

P-multigrid expansion of hybrid multilevel solvers for discontinuous Galerkin finite element discrete ordinate (DG-FEM-S_N) diffusion synthetic acceleration (DSA) of radiation transport algorithms



B. O'Malley^{a,*}, J. Kópházi^a, R.P. Smedley-Stevenson^b, M.D. Eaton^a

^a Nuclear Engineering Group, Department of Mechanical Engineering, City and Guilds Building, Imperial College London, Exhibition Road, South Kensington, London, SW7 2AZ, United Kingdom

^b AWE PLC, Aldermaston, Reading, Berkshire RG7 4PR, UK

ARTICLE INFO

Article history:

Received 29 December 2016

Received in revised form

27 February 2017

Accepted 10 March 2017

Available online 23 March 2017

ABSTRACT

Effective preconditioning of neutron diffusion problems is necessary for the development of efficient DSA schemes for neutron transport problems. This paper uses P-multigrid techniques to expand two preconditioners designed to solve the MIP diffusion neutron diffusion equation with a discontinuous Galerkin (DG-FEM) framework using first-order elements. These preconditioners are based on projecting the first-order DG-FEM formulation to either a linear continuous or a constant discontinuous FEM system. The P-multigrid expansion allows the preconditioners to be applied to problems discretised with second and higher-order elements. The preconditioning algorithms are defined in the form of both a V-cycle and W-cycle and applied to solve challenging neutron diffusion problems. In addition a hybrid preconditioner using P-multigrid and AMG without a constant or continuous coarsening is used. Their performance is measured against a computationally efficient standard algebraic multigrid preconditioner. The results obtained demonstrate that all preconditioners studied in this paper provide good convergence with the continuous method generally being the most computationally efficient. In terms of memory requirements the preconditioners studied significantly outperform the AMG.

© 2017 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction

A major focus in the development of efficient computational methods to solve S_N neutron transport equations is that of diffusion synthetic acceleration (DSA) (Larsen, 1984). The performance of S_N transport codes which utilise DSA is strongly linked to their ability to quickly and efficiently solve the neutron diffusion equation. Preconditioning of the diffusion problem is therefore vital for a DSA scheme to be effective. This paper studies the preconditioning of a discontinuous Galerkin (DG) diffusion scheme developed by Wang and Ragusa, the MIP formulation, which has been shown to be effective for use within DSA (Wang and Ragusa, 2010).

In (O'Malley et al., 2017) two hybrid multilevel preconditioning methods based on methods developed in (Dobrev, 2007) and (Van

Slingerland and Vuik, 2012) are presented which were shown to effectively accelerate the solution of discontinuous neutron diffusion problems. These preconditioners worked by creating a coarse space of either linear continuous or constant discontinuous finite elements. From this coarse space a preconditioning step of an algebraic multigrid (AMG) preconditioner was used to provide a coarse correction, thus leading to a hybrid multilevel scheme.

Both of these preconditioners were valid only for problems which were discretised with first-order finite elements, but in many finite element problems the use of second-order or higher finite elements is more computationally efficient (Gesh, 1999; Mitchell, 2015). It would therefore be valuable to extend the previously specified preconditioners to apply them to higher order elements. In (Bastian et al., 2012) and (Siefert et al., 2014) P-multigrid is used alongside the linear continuous projection defined in (Dobrev, 2007) and an AMG low-level correction in order to precondition high-order element problems.

This paper uses similar concepts to develop preconditioners that

* Corresponding author.

E-mail address: bo712@ic.ac.uk (B. O'Malley).

use P-multigrid with or without the continuous and constant projections used in (O'Malley et al., 2017), alongside a variety of AMG methods for the low-level correction and for various cycle shapes, in order to produce hybrid multilevel solvers. Their computational performance will be benchmarked against AGMG (Notay, 2010, 2012, 2014; Napov and Notay, 2012) a powerful AMG algorithm.

The preconditioners will be judged not only on the speed of convergence but also on how much memory is required to store them. This consideration is very important in neutron transport codes, especially criticality or eigenvalue problems, as for eigenvalue codes with large numbers of energy groups it is necessary to create and store a preconditioner for every energy group for which DSA is to be used.

2. Method

Much of the methodology used in this paper concerning the generation of coarse spaces is the same as in (O'Malley et al., 2017) so it will only be briefly summarised here.

The neutron diffusion equation is an elliptic partial differential equation obtained through an approximation of the neutron transport equation, eliminating terms involving the neutron current J ($\text{cm}^{-2} \text{s}^{-1}$). For scalar neutron flux ϕ ($\text{cm}^{-2} \text{s}^{-1}$), macroscopic removal cross-section Σ_r (cm^{-1}), diffusion coefficient D (cm) and neutron source S ($\text{cm}^{-3} \text{s}^{-1}$) the steady state mono-energetic form of the neutron diffusion equation at position \mathbf{r} is:

$$\nabla D(\mathbf{r}) \nabla \phi(\mathbf{r}) - \Sigma_r(\mathbf{r}) \phi(\mathbf{r}) + S(\mathbf{r}) = 0 \quad (1)$$

This equation is discretised for DG-FEM using the modified interior penalty (MIP) scheme (Wang and Ragusa, 2010), which is a variation of the symmetric interior penalty (SIP) (Arnold et al., 2002; Di Pietro and Ern, 2012). The MIP variation tends to produce a less well conditioned system of equations than SIP, but provides a solution which is more effective for DSA. A key benefit of SIP and MIP is that they generate a symmetric positive definite system of equations, allowing the conjugate gradient (CG) method to be used when solving them.

In (O'Malley et al., 2017) two methods are described to create a two-level preconditioner for a DG-FEM MIP diffusion scheme with first-order elements, differing in the coarse space which the problem was projected onto. The preconditioners presented in this paper will extend these two-level schemes to work with second-order elements.

The first preconditioner creates the coarse space by projecting from a discontinuous first-order finite element formulation to a continuous one. It will be referred to as the “continuous” preconditioner. In order to describe the projection from the discontinuous to the continuous space take η as a given node within the set of all nodes N and τ as a given element within the set of all elements T , assuming a nodal set. $\bar{\tau}_\eta$ will then be the set of elements sharing the node η and $|\bar{\tau}_\eta|$ is the number of elements within this set. For an arbitrary function ϕ then projection operator $R_{\text{continuous}}$ describing the restriction from Ω to Ω^c is defined as (Dobrev, 2007):

$$R_{\text{continuous}} \phi(\eta) = \frac{1}{|\bar{\tau}_\eta|} \sum_{\tau \in \bar{\tau}_\eta} \phi(\eta)_\tau \quad (2)$$

This projection is formed by performing a simple averaging of all discontinuous element function values at a given node in order to obtain the continuous approximation value. It should be noted that it is possible to use this method on problems containing hanging nodes, but in such cases it is necessary to constrain the shape

function values (Schröder, 2011).

