changing the sign of the last m equations in (4) we obtain an equivalent linear system with non-symmetric coefficient matrix

$$\mathbf{H}_+ = \begin{bmatrix} \mathbf{A} & \mathbf{B}^{\mathrm{T}} \\ -\mathbf{B} & \mathbf{C} \end{bmatrix} \tag{6}$$

and it is easy to check that $\mathbf{H}_+$ is positive definite, in the sense that $x^{\mathrm{T}}\mathbf{H}_+ x > 0$ for all real vectors $x \neq 0$. This property implies that all the eigenvalues of $\mathbf{H}_+$ have positive real part. Thus, the convex hull of the spectrum of $\mathbf{H}_+$ does not contain the origin, a desirable property for certain non-symmetric Krylov subspace methods [13]. See Reference [14] for some discussion of symmetry vs non-symmetry in the context of saddle-point problems.

3.3. The Schur complement form

The solution of (4) can be reduced to the solution of two linear systems of smaller size involving the Schur complement matrix $\mathbf{S}$ and the $\mathbf{A}$ block. The Schur complement arises when block Gaussian elimination is carried out on (4), leading to the following system for the scalar flux $\mathbf{\Phi}$:

$$\mathbf{S}\mathbf{\Phi} = (\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^{\mathrm{T}} + \mathbf{C})\mathbf{\Phi} = g + \mathbf{B}\mathbf{A}^{-1}\mathbf{f} \tag{7}$$

which can be interpreted as a discontinuous finite element discretization of the diffusion equation

$$-\nabla \cdot \left(\frac{1}{3\sigma_t(\mathbf{r})} \nabla \mathbf{\Phi}(\mathbf{r}) \right) + \sigma_a(\mathbf{r})\mathbf{\Phi}(\mathbf{r}) = Q_0 - \nabla \frac{1}{\sigma_t(\mathbf{r})} \cdot \mathbf{Q}_1 \tag{8}$$

with suitable boundary conditions. Once the scalar flux $\mathbf{\Phi}$ has been obtained from (7), the current vector $\mathbf{J}$ can be computed by solving

$$\mathbf{A}\mathbf{J} = \mathbf{f} - \mathbf{B}^{\mathrm{T}}\mathbf{\Phi} \tag{9}$$

3.4. Solution strategies

In the present work, we will examine the performance of preconditioned Krylov subspace iterative methods for solving the discontinuous radiation diffusion equations in each of the various forms presented.

Krylov subspace methods appropriate for symmetric indefinite linear systems include the MINRES and SYMMLQ algorithms [15] and the simplified QMR method [16]. For MINRES and SYMMLQ to be used with preconditioning, the preconditioner must be SPD. On the other hand, the simplified QMR method requires only that the preconditioner be symmetric. We ended up choosing MINRES for this formulation, because after extensive testing it performed better and more reliably than the others.

If either the coefficient matrix or the preconditioner is not symmetric then, generally speaking, a non-symmetric Krylov subspace method must be used. The methods we have considered are restarted GMRES, GMRES(m) [17], BiCGStab [18], and TFQMR [19]. We will see in the following sections that the non-symmetric formulation (6) is the more natural one because of the efficiency of our preconditioner in its non-symmetric form.

We should note that TFQMR as well as both the symmetric and non-symmetric versions of QMR, will not be considered further in this paper. This is because after much experimentation we found these methods often exhibited erratic convergence or did not converge, depending on the type of problem being solved and the variant of QMR chosen (even with look-ahead). This non-systematic behaviour is not acceptable for our purposes because the solution of the P_1 equations is itself going to be nested inside one or more other outer iterations. Parameter-free robustness is therefore a requisite in our applications.

Because $\mathbf{S}$ is SPD, CG can be used to solve problems involving (7). In our dG(1) discretization the matrix $\mathbf{S}$ is dense because $\mathbf{A}^{-1}$ is dense, unlike most other LDG methods. It is therefore not practical to form $\mathbf{S}$ explicitly, so at each outer iteration a linear system with coefficient matrix $\mathbf{A}$ is solved using diagonally preconditioned CG (PCG). This is the only viable approach because direct, sparse factorizations of $\mathbf{A}$ proved to be impractical. We will refer to this solution method as PCG–CG (inner PCG, outer CG).

In summary, we will consider the following three solution approaches:

1. Solve (4) with MINRES.
2. Solve (6) with GMRES(m) or BiCGStab.
3. Solve (7) with PCG–CG.

4. PRECONDITIONING

In order to be effective, a preconditioner must take into account the special structure of the coefficient matrix $\mathbf{H}$. A number of preconditioners have been proposed for saddle-point problems, and there is a large literature on this topic; see, e.g. References [20–28] and the references therein.

4.1. The two-level preconditioner

Our approach to preconditioning is motivated by the observation that when a problem is optically thick (total interaction cross-section, σ_t, large) and diffusive (scattering ratio, $c = \sigma_a/\sigma_t$, small), discontinuities in the solution of the P_1 equations disappear (in an integral sense). This is also the case in the limit of increasing mesh refinement. In both of these limits, then, the discontinuous solution on either side of the boundary between adjacent cells approaches the same value, leading us to consider a linear *continuous*, vertex-centred, cG(1), finite element discretization of the scalar diffusion equation (8) to precondition the dG(1) finite element discretization of the P_1 equations. This idea is very much like a two-level multigrid method. The cG(1) finite element space can be thought of as a coarse grid on which the error in the discontinuous dG(1) finite element space (the fine grid) can be approximated. In the limits mentioned above, the cG(1) diffusion equation should therefore do a good job of attenuating low-frequency errors present in the dG(1) scalar fluxes.

Applying the preconditioner is analogous to performing one step of a multigrid V-cycle. On the fine grid, all unknowns ($\mathbf{J}$ and $\mathbf{\Phi}$) undergo (pre-)smoothing. The coarse level consists of projection (restriction) onto the vertices. A correction step on this level follows, which entails solving the cG(1) discretization of (8) with a suitable right-hand side and boundary conditions. This correction is interpolated back onto the fine level, and a step of (post-)smoothing is applied. Applying the preconditioner amounts to performing one step of this V-cycle. We have found that multiple cycles, or multiple pre- or post-smoothing operations in a cycle, are not necessary in that they do not improve convergence enough to justify the additional computation.

While the idea of preconditioning higher order finite elements with elements of lower order is well established, to the best of our knowledge the use of continuous finite elements to precondition discontinuous finite elements as suggested here is new. The preconditioner is displayed in Algorithm 1 for one of the three formulations of our linear system, represented by $\mathbf{F}$. The algorithm may be viewed as 'solving' $\mathbf{M}z = y$. In the symmetric indefinite formulation (4) $\mathbf{F}$ is replaced by $\mathbf{H}$, in the non-symmetric formulation (6) $\mathbf{F}$ is replaced by $\mathbf{H}_+$, and in the Schur complement formulation (7) $\mathbf{F}$ is replaced by $\mathbf{S}$.

Algorithm 1. Two-level preconditioner

$z \leftarrow 0$
$r \leftarrow y - \mathbf{F}z$
$z \leftarrow z + \omega\tilde{\mathbf{F}}^{-1}r$
$r \leftarrow y - \mathbf{F}z$
$z \leftarrow z + \mathbf{G}r$
$r \leftarrow y - \mathbf{F}z$
$z \leftarrow z + \omega\tilde{\mathbf{F}}^{-1}r$

Notice the symmetric pre- and post-smoothing operations, where $\tilde{\mathbf{F}}$ denotes some simple approximation of $\mathbf{F}$. This ensures that the preconditioning *operator* $\mathbf{M}^{-1}$ is non-singular. Although the first few steps could be combined into a single operation, we write it this way to emphasize the V-cycle nature of the algorithm. In the non-symmetric formulation (6) we take $\tilde{\mathbf{F}}$ to be the cellwise block diagonal part of $\mathbf{H}_+$ (damped block-Jacobi smoothing, denoted by [BS]), denoted by $\tilde{\mathbf{F}} = \mathbf{H}_+^{\mathrm{D}}$. For the symmetric formulations (4) and (7) we set $\tilde{\mathbf{F}} = \mathbf{I}$ (Richardson smoothing, denoted by [RS]) for reasons set forth below. In any case, we found that the value $\omega = 0.9$ for the damping parameter gave good results.

The key to the effectiveness of Algorithm 1 is the sequence of operations represented by $\mathbf{G} = \mathbf{P}\mathbf{D}^{-1}\mathbf{R}$. It is based on the cG(1) discretization of the diffusion equation, denoted by $\mathbf{D}$, and the projection (restriction) and interpolation (prolongation) operations, denoted by the rectangular matrices $\mathbf{R}$ and $\mathbf{P}$. The matrix $\mathbf{D}$ is SPD so it can be solved efficiently with PCG or, perhaps optimally, with an algebraic multigrid (AMG) method (or a combination of the two methods). In all cases in this work, we use PCG with diagonal preconditioning to approximately solve linear systems involving $\mathbf{D}$.

The interpolation operation is simple: the discontinuous scalar fluxes surrounding a vertex are all set to the same value of the continuous diffusion solution at that vertex. While it is possible to define more complicated interpolations to attack errors in the currents, we have found that simply leaving the currents alone is adequate and any additional improvement in convergence is minor (and less efficient overall).

The projection operator is relatively more complicated. We find it by deriving the cG(1) discretization of the diffusion equation (8) directly from the dG(1) P_1 equations. We start with the assumption that discontinuous quantities in the dG(1) discretization of the P_1 equations are actually continuous. That is, in (2), we no longer use the upwind definition of the boundary terms given by (3). Instead the boundary terms are given the same values in both cells that share a given face. The remaining within-cell quantities are also assumed to reside on the cell vertices. We combine the moment equations (2a) for cell k to eliminate the currents in favor of the scalar fluxes in the balance equations (2a). The manipulations are carried out with a vector of discontinuous unknowns, r, on the right-hand side of (2). Details of this procedure in two-dimensional rectangular cells may be found in Reference [9]. We are left with four discrete diffusion equations at every vertex on a cell which is the standard cG(1) finite element diffusion equation discretization. The right-hand side that is left defines the 'appropriate' projection from the higher dimensional discontinuous space to the continuous space. Omitting details of the derivation, the final result for a vertex j is

$$\frac{\mathbf{a}_j}{27\sigma_{t,k}V_k}\cdot\left(\sum_{i=1}^{4}\mathbf{a}_i\,\phi_i\right)+\frac{\sigma_{a,k}V_k}{20}\left(2\phi_j+\sum_{\substack{i=1\\ i\neq j}}^{4}\phi_i\right)=r_{j,k}-\frac{\mathbf{a}_j}{3\sigma_{t,k}V_k}\cdot\left(\sum_{i=1}^{4}\mathbf{r}_{i,k}\right) \tag{10}$$

where the vertex-centred unknowns, ϕ_j, are given a global ordering of the mesh vertices based on the cell k and the local index on the cell j. The $r_{j,k}$ terms are the scalar flux components of r at vertex j in cell k, $\mathbf{r}_{i,k}$ are the current vector components, $\mathbf{a}_j$ is the 'area vector' of face j (face area times unit *outward* surface normal) and V_k is the volume of cell k. The matrix $\mathbf{D}$ is assembled (in the usual continuous finite element manner) by summing the contributions to (10) over all cells in the mesh. The form of the right-hand side after summation defines the projection operation.