The second preconditioner creates a coarse space by instead projecting from a space of discontinuous first-order finite elements to one of discontinuous zeroth-order finite elements with a single degree of freedom per element, again assuming a nodal set. It will be referred to as the “constant” preconditioner. Here the restriction matrix R_{constant} is defined on element τ where Y_τ represents the set of discontinuous nodes (v) within τ as:

$$R_{\text{constant}} \phi(v) = \frac{1}{|Y_\tau|} \sum_{Y_\tau} \phi(v) \quad (3)$$

2.1. P-multigrid

The two methods presented so far create a coarse approximation of a problem discretised with first-order elements. In order to extend these methods to work on problems with higher order elements it is necessary to define a scheme that can project from second-order elements to first-order and so on. Multilevel methods that use such projections are often referred to as P-multigrid methods (Rønquist and Patera, 1987). It is worth noting that the previously defined “constant” preconditioner is effectively a P-multigrid step, projecting from first-order to zeroth-order. However, in order to keep the two concepts separate, whenever this paper refers to a P-multigrid step it means a restriction from an FEM order which is greater than 1. The results in this paper are extended only as high as second-order elements but P-multigrid may be extended to arbitrarily high-order elements as required.

Fig. 1 illustrates how a p-multigrid coarsening would appear for a regular quadrilateral element from second-order to first-order. It is equivalent to an L_2 projection of the higher order basis functions to the lower order finite element L_2 space. The restriction matrix R for a p-multigrid formulation is obtained by expressing the low-order shape functions as a linear combination of the higher order shape functions. This restriction must be separately calculated for each element type and order.

Using triangular elements as an example, take a reference triangular element which has corners which lie at $(0,0)$, $(0,1)$, $(1,0)$ on the $x-y$ plain. Letting $\lambda = 1 - x - y$ the first-order finite element basis functions for the triangle are:

$$\begin{aligned} N_1^{\text{1st}} &= x \\ N_2^{\text{1st}} &= y \\ N_3^{\text{1st}} &= \lambda \end{aligned} \quad (4)$$

and the second-order basis functions are:

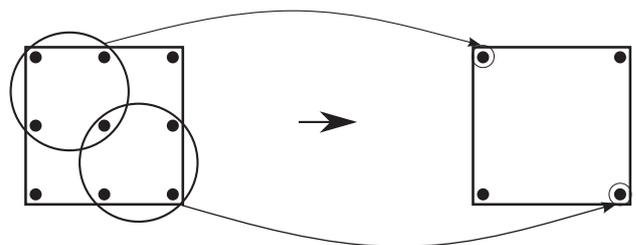


Fig. 1. Projection from second-order quadrilateral element to first-order.

$$\begin{aligned}
N_1^{2\text{nd}} &= x(2x - 1) \\
N_2^{2\text{nd}} &= y(2y - 1) \\
N_3^{2\text{nd}} &= \lambda(2\lambda - 1) \\
N_4^{2\text{nd}} &= 4xy \\
N_5^{2\text{nd}} &= 4x\lambda \\
N_6^{2\text{nd}} &= 4y\lambda
\end{aligned} \tag{5}$$

It can then be shown that:

$$\begin{aligned}
N_1^{1\text{st}} &= N_1^{2\text{nd}} + \frac{1}{2}(N_4^{2\text{nd}} + N_5^{2\text{nd}}) \\
N_2^{1\text{st}} &= N_2^{2\text{nd}} + \frac{1}{2}(N_4^{2\text{nd}} + N_6^{2\text{nd}}) \\
N_3^{1\text{st}} &= N_3^{2\text{nd}} + \frac{1}{2}(N_5^{2\text{nd}} + N_6^{2\text{nd}})
\end{aligned} \tag{6}$$

This defines the P-multigrid projection from the second-order triangle to the first-order, similar projections may be found for other element types.

2.2. Preconditioning algorithm

The preconditioning algorithm is composed of several projections and smoothing steps, as well as a coarse correction. The flow chart in Fig. 2 demonstrates the sequence of restriction and smoothing steps used in order to create the low-level problem which is then passed to the AMG algorithm for a single preconditioning step. After this a similar pattern of smoothing and interpolation steps projects back to the high-level problem so that the preconditioned residual vector may be returned.

A more exact description of the algorithm or a generalised multilevel scheme with N levels now follows. Let $\mathbf{X}^{(n)}$ represent any vector or operator at level n where $1 \leq n \leq N$ with $n = 1$ denoting the coarsest level. The operator $R^{(n) \rightarrow (n-1)}$ represents a restriction from one level to the next coarsest and $I^{(n-1) \rightarrow (n)}$ represents the interpolation back. The system matrix A is projected to a coarser level using the equation:

$$A^{(n-1)} = R^{(n) \rightarrow (n-1)} A^{(n)} I^{(n-1) \rightarrow (n)} \tag{7}$$

Smoothing steps are performed by a block Jacobi smoothing operator $M_{EB}^{(n-1)}$, the inverse of the matrix $M_{EB}^{(n)}$ which consists of the elementwise diagonal blocks of matrix $A^{(n)}$:

$$M_{EB} = \text{diag}_{EB}(A) \tag{8}$$

The smoother will be damped by a scalar value $\omega^{(n)}$ which lies between 0 and 1. Section 3.4 will discuss the selection of values for ω at each level.

Finally on the coarsest level $n = 1$ the error correction must be obtained which requires an approximation of the inverse of $A^{(1)}$. The approximation of this inverse will be represented by the operator $B^{(1)}$ so that $B^{(1)} = \text{approx}[A^{(1)-1}]$. This correction is obtained by using a single preconditioning step of an algebraic multigrid (AMG) preconditioner, discussed further in section 3.2.

Now that all of the pieces of the multilevel preconditioners have been individually described, they will be combined to form a complete preconditioning algorithm. This algorithm will then be used to precondition a conjugate gradient (CG) solver. With a CG solver the preconditioning step involves taking the calculated residual $\mathbf{r}^{(N)}$ of the problem and through application of the preconditioner P^{-1} obtain the preconditioned residual $\mathbf{z}^{(N)}$ such that $\mathbf{z}^{(N)} = P^{-1}\mathbf{r}^{(N)}$. In addition to this the CG solver requires that the matrix to be solved is symmetric positive definite (SPD), this means that the preconditioning algorithm must be designed to also be

SPD.

$$\begin{aligned}
P^{-1}\mathbf{r}^{(N)} = \mathbf{z}^{(N)} &\rightarrow \\
\text{FOR : } [n = N \rightarrow n = 2] & \\
\mathbf{y}_1^{(n)} &= \omega^{(n)} M_{EB}^{(n-1)} \mathbf{r}^{(n)} \text{ (pre-smooth)} \\
\mathbf{y}_2^{(n)} &= \mathbf{r}^{(n)} - A^{(n)} \mathbf{y}_1^{(n)} \\
\mathbf{r}^{(n-1)} &= R^{(n) \rightarrow (n-1)} \mathbf{y}_2^{(n)} \text{ (restriction)} \\
\text{ENDFOR} & \\
\mathbf{z}^{(1)} &= B^{(1)} \mathbf{r}^{(1)} \text{ (coarse level correction)} \\
\text{FOR : } [n = 2 \rightarrow n = N] & \\
\mathbf{y}_2^{(n)} &= I^{(n-1) \rightarrow (n)} \mathbf{z}^{(n-1)} \text{ (interpolation)} \\
\mathbf{y}_3^{(n)} &= \mathbf{y}_1^{(n)} + \mathbf{y}_2^{(n)} \\
\mathbf{y}_4^{(n)} &= \omega^{(n)} M_{EB}^{(n-1)} (\mathbf{r}^{(n)} - A^{(n)} \mathbf{y}_3^{(n)}) \text{ (post-smooth)} \\
\mathbf{z}^{(n)} &= \mathbf{y}_3^{(n)} + \mathbf{y}_4^{(n)} \\
\text{ENDFOR} &
\end{aligned} \tag{9}$$