The preconditioning algorithm represents the operator $\mathbf{M}^{-1}$, which can be written explicitly as

$$\mathbf{M}^{-1}=2\mathbf{I}-\mathbf{F}+(\mathbf{I}-\omega\tilde{\mathbf{F}}^{-1}\mathbf{F})\mathbf{G}(\mathbf{I}-\omega\tilde{\mathbf{F}}^{-1}\mathbf{F}) \tag{11}$$

Because $\mathbf{P}\neq\mathbf{R}^{\mathrm{T}}$, the preconditioner is non-symmetric and the preconditioned system will be non-symmetric whether it is applied to the solution of either (4) or (6). A symmetric preconditioner can be obtained by *defining* $\mathbf{P}:=\mathbf{R}^{\mathrm{T}}$. In solving the Schur complement linear system (7), note that the currents have been eliminated in favor of the scalar fluxes. Therefore, the projections and interpolations are simple, symmetric operations, that is, $\mathbf{P}=\mathbf{R}^{\mathrm{T}}$ where the right-hand side for the linear system assembled from equation (10) involves only scalar flux components $r_{j,k}$. In either formulation, setting $\tilde{\mathbf{H}}=\mathbf{I}$ (Richardson smoothing) enables us to easily estimate and calculate a suitable scaling for (11) that ensures it is SPD, as required by both MINRES and PCG.

4.2. Other approaches

We anticipate that in the future we will have applications involving the radiation diffusion equations that require us to solve for the current vector. We could use the solution for the scalar flux from the reduced system in another preconditioned iteration to compute the current vector. Such an algorithm, which is closely related to Uzawa's method, has been tested and compared against solving the full system by other researchers on related (fluid flow) problems;

see Reference [24], for example. They have found situations where either method computes both components of the solution more efficiently than the other. The overall efficiency depends strongly on the preconditioner.

A few comments are in order regarding other preconditioning strategies. We note that if the reduced Schur complement problem could be solved efficiently (optimally), then the full discontinuous P_1 system could be solved efficiently (optimally) using one of the preconditioners suggested recently by Ipsen [28]; see also Reference [26]. For instance, with the preconditioner

$$\mathbf{M} = \begin{bmatrix} \mathbf{A} & \mathbf{O} \\ \mathbf{O} & \mathbf{S} \end{bmatrix} \tag{12}$$

convergence should be obtained in a very small number of iterations (four, in exact arithmetic), independent of problem size. The cG(1) diffusion equations can be viewed as an approximation to the Schur complement that might be used to replace $\mathbf{S}$ in this expression and lead to a more efficient algorithm overall. This is an interesting area for future exploration. Also, note that none of the algebraically constructed preconditioners we tried displayed anything close to the efficiency of our approach. This included traditional incomplete factorizations and several different variations of sparse approximate inverses and AMG. We found these methods were for the most part ineffective and results with using these methods will not be reported here.

4.3. Inner–outer iteration strategy

All the solution approaches involve an inner–outer Krylov iteration. The inner iteration is the diagonally preconditioned PCG iteration for the cG(1) diffusion equation discretization. In the case of the Schur complement there is an intermediate iteration to compute the action of the matrix block $\mathbf{A}$ on a vector as well. It is possible to improve the overall solution efficiency of inner–outer iterations by altering the convergence tolerance of the PCG iterations. We use an approach for improving the solution efficiency of nested inner–outer Krylov methods first suggested in References [29, 30]. These papers considered GMRES as the outer iteration, inspired by the work in Reference [31] where both inner and outer iterations are CG.

As suggested in Reference [29], we set the inner iteration stopping criterion inversely proportional to the norm of the residual of the outer iteration. We refer to this approach as the 'BFG strategy'. Assuming that at some iteration n the outer residual norm is $r^n > 0$, then we set the inner stopping criterion $\varepsilon_{\text{inner}}$ according to

$$\varepsilon_{\text{inner}} = \begin{cases} \tau\varepsilon & \text{if } n = 0 \text{ (or } n \bmod m = 0) \\ \tau \min(1, \varepsilon/\min(r^n, 1)) & \text{otherwise} \end{cases}$$

where ε is the outer iteration stopping criterion and m is the restart frequency if the outer iteration is GMRES(m). The parameter τ is a scaling factor; a very conservative choice is $\tau \lesssim 1/10$. This approach is somewhat contrary to intuition but we found this approach worked very well. It reduces the number of inner PCG iterations without affecting convergence of the outer iterations. Theoretical justification underlying the empirically observed success of the BFG strategy has been recently developed in Reference [32].

4.4. Summary of preconditioned solution methods

The forms of the linear system and corresponding preconditioned solution methods are summarized here. The list which follows indicates what matrices are to be used in Algorithm 1

when preconditioning.

1. Solve (4) with preconditioned MINRES; $\mathbf{F}=\mathbf{H}$ and $\tilde{\mathbf{F}}=\mathbf{I}$.
2. Solve (6) with preconditioned GMRES(30) or BiCGStab; $\mathbf{F}=\mathbf{H}_{+}$, and $\tilde{\mathbf{F}}=\mathbf{H}_{+}^{\mathrm{D}}$ [BS] or $\tilde{\mathbf{F}}=\mathbf{I}$ [RS].
3. Solve (7) with PCG–PCG $\mathbf{F}=\mathbf{S}$ and $\tilde{\mathbf{F}}=\mathbf{I}$.

The last formulation is denoted by PCG–PCG, indicating the use of PCG for the outer iteration for (7) as well as the inner iteration with the $\mathbf{A}$ block of (7). Recall that in the case that we are not preconditioning then this formulation is referred to as PCG–CG, as in Section 3.

Because the preconditioner itself involves an (inner) iteration, a flexible outer iteration method might be necessary. We compared FGMRES(m) [33] to GMRES(m) for several values of m and found it made little difference in convergence of the non-symmetric formulation. We used an implementation of GMRES(m) with iterated modified Gram–Schmidt orthogonalization. A flexible CG method [34] may or may not improve the performance of the Schur complement formulation, although we have not implemented this method at this time.

5. NUMERICAL RESULTS

We will now present a series of numerical experiments that contrast and compare the performance of the various solution methods with and without preconditioning. In all problems, a random scalar flux source, that is, $g\neq 0$ and $\mathbf{f}=0$, is distributed throughout the problem domains. This is close to the type of sources we can expect to get from transport applications. Other applications would typically have source terms for the vector components $\mathbf{f}$ as well, however.

The first set of numerical experiments is a scaling study on a tetrahedral mesh. The mesh consists of a cube, one cm on a side, divided into f equal-width 'boxes' each dimension. Each box is further divided into six tetrahedra for a total number of cells of $6f^3$. With 16 unknowns per cell, the number of unknowns in the problem scales as $96f^3$. Boundary conditions are reflection on the $x=0$, $y=0$ and $z=0$ faces of the problem and vacuum on the remaining three faces. The total cross-section is $\sigma_t=3.5\,\mathrm{cm}^{-1}$ and the absorption cross-section is $\sigma_a=0.001\times 3.5\,\mathrm{cm}^{-1}$ (i.e. the scattering ratio is $c=0.999$). The problem is solved using the non-symmetric formulation with a relative convergence stopping criterion of 10^{-5} for the GMRES(30) outer iterations. The BFG strategy with $\tau=10^{-2}$ is used to define the PCG inner iteration stopping criterion. Results are shown in Figures 1 and 2 as a function of problem size on a linear scale giving some indication of how the preconditioning method scales. First, note that Figure 1 shows that the preconditioner significantly reduces the number of iterations. This number does grow, albeit slowly, with problem size. Second, Figure 2 shows the number of total floating point operations (FLOP) per unknown. Measurements for solutions without preconditioning are compared against those in which the block Jacobi matrices are factored and stored initially for subsequent use (more efficient) as well as those in which the blocks are factored at every iteration (less storage). A flat curve would be considered computationally optimal. Computing the inverse blocks at every iteration is not competitive. When they are stored and used in the preconditioner, the preconditioned solution is computed with about 25–30% less work per unknown compared to no preconditioning. Overall, however, the preconditioner is not optimal as the work per unknown is also increasing with problem

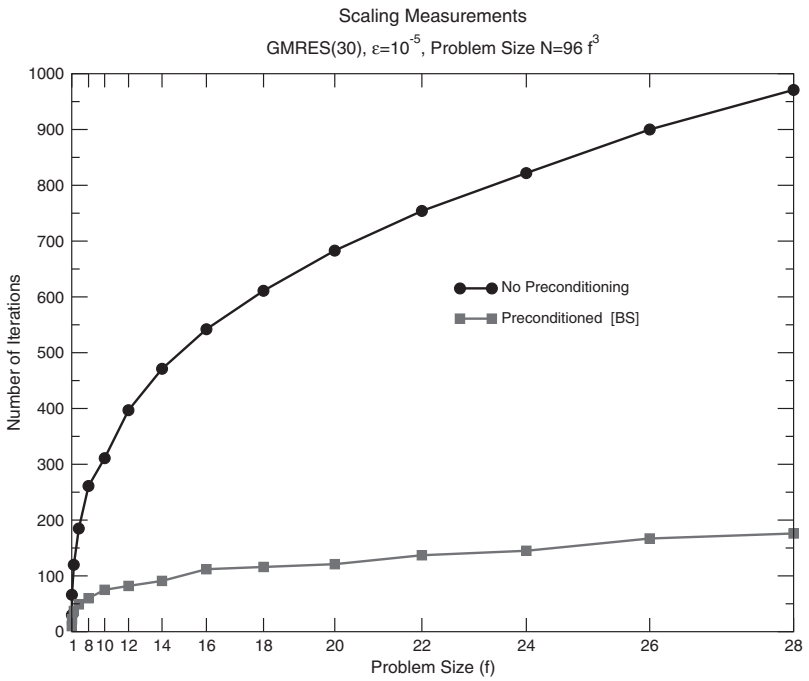

Figure 1. Number of GMRES(30) iterations as a function of problem size.