Equation (9) shows the algorithm for an N level multilevel V-cycle, which is the simplest form of a multilevel cycle (Briggs et al., 2000; Stuben et al., 2001). As previously stated it is vital for effective performance that the preconditioning system is SPD. This is achieved by including a smoothing step before and after each coarse correction (except for $n = 1$), a non symmetric preconditioner would only require a single smoothing step per level. This algorithm is a multilevel variant of the two level algorithm defined in (Van Slingerland and Vuik, 2015).

$$\begin{aligned}
P^{-1}\mathbf{r}^{(N)} = \mathbf{z}^{(N)} &\rightarrow \\
\text{FOR : } [n = N \rightarrow n = 3] & \\
\mathbf{y}_1^{(n)} &= \omega^{(n)} M_{EB}^{(n-1)} \mathbf{r}^{(n)} \\
\mathbf{y}_2^{(n)} &= \mathbf{r}^{(n)} - A^{(n)} \mathbf{y}_1^{(n)} \\
\mathbf{r}^{(n-1)} &= R^{(n) \rightarrow (n-1)} \mathbf{y}_2^{(n)} \\
\text{ENDFOR} & \\
\mathbf{y}_1^{(2)} &= \omega^{(2)} M_{EB}^{(2-1)} \mathbf{r}^{(2)} \\
\text{FOR : } [i = 1 \rightarrow i = J] & \\
\mathbf{y}_2^{(2)} &= \mathbf{r}^{(2)} - A^{(2)} \mathbf{y}_1^{(2)} \\
\mathbf{r}^{(1)} &= R^{(2) \rightarrow (1)} \mathbf{y}_2^{(2)} \\
\mathbf{z}^{(1)} &= B^{(1)} \mathbf{r}^{(1)} \\
\mathbf{y}_2^{(2)} &= I^{(1) \rightarrow (2)} \mathbf{z}^{(1)} \\
\mathbf{y}_3^{(2)} &= \mathbf{y}_1^{(2)} + \mathbf{y}_2^{(2)} \\
\mathbf{y}_4^{(2)} &= \omega^{(2)} M_{EB}^{(2-1)} (\mathbf{r}^{(2)} - A^{(2)} \mathbf{y}_3^{(2)}) \\
\mathbf{y}_1^{(2)} &= \mathbf{y}_3^{(2)} + \mathbf{y}_4^{(2)} \\
\text{ENDFOR} & \\
\mathbf{z}^{(2)} &= \mathbf{y}_1^{(2)} \\
\text{FOR : } [n = 3 \rightarrow n = N] & \\
\mathbf{y}_2^{(n)} &= I^{(n-1) \rightarrow (n)} \mathbf{z}^{(n-1)} \\
\mathbf{y}_3^{(n)} &= \mathbf{y}_1^{(n)} + \mathbf{y}_2^{(n)} \\
\mathbf{y}_4^{(n)} &= \omega^{(n)} M_{EB}^{(n-1)} (\mathbf{r}^{(n)} - A^{(n)} \mathbf{y}_3^{(n)}) \\
\mathbf{z}^{(n)} &= \mathbf{y}_3^{(n)} + \mathbf{y}_4^{(n)} \\
\text{ENDFOR} &
\end{aligned} \tag{10}$$

Equation (10) is the algorithm for the more complex W-cycle. A W-cycle can take many forms, this one restricts to level 2 and then repeats the coarse correction on level 1 a total of J times where J is a parameter that may be chosen by the user. Note that if $J = 1$ then this algorithm is identical to the V-cycle. Again the preconditioner is designed to ensure symmetry. This paper will refer to a cycle where $J = 2$ as a 2W-cycle and so on.

Both the V-cycle and W-cycle algorithms above will be used to form multilevel preconditioners for higher-order DG-FEM SIP diffusion problems. All preconditioners studied will form coarse

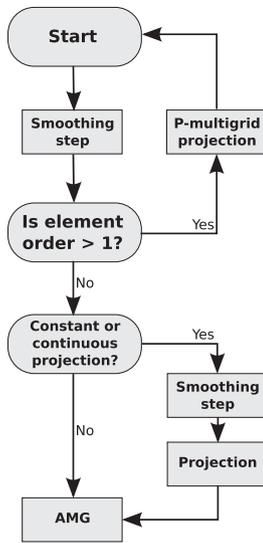


Fig. 2. Flow start for preconditioning algorithm up until low-level AMG correction.

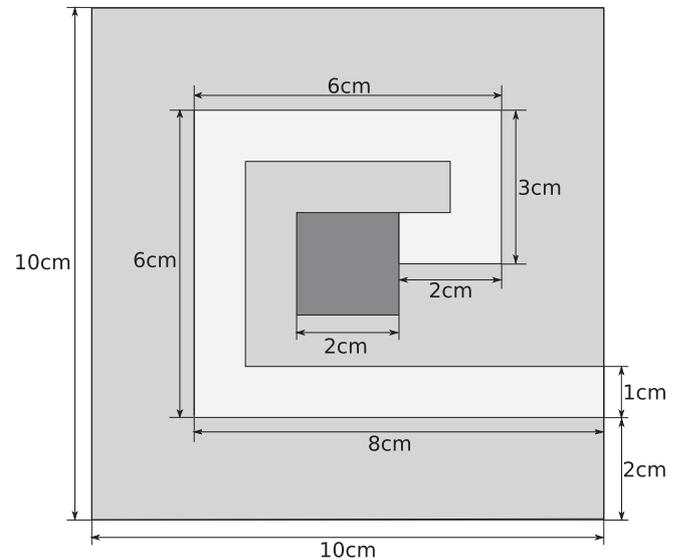


Fig. 3. Visualisation of the 2D cranked duct test problem.

spaces using P-multigrid until the problem has been restricted to a first-order (linear) DG-FEM method. At this point a final coarsening step may be obtained using either the discontinuous piecewise constant or the continuous piecewise linear approximations.

3. Results

3.1. Test cases

In order to study the practical effectiveness of the methods presented so far challenging test problems are required. For this purpose the 2D and 3D cranked duct case which was developed for use in (O'Malley et al., 2017) is used again. The 2D and 3D case both contain a central source region with a prescribed fixed neutron source of $1.0 \text{ cm}^{-3} \text{ s}^{-1}$, a scatter cross-section of 1.0 cm^{-1} and zero absorption. Surrounding the source is a thick region with zero absorption, no neutron source, and a scatter cross-section of $r \text{ cm}^{-1}$. Running from the central source to the boundary of the problem is a cranked duct with zero absorption, no neutron source, and a scatter cross-section of $1/r \text{ cm}^{-1}$. The value r is therefore a parameter which is used to control how heterogeneous the problem is, with $r = 1.0$ yielding a homogeneous problem (aside from the centralised source).

The 2D problem (Fig. 3) has dimensions $10 \text{ cm} \times 10 \text{ cm}$. The central source region is a 2 cm side square and the cranked duct is 1 cm wide. The 3D problem (4) has dimensions $10 \text{ cm} \times 10 \text{ cm} \times 10 \text{ cm}$, with the source being a 2 cm side cube and the duct having a square cross section of side 1 cm (see Fig. 4).