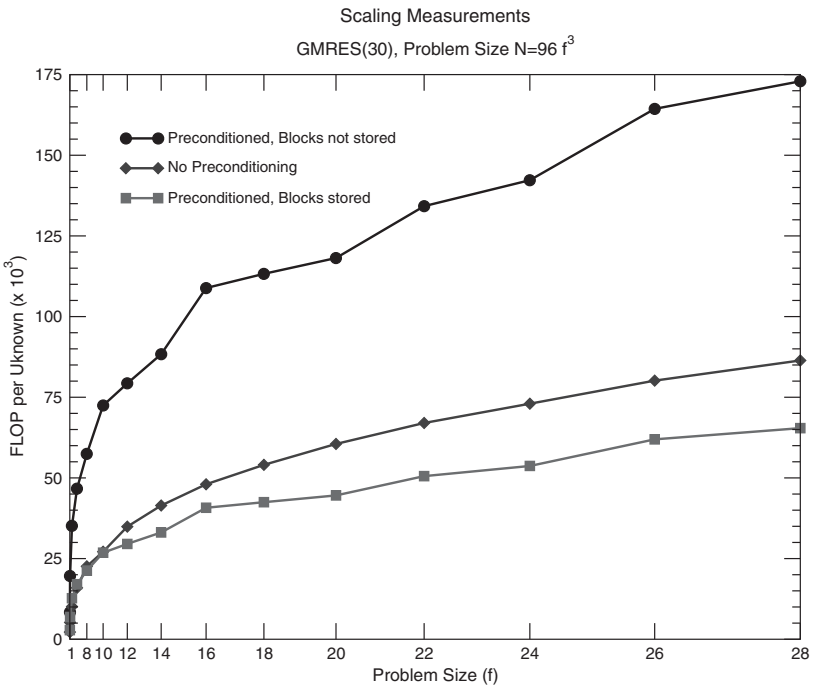

Figure 2. Total floating point operations per unknown as a function of problem size.

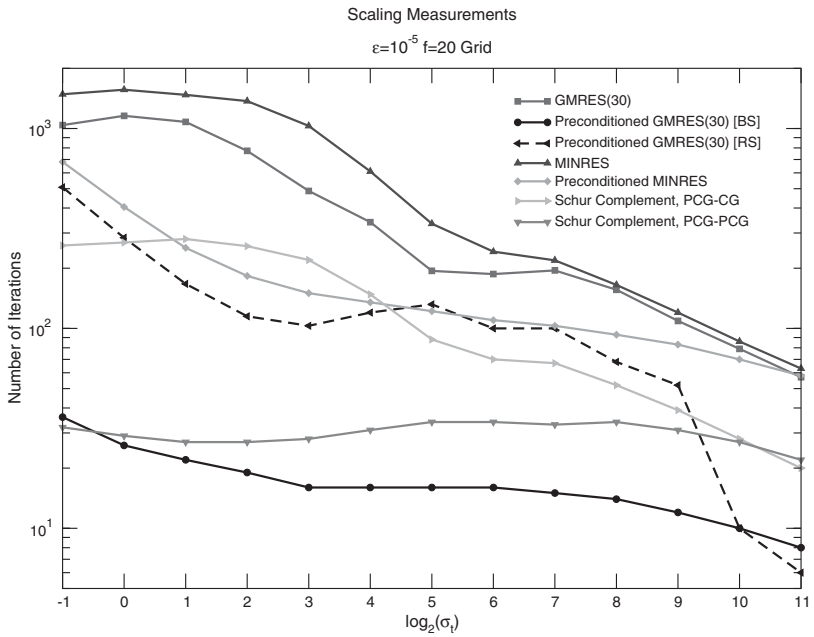

Figure 3. Number of iterations as a function of total cross-section, illustrating how the preconditioner scales with the diffusivity of the problem.

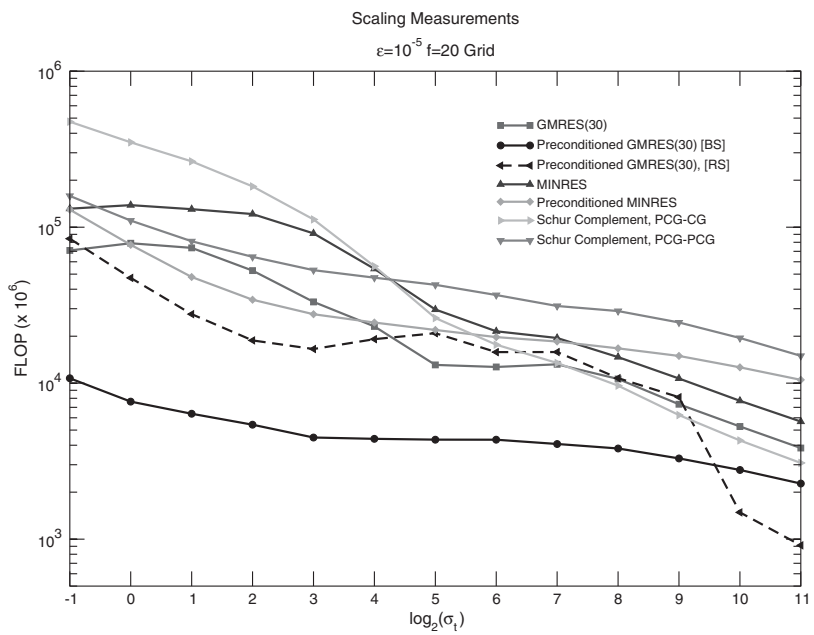

Figure 4. Total floating point operations as a function of total cross-section, illustrating how the preconditioner scales with the diffusivity of the problem.

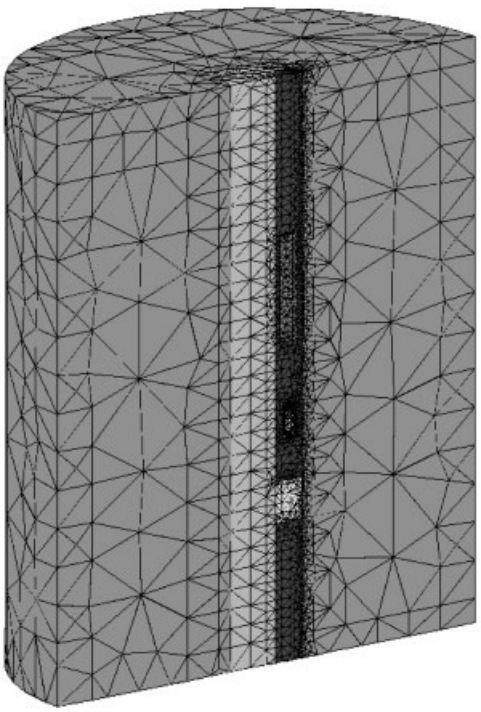

Figure 5. Oil well neutron logging tool mesh, 140 cm tall, 60 cm radius. The bore hole is surrounded by 'earth'. There is an off-centre instrument package that contains the neutron source and two neutron detectors. A reflective symmetry boundary condition is imposed on the half cylinder to reduce computation time.

size. This is in part due to the fact that the inner PCG iterations are not optimal. This could be addressed by implementing an optimal AMG method which should improve the overall optimality of the preconditioned solution.

The next set of numerical experiments have the same problem specifications as before except that we vary the total cross-section σ_t and keep c fixed, to see how the preconditioned solution scales with this parameter. The problem becomes more diffusive and the preconditioner becomes more effective as σ_t increases because the difference in the discontinuous solution between any two adjacent cells disappears (in a weighted integral sense), in which case the solution is better approximated by the continuous diffusion solution. This scaling is also important for our transport applications because it is just such highly diffusive problems for which acceleration is needed. This observation is borne out by the results shown in Figures 3 and 4, where the number of iterations and computational effort measured by FLOP count are shown for a fixed characteristic grid dimension, in this case $h = \frac{1}{20}$. Results are shown for the three formulations, with and without preconditioning.

The preconditioned GMRES(30) solution with block Jacobi smoothing [BS] is computationally efficient and has a faster convergence rate than all the other methods with the exception of the GMRES(30) with the Richardson smoother [RS] in the case of large σ_t. This is because in the limit of very large total cross-section the continuous diffusion solution approximates

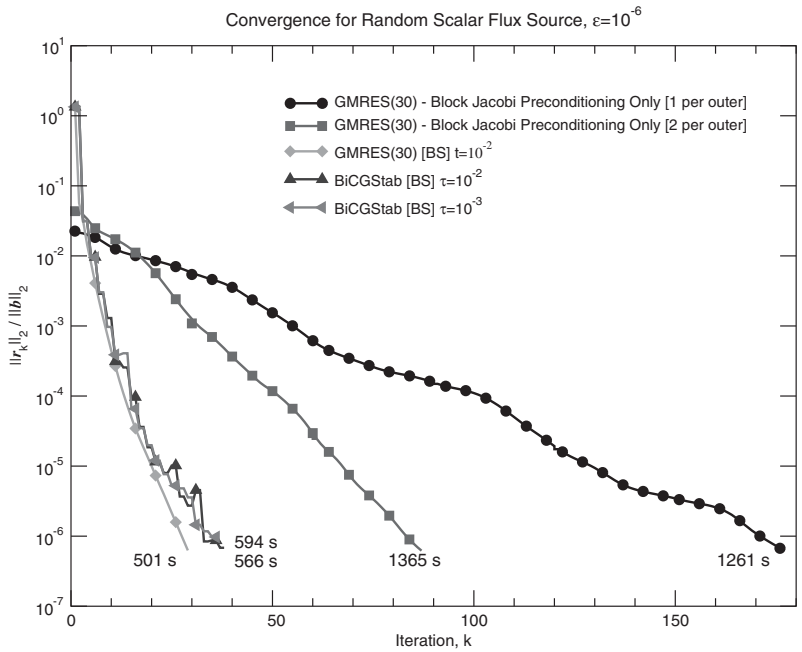

Figure 6. Convergence history of the well logging tool problem illustrating the effectiveness of the preconditioner based on the cG(1) diffusion equation discretization.

the discontinuous solution very well and the only role of the smoother is to keep the preconditioner non-singular.

The final results we show are for a problem with multiple materials and complicated geometry that models an oil well neutron logging tool. It is a reflected half-cylinder, 140 cm tall and 60 cm in radius consisting of 43 012 tetrahedral cells. An illustration of the mesh is shown in Figure 5.

Most of the problem is earth that surrounds the oil well bore hole in the centre of the cylinder. Inside the bore hole, assumed to be filled with air, is the instrument package containing a neutron source and two He-3 neutron detectors. We took this configuration from a full, energy-dependent transport calculation described in Reference [35]. This single energy problem is intended to illustrate how our preconditioner performs on a realistic mesh with realistic cross-sections. In Figure 6 we show the convergence history for the non-symmetric formulation with GMRES(30) and BiCGStab. The CPU times on a 550 MHz Xeon PIII processor are shown alongside the convergence curves to get some idea of the relative computational effort. We also show the results of using only block Jacobi iterations as a preconditioner. In those cases, the preconditioner consisted solely of taking either one or two block Jacobi iterations. This shows that the continuous diffusion solution plays a significant role in preconditioning the iterative solution. Figure 7 shows the convergence history of those methods for which the iterative solution did not converge after 2500 iterations. That indicates preconditioning is absolutely essential for this problem and our experience with even larger problems further indicates that this is a fairly typical result. From this, we conclude that for practical problems we can indeed compute solutions to the dG(1) P_1 equations efficiently.