Both 2D and 3D case were created using the GMSH mesh generation software (Geuzaine and Remacle, 2009) for a variety of element types and mesh refinements.

In addition to the cranked duct an alternative test case is presented which aims to provide an similarly challenging problem but this time in an unstructured mesh environment. Fig. 5 displays a radial cross-section of the problem. Just as with the cranked duct the problem is split into three separate material regions, a source region at the centre shown in black with a fixed neutron source of $1.0 \text{ cm}^{-3} \text{ s}^{-1}$ and a scatter cross-section of 1.0 cm^{-1} , a thick region shown in gray with a scatter cross-section of $r \text{ cm}^{-1}$ and a thin region in white with a scatter cross-section of $1/r \text{ cm}^{-1}$. The variable r is once again a measure of the heterogeneity of the problem. The spherical boundary is a vacuum and all other boundaries are

reflective in order to accurately represent a full sphere.

3.2. Low-level correction

The algorithms described in section 2.2 require that an approximation of the inverse of the low-level matrix is obtained in order to provide the coarse correction. This is achieved through a preconditioning step of an AMG preconditioner (Stuben, 2001). There are numerous AMG algorithms available, the methods presented here were run using BoomerAMG (Henson and Weiss, 2002; Lawrence Livermore National Laboratory, 0000), ML (Sala et al., 2004), AGMG (Notay, 2010, 2012, 2014; Napov and Notay, 2012), and GAMG which is available through the PETSC software package (Balay et al., 1997, 2014).

Some of these AMG algorithms have a large variety of input parameters. Here for the sake of simplicity default settings of each AMG method are always used and they are always called as a single preconditioning step and not a full solution to the low-level problem. In (O'Malley et al., 2017) a brief study into the impact of more thoroughly solving the low-level problem indicated that the improved convergence is unlikely to be worth the increased computational cost.

The AMG method which leads to the fastest solution will vary depending on the problem and preconditioning algorithm. For the sake of simplicity the results that follow will show only the times obtained with the AMG method which was found to be optimal for that case.

3.3. Alternative preconditioners

As well as the constant and continuous methods the performance of a third preconditioner is studied, one which uses P-multigrid to restrict to a linear discontinuous problem and then applies the AMG correction without a further restriction step. Such a method would rely more heavily on the performance of the AMG algorithm used. A block Jacobi smoother is again used. For problems with second-order elements this preconditioning algorithm will be set up as shown in equation (9) for $N = 2$. This method will be referred to as the "P-multigrid" preconditioner.

In addition AMG applied directly to the problem with no other coarsening methods is used as a benchmark. Of the AMG

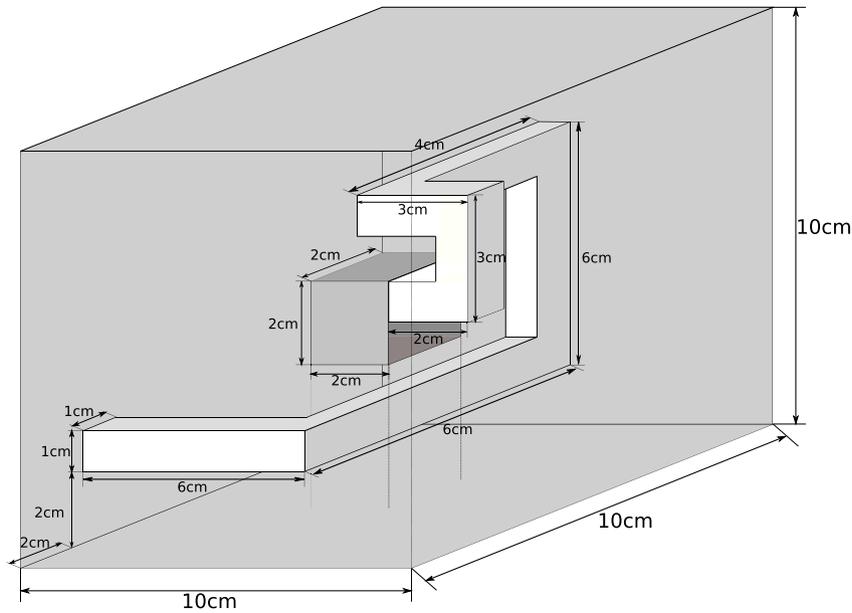


Fig. 4. Visualisation of the 3D cranked duct test problem.

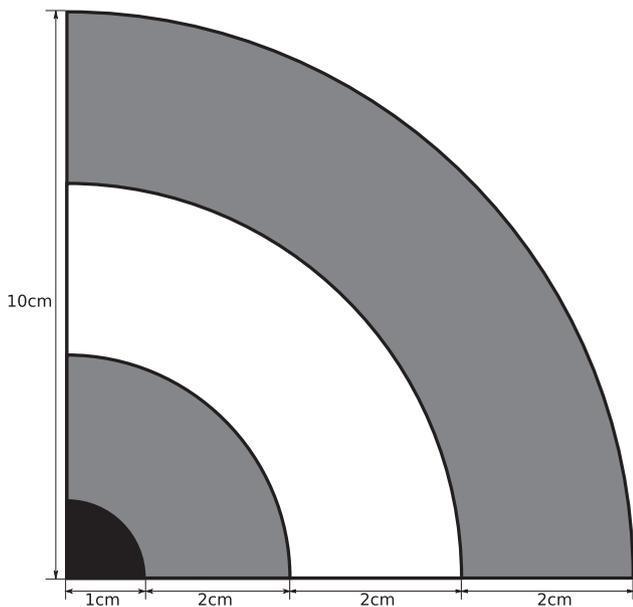


Fig. 5. Radial cross section of the 3D concentric sphere test problem.

Table 1

Iterations to convergence of two-level preconditioner for varying ω . BoomerAMG used for low-level correction.

ω	Iterations
1	28
0.9	27
0.8	27
0.7	28
0.6	30
0.5	32

preconditioners presented in section 3.2 the AGMG consistently outperformed the others. This is consistent with results in (Turcksin and Ragusa, 2014) and (O'Malley et al., 2017). Therefore all problems studied will use AGMG as the benchmark AMG preconditioner.

3.4. Optimising smoother damping

Varying the damping factor (ω) of the smoother in a multilevel preconditioner may impact how well it performs. In order to achieve a fair comparison of the preconditioners presented here it is therefore necessary to ensure that a close to optimal damping is used in all cases. In this section the preconditioners are tested with varying values of omega in order to gain some insight into the optimal value. The test problem used in this section is a

homogeneous ($r = 1.0$) case of the 3D cranked duct problem discretised with 1000 s-order structured hexahedral elements. For each preconditioner a value of ω must be specified for each level but the coarsest, so for example a two-level method has one independent value of ω . It is important that ω is constant for different smoothing stages on the same level as this is necessary to ensure symmetry of the preconditioner.