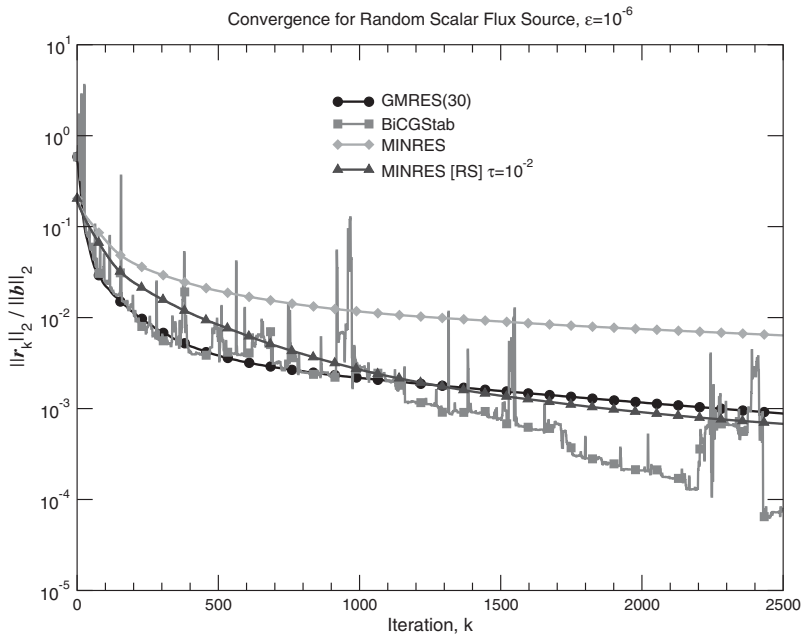

Figure 7. Convergence history of the well logging tool problem illustrating slow convergence without preconditioning and for the symmetric preconditioner (MINRES [RS]).

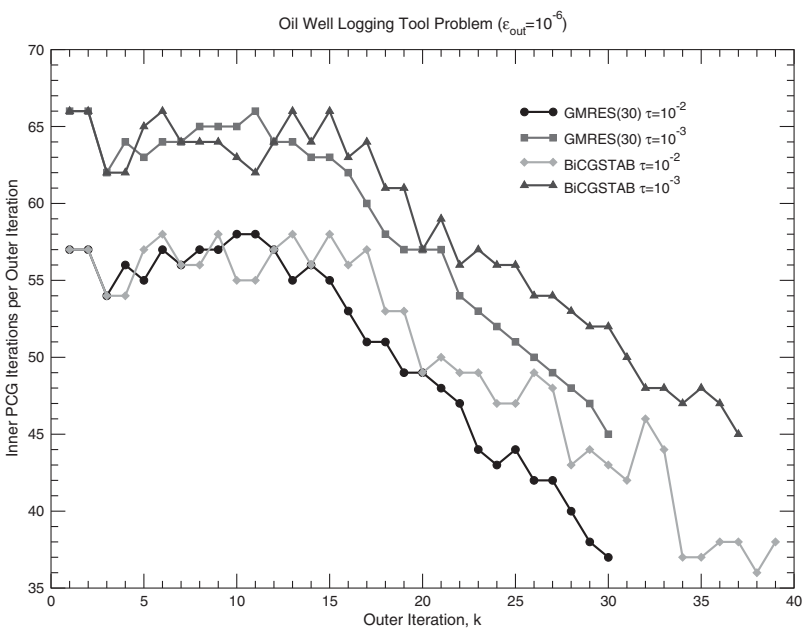

Figure 8. Number of inner CG iterations at each outer iteration, illustrating the use of the BFG strategy.

Finally, we show how the number of inner CG iterations changes over the course of the outer iterations using the BFG strategy with different values of τ in Figure 8.

This illustrates that we can in fact save some computational cycles through the use of this strategy (with no effect on the outer convergence rate whatsoever). While it may not be as important on tetrahedral meshes because there are so many fewer nodes than cells (typically a factor of 4–5) and the CG linear systems are therefore relatively small, this could have a much greater impact on hexahedral meshes, for example.

6. CONCLUSIONS

We have found that our preconditioning method is very effective in reducing the effort in computing iterative solutions of the dG(1) radiation diffusion, or P_1, equations. The preconditioner is based on the relatively small, SPD matrix of the cG(1) diffusion equation discretization, and the inner PCG iterations are therefore also computed fairly efficiently. Numerical results showed that overall the preconditioner is quite efficient and close to optimal. As a result, even though we did not illustrate it here, other algebraic preconditioners could not at this time be considered competitive. Of the possible formulations of the problem, we found that solving the non-symmetric formulation (6) with GMRES and the non-symmetric form of the preconditioner with the block Jacobi smoother was the most reliable and efficient approach.

We would like to again emphasize that our approach is not necessarily limited to the P_1 equations discretized with equal-order mixed dG(1) finite elements. It can most likely be generalized to mixed dG(k) methods for $k>1$, or to other types of physical models and applications, such as the Oseen equations or generalized Stokes flow equations.

ACKNOWLEDGEMENTS

We would like to thank Luc Giraud (CERFACS) for helpful discussions and for the use of his GMRES routines. Thanks also to Ilaria Perugia (University of Pavia) for useful pointers to the literature. This work was performed under the auspices of the U.S. Department of Energy at Los Alamos National Laboratory.

REFERENCES

1. Faber V, Manteuffel TA. A look at transport theory from the point of view of linear algebra. In *Transport Theory*, *Invariant Imbedding and Integral Equations*, Nelson P, Faber V, Manteuffel TA, Seth DL, White AB (eds). Lecture Notes in Pure and Applied Mathematics, vol. 115. Marcel Dekker: New York, NY, 1989; 31–61.
2. Ashby SF, Brown PN, Dorr MR, Hindmarsh AC. A linear algebraic analysis of diffusion synthetic acceleration for the Boltzmann transport equation. *SIAM Journal on Numerical Analysis* 1995; **32**:128–178.
3. Guthrie B, Holloway JP, Patton BW. GMRES as a multi-step transport sweep accelerator. *Transport Theory and Statistical Physics* 1999; **28**:83–102.
4. Alcouffe RE. Diffusion synthetic acceleration methods for diamond-differenced discrete-ordinates equations. *Nuclear Science and Engineering* 1977; **64**:344–355.
5. Larsen EW. The asymptotic diffusion limit of discretized transport problems. *Nuclear Science and Engineering* 1992; **112**:336–346.
6. Wareing TA, McGhee JM, Morel JE, Pautz SD. Discontinuous finite element S_n methods on three-dimensional unstructured grids. *Nuclear Science and Engineering* 2001; **138**:1–13.