The first case is for the P-multigrid preconditioner, with the results displayed in Table 1. What is most noticeable from this table is that although the optimal value for ω is approximately 0.7–0.8 the iteration number is relatively insensitive to ω as long as it is fairly close to the optimal value. This is important because different material properties or finite element discretisations will lead to slight changes in the optimal value of ω and it is unlikely to be practical to calculate this in all cases. Therefore it is reasonable to set ω to a fixed value that should be close to the optimal value in all cases. In this paper $\omega = 0.8$ is used in all cases for the two-level preconditioner.

In the case of multi-level preconditioners the issue is somewhat more complicated due to the fact that smoothing occurs here on multiple levels, each of which may use an independent value for ω . As the model problem is discretised with second-order elements ($N = 3$) there will be two independent values of ω to be selected, one for smoothing on the second-order FEM problem (high-level ω) and another for smoothing on the first-order FEM problem (low-level ω).

Table 2 shows how the iterations to converge vary with both

Table 2
Iterations to convergence of multi-level preconditioners for varying of both high-level and low-level ω . BoomerAMG used for low-level correction.

(a) Continuous Low-Level							
		High-Level ω					
		1.0	0.9	0.8	0.7	0.6	0.5
Low-Level ω	1.0	52	27	26	26	28	29
	0.9	27	26	26	27	28	29
	0.8	27	26	26	27	28	30
	0.7	27	26	26	27	29	31
	0.6	27	26	26	28	29	31
	0.5	27	26	27	28	30	32
(b) Constant Low-Level							
		High-Level ω					
		1	0.9	0.8	0.7	0.6	0.5
Low-Level ω	1.0	50	49	48	48	48	48
	0.9	49	49	48	49	48	49
	0.8	50	49	48	49	49	50
	0.7	51	50	49	49	49	50
	0.6	52	50	49	50	50	51
	0.5	53	51	50	50	51	52

values of ω . Again it is worth noting that both preconditioners appear to be fairly insensitive to small variations in ω . This is particularly true for the ω on the low level smoother. The primary exception to this rule is for the continuous preconditioner when both values of ω are equal to 1.0, in which case performance is severely weakened.

For all results in this paper, the continuous preconditioner will use $\omega_{\text{highlevel}} = 0.9$ and $\omega_{\text{lowlevel}} = 0.7$. The constant preconditioner will use $\omega_{\text{highlevel}} = 0.6$ and $\omega_{\text{lowlevel}} = 0.9$. Across the various problems which are to be examined as well as variations on the preconditioners being used it may be that these values are not always those that yield the precisely optimal convergence. They will however be close to the optimal value and since it has been shown that small deviations from the ideal value of ω have a small impact on convergence it should not be a cause for great concern. Calculating optimal values for smoother damping for each individual problem would not be practical.

3.5. Performance of standard multi-level V-Cycles

The constant and continuous multi-level preconditioners are

now tested in comparison to the two benchmark preconditioners previously specified. The methods are first implemented using a standard V-cycle, as defined in equation (9) where $N = 3$. For each preconditioner the number of CG cycles required to reach convergence and the time in seconds taken to do so is recorded. For this case and all other cases unless otherwise stated the simulations are run on the same computer in serial.

Tables 3 and 4 show the results obtained for the 2D and 3D case of the cranked duct problem when discretised with structured elements. Of the four methods studied it is the continuous method that displays the strongest overall performance in terms of solution time, consistent with the results in (O'Malley et al., 2017). The constant method used in a V-cycle, though it provides stable convergence, is consistently the slowest of the four preconditioners.

The P-multigrid is competitive with the continuous method. It is marginally slower than the continuous preconditioner in most cases and in some 2D homogeneous cases is in fact faster. The AGMG method is slower than the continuous or P-multigrid methods in most cases and, when heterogeneity is increased in the 3D case, its convergence time is increased by a larger degree than either of them. In addition, the AGMG preconditioner was not able to find a solution for the largest 3D problem due to the memory requirements of the preconditioner set-up exceeding what was available on the computer being used. This suggests that AGMG has larger memory requirements than the other preconditioners, an issue that will be examined in section 3.8.

In order to demonstrate the impact of AMG choice Fig. 6 plots results for a single 2D problem with all AMG variants shown.

The next set of results in Table 5 are for the concentric sphere problem, which is discretised with unstructured tetrahedral elements. The preconditioners perform relative to each other in a similar manner as with the structured case. These cases further demonstrate that the AGMG preconditioner when used alone struggles with high heterogeneity problems. Once more the continuous preconditioner consistently displays superior performance to all others.

3.6. Multi-level W-Cycle

The W-cycle, as described in equation (10), is a variant of the multilevel method that does more work on the lower level grids for each preconditioning step. This naturally means that the computational cost of each preconditioning step will be higher, but it may

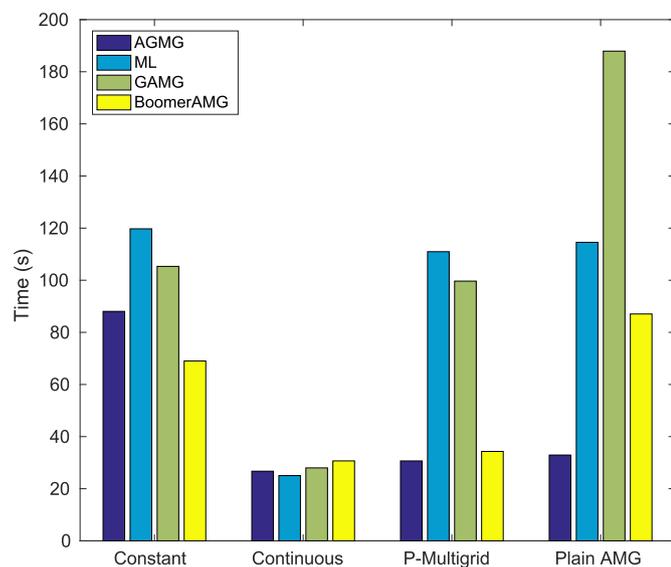
Table 3
Iterations and time taken to solve the MIP diffusion 2D cranked duct problem discretised with second-order structured quadrilaterals.

Heterogeneity Factor $r = 1.0$								
Elements	Constant + BoomerAMG		Continuous + ML		P-Multigrid + AGMG		AGMG	
	Iterations	Time(s)	Iterations	Time(s)	Iterations	Time(s)	Iterations	Time(s)
1600	42	0.240	17	0.096	17	0.088	27	0.124
6400	42	0.764	17	0.320	17	0.296	27	0.420
25,600	42	3.12	17	1.29	16	1.12	27	1.71
102,400	41	12.3	17	5.21	16	4.58	28	7.06
409,600	39	47.0	15	19.0	16	18.9	29	30.4
Heterogeneity Factor $r = 100.0$								
Elements	Constant + BoomerAMG		Continuous + ML		P-Multigrid + AGMG		AGMG	
	Iterations	Time(s)	Iterations	Time(s)	Iterations	Time(s)	Iterations	Time(s)
1600	54	0.300	21	0.124	27	0.148	29	0.140
6400	54	0.988	21	0.392	25	0.460	29	0.452
25,600	57	4.20	21	1.58	24	1.82	29	1.84
102,400	57	17.1	22	6.79	25	7.42	29	7.46
409,600	57	69.0	20	25.0	25	30.7	31	32.9

Table 4

Iterations and time taken to solve the MIP diffusion 3D cranked duct problem discretised with second-order structured hexahedra.