7. Cockburn B, Shu CW. The local discontinuous Galerkin finite element method for convection-diffusion systems. *SIAM Journal on Numerical Analysis* 1998; **35**:2440–2463.
8. Cockburn B, Karniadakis GE, Shu CW (eds). *Discontinuous Galerkin Methods*. Springer: Berlin, Heidelberg, 2000.
9. Warsa JS, Wareing TA, Morel JE. Solution of the discontinuous P_1 equations in two-dimensional Cartesian geometry with two-level preconditioning. *SIAM Journal on Scientific Computing*, 2003; **24**:2093–2124.
10. Warsa JS, Wareing TA, Morel JE. Fully consistent diffusion synthetic acceleration of linear discontinuous transport discretizations on three-dimensional unstructured meshes. *Nuclear Science and Engineering* 2002; **141**:1–16.
11. Perugia I, Schötzau D, Warsa JS. A note on a discontinuous Galerkin method for radiation diffusion. *Los Alamos National Laboratory Technical Report LA-UR-02-4754*, Los Alamos, NM, May 2002.
12. Arnold DN, Brezzi F, Cockburn B, Marini D. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM Journal on Numerical Analysis* 2002; **39**:1749–1779.
13. Greenbaum A. *Iterative Methods for Solving Linear Systems*. SIAM: Philadelphia, PA, 1997.
14. Fischer B, Ramage A, Silvester DJ, Wathen AJ. Minimal residual methods for augmented systems. *BIT* 1998; **38**:527–543.
15. Paige CC, Saunders MA. Solution of sparse indefinite systems of linear equations. *SIAM Journal on Numerical Analysis* 1975; **12**:617–629.
16. Freund RW, Nachtigal NM. Software for simplified Lanczos and QMR algorithms. *Applied Numerical Mathematics* 1995; **19**:319–341.
17. Saad Y, Schultz MH. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing* 1986; **7**:856–869.
18. van der Vorst HA. Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing* 1992; **13**:631–644.
19. Freund RW. A transpose-free quasi-minimal residual algorithm for non-Hermitian linear systems. *SIAM Journal on Scientific Computing* 1993; **14**:425–448.
20. Bank RE, Welfert BD, Yserentant HY. A class of iterative methods for solving saddle point problems. *Numerische Mathematik* 1990; **56**:645–666.
21. Elman HC, Golub GH. Inexact and preconditioned Uzawa algorithms for saddle point problems. *SIAM Journal on Numerical Analysis* 1994; **31**:1645–1661.
22. Bramble JH, Pasciak JE. A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems. *Mathematics of Computation* 1988; **50**:1–17.
23. Rusten T, Winther R. A preconditioned iterative method for saddlepoint problems. *SIAM Journal on Matrix Analysis and Applications* 1992; **13**:887–904.
24. Ramage A, Wathen AJ. Iterative solution techniques for the Stokes and Navier–Stokes equations. *International Journal for Numerical Methods in Fluids* 1994; **19**:67–83.
25. Elman HC, Silvester DJ, Wathen AJ. Iterative methods for problems in computational fluid dynamics. In *Iterative Methods in Scientific Computing*, Chan R, Chan T, Golub GH (eds). Springer: Singapore, 1997; 271–327.
26. Murphy MF, Golub GH, Wathen AJ. A note on preconditioning for indefinite linear systems. *SIAM Journal on Scientific Computing* 2000; **21**:1969–1972.
27. Perugia I, Simoncini V. Block-diagonal and symmetric indefinite preconditioners for mixed finite element formulations. *Numerical Linear Algebra with Applications* 2000; **7**:585–616.
28. Ipsen ICF. A note on preconditioning non-symmetric matrices. *SIAM Journal on Scientific Computing* 2001; **23**:1050–1051.
29. Bouras A, Frayssé V. A relaxation strategy for inexact matrix-vector products for Krylov methods. *Technical Report CERFACS TR/PA/00/15*, Toulouse, France, September 2000.
30. Bouras A, Frayssé V, Giraud L. A relaxation strategy for inner–outer linear solvers in domain decomposition methods. *Technical Report CERFACS TR/PA/00/17*, Toulouse, France, September 2000.
31. Golub GH, Ye Q. Inexact preconditioned conjugate gradient method with inner–outer iteration. *SIAM Journal on Scientific Computing* 1999; **21**:1305–1320.
32. Simoncini V, Szyld DB. Theory of inexact Krylov subspace methods and applications to scientific computing. *SIAM Journal on Scientific Computing*, 2003; **25**:454–477.
33. Saad Y. A flexible inner–outer preconditioned GMRES algorithm. *SIAM Journal on Scientific Computing* 1993; **14**:461–469.
34. Notay Y. Flexible conjugate gradients. *SIAM Journal on Scientific Computing* 2000; **22**:1444–1460.
35. Wareing TA, McGhee JM, Morel JE. ATTILA: A three-dimensional, unstructured tetrahedral mesh discrete ordinates transport code. *Transactions of the American Nuclear Society* 1996; **75**:146–147.