Heterogeneity Factor $r = 1.0$									
Elements	Constant + BoomerAMG		Continuous + AMG		P-Multigrid + AMG		AMG		Time(s)
	Iterations	Time(s)	Iterations	Time(s)	Iterations	Time(s)	Iterations		
1000	47	1.02	26	0.568	31	0.672	28	0.764	
8000	52	9.44	27	4.98	30	5.81	28	6.36	
64,000	51	77.7	26	39.6	29	45.1	28	44.4	
512,000	50	607.3	25	310.8	26	328.5	n/a	n/a	
Heterogeneity Factor $r = 100.0$									
Elements	Constant + BoomerAMG		Continuous + AMG		P-Multigrid + AMG		AMG		Time(s)
	Iterations	Time(s)	Iterations	Time(s)	Iterations	Time(s)	Iterations		
1000	94	2.00	54	1.16	59	1.31	47	1.24	
8000	76	13.8	35	6.51	40	7.95	36	8.29	
64,000	74	110.7	31	47.4	34	53.3	42	66.8	
512,000	74	912.2	30	373.4	33	422.5	n/a	n/a	

**Fig. 6.** Timing comparison with all AMG variants for the $r = 100.0$ case of the 2D cranked duct problem discretised with 409600 structured quadrilateral elements.

also lead to the total number of iterations required to achieve convergence being reduced. For the results from the V-cycles the constant preconditioner in particular required a large number of iterations.

The parameter J in equation (10) determines the precise shape of the W-cycle with J representing the number of times the cycle visits the coarsest level per preconditioning step. This paper will refer to a W-cycle with $J = 2$ as a 2W-cycle, with $J = 3$ as a 3W-cycle and so on. A W1-cycle would be identical to the V-cycle in equation (9).

The heterogeneous variation of the 3D cranked duct problem discretised with second-order structured hexahedral elements is used to test the impact of these W-cycles and provide a comparison to the V-cycle results.

Table 6 shows how increasingly large W-cycles impact the performance of the constant preconditioner. It is clear that the addition of a W-cycle can provide a significant improvement in convergence rate. Increasing the length of the W-cycle continues to further reduce iteration number until saturation is reached at 7W–8W. This naturally leads to significantly lower computational times with the time saved by reducing iteration number exceeding the additional cost of each preconditioning step. For this case the optimal W-cycle appears to be 5W–7W.

In Table 7 the W-cycle is applied to the continuous preconditioner. Here the impact on iteration number of the W-cycle is very small, with a 4W-cycle leading to at best 1–2 iterations fewer than for the V-cycle. Because of this the V-cycle has the fastest convergence time for all cases, providing strong evidence that W-cycles for the continuous preconditioner are not beneficial.

Table 8 takes the optimal cycle for both the constant and continuous preconditioner and compares them once again to the P-multigrid and AMG cases. The continuous preconditioner, which has not changed, remains the fastest. However, the constant preconditioner with a W-cycle is now, while still the slowest, much more competitive with the P-multigrid and AMG.

3.7. Eigenmode analysis

As well as the computational results above further insight into the performance of preconditioners may be obtained by examining the eigenvalues and respective eigenvectors of the preconditioned matrix. The eigenvectors correspond to the error modes in the system and their eigenvalues indicate how effectively iterative solvers will be able to reduce their magnitude.

Calculating the eigenvalues and eigenvectors of a system is computationally intensive, therefore this section will focus on problems with a small number of degrees of freedom. The results presented here are for a homogeneous 2D problem consisting of 100 s-order quadrilateral elements. As each element will have 9 degrees of freedom this will lead to 900 independent eigenvalues and eigenvectors.

Fig. 7 illustrates the distribution of eigenvalues for the P-multigrid preconditioner, the constant and continuous V-cycle preconditioners and the constant 5W-cycle preconditioners. Continuous W-cycle preconditioners are not examined due to the previous results indicating that the addition of the W-cycle has a minimal effect on the convergence when compared to the V-cycle.

Fig. 7 shows that the largest eigenvalues belong to the constant V-cycle preconditioner, this is consistent with the previous results where the constant V-cycle required more iterations to converge in comparison to the others. A small group of eigenvalues for the constant V-cycle at the left hand side are particularly problematic, as some of them get quite close to 1 which is the point at which a system's convergence can greatly suffer. The continuous preconditioner on the other hand has lower eigenvalues than the P-multigrid method in almost all cases, however its largest eigenvalue is quite close to the largest eigenvalue of the two-level method. This agrees with the computational results which showed that while the continuous preconditioner typically

Table 5
Iterations and time taken to solve the MIP diffusion 3D concentric sphere problem discretised with second-order unstructured tetrahedra.

Heterogeneity Factor $r = 1.0$								
Elements	Constant + AGMG		Continuous + AGMG		P-Multigrid + AGMG		AGMG	
	Iterations	Time(s)	Iterations	Time(s)	Iterations	Time(s)	Iterations	Time(s)
1118	99	0.408	70	0.280	71	0.292	78	0.316
7098	119	3.88	87	2.81	84	2.77	97	3.47
47,689	149	36.6	97	23.7	97	23.8	134	36.0
370,971	164	242.7	104	153.1	110	159.8	194	275.6
1,228,250	274	3594.4	165	2160.1	182	2343.7	230	2476.7
Heterogeneity Factor $r = 100.0$								
Elements	Constant + AGMG		Continuous + AGMG		P-Multigrid + AGMG		AGMG	
	Iterations	Time(s)	Iterations	Time(s)	Iterations	Time(s)	Iterations	Time(s)
1118	114	0.372	81	0.264	77	0.248	82	0.212
7098	131	4.28	93	3.00	93	3.12	87	2.78
47,689	158	38.7	102	25.0	102	25.1	125	31.5
370,971	185	358.2	115	174.1	132	197.3	167	234.3
1,228,250	312	4084.8	186	2446.7	219	2772.9	298	3683.5

Table 6
Effect of W-cycle on constant preconditioner. 3D-cranked duct problem discretised with structured second-order hexahedra, heterogeneity factor $r = 100.0$.

Constant + AGMG								
Iterations								
Elements	V-Cycle	2W-Cycle	3W-Cycle	4W-Cycle	5W-Cycle	6W-Cycle	7W-Cycle	8W-Cycle
1000	96	76	67	62	60	60	61	60
8000	81	62	53	48	45	43	42	41
64,000	80	59	50	45	41	39	37	37
512,000	82	60	49	44	41	38	37	36
Time(s)								
Elements	V-Cycle	2W-Cycle	3W-Cycle	4W-Cycle	5W-Cycle	6W-Cycle	7W-Cycle	8W-Cycle
1000	2.05	1.72	1.60	1.56	1.58	1.66	1.78	1.84
8000	14.9	12.1	10.8	10.3	10.3	10.2	10.6	10.6
64,000	119.1	93.1	83.4	79.2	75.8	75.5	75.1	79.6
512,000	992.1	769.8	664.3	629.0	615.5	598.1	609.0	618.5

Table 7
Effect of W-cycle on continuous preconditioner. 3D-cranked duct problem discretised with structured second-order hexahedra, heterogeneity factor $r = 100.0$.

Continuous + AGMG				
Iterations				
Elements	V-Cycle	2W-Cycle	3W-Cycle	4W-Cycle
1000	54	54	53	52
8000	35	36	35	35
64,000	31	31	30	30
512,000	30	30	29	29
Time(s)				
Elements	V-Cycle	2W-Cycle	3W-Cycle	4W-Cycle
1000	1.16	1.25	1.30	1.35
8000	6.51	7.20	7.90	8.17
64,000	47.4	52.0	56.3	59.5
512,000	373.4	405.6	419.8	449.4

Table 8
Time to solve MIP diffusion 3D cranked duct problem discretised with second-order structured hexahedra, heterogeneity factor $r = 100.0$. Using best case cycle for constant and continuous preconditioner.

Elements	Constant + AGMG 6W-Cycle	Continuous + AGMG V-Cycle	P-Multigrid + AGMG	AGMG
1000	1.66	1.16	1.31	1.24
8000	10.2	6.51	7.95	8.29
64,000	75.5	47.4	53.3	66.8
512,000	598.1	373.4	422.5	n/a

converges with less iterations than the two-level the difference is fairly small.

When the 5W-cycle is applied to the constant preconditioner some of the largest eigenvalues are substantially reduced, which again agrees with the computational results. Note that the general shape of the eigenvalue plot for the constant W-cycle is closer to that of the continuous and two-level preconditioners than when it was run with a V-cycle, particularly for the largest eigenvalues. This indicates that there were perhaps several eigenmodes particularly problematic for the constant V-cycle and not the continuous and two-level preconditioners that the implementation of the W-cycle has helped to suppress.

3.8. Memory usage

So far the metric by which all the preconditioners presented have been judged has been simply speed of convergence. However,

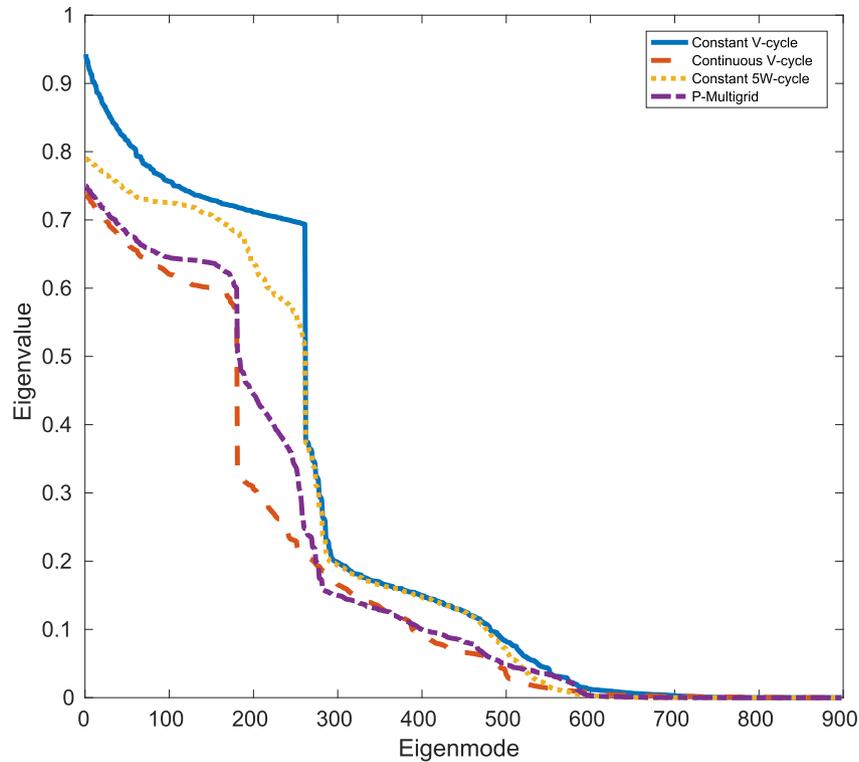


Fig. 7. Preconditioner eigenvalue distribution.

in many large supercomputer calculations an equally important consideration can be the memory requirement of a method. Multilevel preconditioners necessitate extra memory in order to store information about the low-level systems. Additionally the methods present here calculate and store the inverted blocks for the block Jacobi smoother in the setup phase in order to reduce run-time, which further increases preconditioner memory requirements.

The 3D cranked duct problem and concentric sphere problem were run again, and this time the virtual memory usage was recorded. The memory requirement for each preconditioner is obtained by recording the total memory used when run with that preconditioner and subtracting the memory used when running with no preconditioning. The results are displayed in Tables 9 and 10.

The results show that the constant preconditioning method is the most memory efficient preconditioner presented here. The continuous method uses slightly more than the constant for the

hexahedral element case and roughly the same for the tetrahedral problem. The two-level preconditioner, although competitive with the constant preconditioner in timings, has consistently higher memory requirements.

The AGMG method has significantly higher memory requirements than all other methods. For the largest hexahedral problem the memory requirement was more than was available on the computer being used so the problem could not be completed. An estimate for this case is provided, based on memory usage at the time the program reached the memory cap.

4. Conclusions

This paper applied the P-multigrid principle in order to expand two hybrid multilevel techniques developed for linear DG-FEM MIP diffusion problems, the “constant” and the “continuous” preconditioners, to higher order elements. Although the results here focused exclusively on second-order elements the methods expand

Table 9
Memory required to store preconditioner for the 3D cranked duct problem, with structured hexahedral elements.

Memory Usage of Preconditioners (Gb)				
Elements	Constant + AGMG	Continuous + AGMG	P-Multigrid + AGMG	AGMG
8000	0.14	0.18	0.33	0.82
64,000	0.8	1.1	1.3	6.4
512,000	7.1	8.4	10.7	n/a (estimate:50–60)

Table 10
Memory required to store preconditioner for the 3D concentric sphere problem, with unstructured tetrahedral elements.

Memory Usage of Preconditioners (Gb)				
Elements	Constant + AGMG	Continuous + AGMG	P-Multigrid + AGMG	AGMG
370,971	1.1	1.0	1.5	3.4
1,228,250	7.8	7.9	11.5	27.3

naturally to higher orders. In addition the performance of P-multigrid without a constant or continuous correction was examined. These preconditioners used a correction from an AMG algorithm at the coarse level to form a hybrid multilevel scheme. These preconditioned diffusion schemes may then be applied as DSA for neutron transport solvers in order to solve reactor physics problems.

As a benchmark AGMG, a strong AMG algorithm, was used to precondition the problem directly. For the constant, continuous and P-multigrid methods a variety of AMG methods were used to generate the low-level correction and results are displayed for whichever was found to be optimal for a particular case.

An initial comparison of the methods, with a V-cycle being used for the multilevel schemes, found that the continuous preconditioner provided the fastest convergence on almost all problems. The P-multigrid method was next fastest, followed by AGMG and finally the constant method. The AGMG showed a noticeably greater worsening of its performance when heterogeneity in a problem was increased in comparison to the other methods, particularly for 3D cases.

The constant and continuous method were then adapted to work with W-cycles of various shapes. It was found that, while the continuous method displayed weaker performance when run in a W-cycle, the constant method was significantly improved. When used in a W-cycle the constant method displayed convergence times which were very close to that of the P-multigrid and, in some cases, faster. The continuous method with a V-cycle remained the fastest method however.

As an alternative to the speed of convergence another metric was examined, the memory requirements of each preconditioner. In this study it was the constant preconditioner which was found to have the lowest memory requirements, closely followed by the continuous method. The P-multigrid required more memory than either constant or continuous and AGMG's usage was significantly higher than the others.

While the continuous preconditioner is fastest, all preconditioners shown are effective for reducing problem convergence times. It is in terms of memory usage where the hybrid multilevel methods, particularly the constant and continuous, significantly outperform AMG. With DSA neutron transport codes frequently requiring preconditioners to be created and stored for a large number of energy levels the benefit of such memory savings could be very significant.

Further work could examine further the cycles used in the multilevel formulation of the constant and continuous methods in order to further optimise them, going beyond the relatively simple V-cycle and W-cycles presented here. In addition the impact using different smoothers, or methods other than AMG to calculate the low-level correction could be examined. Finally a variation on the continuous method whereby the high-order discontinuous FEM is restricted to a high-order continuous FEM may be a valuable area of study.

Data statement

In accordance with EPSRC funding requirements all supporting data used to create figures and tables in this paper may be accessed at the following DOI: <https://doi.org/10.5281/zenodo.376518>.

Acknowledgements

B.O'Malley would like to acknowledge the support of EPSRC under their industrial doctorate programme (EPSRC grant number: EP/G037426/1), Rolls-Royce for industrial support and the Imperial College London (ICL) High Performance Computing (HPC) Service

for technical support. M.D. Eaton and J. Kópházi would like to thank EPSRC for their support through the following grants: Adaptive Hierarchical Radiation Transport Methods to Meet Future Challenges in Reactor Physics (EPSRC grant number: EP/J002011/1) and RADIANT: A Parallel, Scalable, High Performance Radiation Transport Modelling and Simulation Framework for Reactor Physics, Nuclear Criticality Safety Assessment and Radiation Shielding Analyses (EPSRC grant number: EP/K503733/1).

References

- Arnold, D.N., Brezzi, F., Cockburn, B., Marini, L.D., 2002. Unified analysis of discontinuous galerkin methods for elliptic problems. *J. Numer. Analysis* 39, 1749–1779.
- Balay, Satish, Gropp, William D., McInnes, Lois Curfman, Smith, Barry F., 1997. Efficient management of parallelism in object oriented numerical software libraries. In: Arge, E., Bruaset, A.M., Langtangen, H.P. (Eds.), *Modern Software Tools in Scientific Computing*. Birkhäuser Press, pp. 163–202.
- Balay, Satish, Abhyankar, Shrirang, Adams, Mark F., Brown, Jed, Brune, Peter, Buschelman, Kris, Eijkhout, Victor, Gropp, William D., Kaushik, Dinesh, Knepley, Matthew G., McInnes, Lois Curfman, Rupp, Karl, Smith, Barry F., Zhang, Hong, 2014. *PETSc users manual*. Technical Report ANL-95/11-Revision 3.5. Argonne Natl. Lab.
- Bastian, P., Blatt, M., Scheichl, R., 2012. Algebraic multigrid for discontinuous galerkin discretizations of heterogeneous elliptic problems. *Numer. Linear Algebra Appl.* 19, 367–388.
- Di Pietro, D.A., Ern, A., 2012. *Mathematical Aspects of Discontinuous Galerkin Methods* (chapter 4). Springer, pp. 117–184.
- Dobrev, V.A., 2007. Preconditioning of Symmetric Interior Penalty Discontinuous Galerkin FEM for Elliptic Problems. PhD thesis. Texas A&M University.
- Gesh, C.J., 1999. Finite Element Methods for Second Order Forms of the Transport Equation. PhD thesis. Texas A&M University.
- Geuzaine, C., Remacle, J.F., 2009. GMSH: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *Int. J. Numer. Methods Eng.* 0 (1–24).
- Henson, V.E., Weiss, R., 2002. BoomerAMG: a parallel algebraic multigrid solver and preconditioner. *Appl. Numer. Math.* 41, 155–177.
- Larsen, E.W., 1984. Diffusion-synthetic acceleration methods for discrete-ordinates problems. *Transp. Theory Stat. Phys.* 13, 107–126.
- Lawrence Livermore National Laboratory. HYPRE: High Performance Preconditioners. Lawrence Livermore National Laboratory. <http://www.llnl.gov/CASC/hypre/>.
- Mitchell, W.F., 2015. How High a Degree Is High Enough for High Order Finite Elements? Technical report National Institute of Standards and Technology.
- Napov, A., Notay, Y., 2012. An algebraic multigrid method with guaranteed convergence rate. *J. Sci. Comput.* 34 (1079–1109).
- Notay, Y., 2010. An aggregation-based algebraic multigrid method. *Electron. Trans. Numer. Analysis* 37 (123–146).
- Notay, Y., 2012. Aggregation-based algebraic multigrid for convection-diffusion equations. *J. Sci. Comput.* 34, 2288–2316.
- Notay, Y., 2014. User's Guide to AGMG. Technical report. Université Libre de Bruxelles.
- O'Malley, B., Kópházi, J., Smedley-Stevenson, R.P., Eaton, M.D., 2017. Hybrid multilevel solver for discontinuous galerkin finite element discrete ordinate (DG-FEM- S_N) diffusion synthetic acceleration (DSA) of radiation transport algorithms. *Ann. Nucl. Energy* 102, 134–147.
- Rønquist, E.M., Patera, A.T., 1987. Spectral element multigrid I. Formulation and numerical results. *J. Sci. Comput.* 2, 389–406.
- Sala, M., Hu, J.J., Tuminaro, R.S., 2004. ML3.1 Smoothed Aggregation User's Guide. Technical Report SAND2004–4821. Sandia National Laboratories.
- Schröder, A., 2011. Spectral and High Order Methods for Partial Differential Equations, Chapter Constrained Approximation in Hp-FEM: Unsymmetric Subdivisions and Multi-level Hanging Nodes. Springer, pp. 317–325.
- Siefert, C., Tuminaro, R., Gerstenberger, A., Scovazzi, G., Collis, S.S., 2014. Algebraic multigrid techniques for discontinuous Galerkin methods with varying polynomial order. *Comput. Geosci.* 18, 597–612.
- Stuben, K., 2001. A review of algebraic multigrid. *J. Comput. Appl. Math.* 128, 281–309.
- Stuben, K., Oswald, P., Brandt, A., 2001. *Multigrid*. Elsevier.
- Turcksin, B., Ragusa, J.C., 2014. Discontinuous diffusion synthetic acceleration for S_N transport on 2D arbitrary polygonal meshes. *J. Comput. Phys.* 274, 356–369.
- Van Slingerland, P., Vuik, C., 2012. Scalable Two-level Preconditioning and Deflation Based on a Piecewise Constant Subspace for (SIP)DG Systems. Technical report. Delft University of Technology.
- Van Slingerland, P., Vuik, C., 2015. Scalable two-level preconditioning and deflation based on a piecewise constant subspace for (SIP)DG systems for diffusion problems. *J. Comput. Appl. Math.* 275, 61–78.
- Wang, Y., Ragusa, J.C., 2010. Diffusion synthetic acceleration for high-order discontinuous finite element S_N transport schemes and application to locally refined unstructured meshes. *Nucl. Sci. Eng.* 166, 145–166.
- Briggs, William L., Henson, V.E., McCormick, S.F., 2000. *A Multigrid Tutorial*. SIAM.