SAND2017-12423J

On performance of Krylov smoothing for fully-coupled AMG preconditioners for VMS resistive MHD☆

Paul T. Lin[a,*], John N. Shadid[a,b], Paul H. Tsuji[c]

[a]*Sandia National Laboratories, Albuquerque, NM, 87185*
[b]*Department of Mathematics and Statistics, University of New Mexico, Albuquerque, NM, 87131*
[c]*Lawrence Livermore National Laboratory, Livermore, CA, 94550*

Abstract

This study explores the performance and scaling of a GMRES Krylov method employed as a smoother for an algebraic multigrid (AMG) preconditioned Newton-Krylov solution approach applied to a fully-implicit variational multiscale (VMS) finite element (FE) resistive magnetohydrodynamics (MHD) formulation. In this context a Newton iteration is used for the nonlinear system and a Krylov (GMRES) method is employed for the linear subsystems. The efficiency of this approach is critically dependent on the scalability and performance of the AMG preconditioner for the linear solutions and the performance of the smoothers play a critical role. Krylov smoothers are considered in an attempt to reduce the time and memory requirements of existing robust smoothers based on additive Schwarz domain decomposition (DD) with incomplete LU factorization solves on each subdomain. Three time dependent resistive MHD test cases are considered to evaluate the method. The results demonstrate that the GMRES smoother can be faster due to a decrease in the preconditioner setup time and a reduction in outer GMRESR solver iterations, and requires less memory (typically 35% less memory for global GMRES smoother) than the DD ILU smoother.

1. Introduction

The resistive magnetohydrodynamics (MHD) model describes the dynamics of charged fluids in the presence of electromagnetic fields and is used as a base-level continuum plasma model. Resistive MHD is used to model aspects of fundamental plasma physics phenomena (reconnection and hydromagnetic instabilities), technology applications (e.g. tokamaks, and plasma reactors), and astrophysical phenomena (e.q. solar physics and planetary dynamos) [1, 2]. The

☆This work was supported by the DOE NNSA ASC Algorithms effort and the DOE Office of Science AMR program at Sandia National Laboratories under contract DE-NA-0003525.

*Corresponding author
Email address: ptlin@sandia.gov (Paul T. Lin)

MHD system is strongly coupled, highly nonlinear and characterized by coupled physical phenomena that induce a very large range of time-scales in the response of the system. These characteristics make the scalable, robust, accurate, and efficient computational solution of these systems extremely challenging. In this context fully-implicit formulations, coupled with effective robust iterative solution methods, become attractive, as they have the potential to provide stable, higher-order time-integration of these complex multiphysics systems when long dynamical time-scales are of interest (see e.g. [3, 4, 5, 6, 7]). Currently leading-edge implicit resistive MHD simulations can demand high-resolution spatial discretization of 2D and 3D geometries with both structured or unstructured meshes that generate on the order of 10^6 to 10^{10} coupled unknowns. For this reason the implicit transient simulation of these systems requires efficient and scalable parallel iterative solution methods.

Krylov iterative linear solver algorithms are among the fastest and most robust iterative solvers for a wide variety of applications [8, 9]. The key factor influencing the robustness and efficiency of these solution methods is the choice of preconditioner. Among current preconditioning techniques multilevel type methods (e.g. two-level domain decomposition, multigrid and algebraic multigrid (AMG)) have been demonstrated to provide scalable solutions to a wide range of challenging linear systems. This includes elliptic scalar problems, linear elasticity, and computational fluid dynamics as just a few examples [10, 11, 12]. Multigrid scalability and performance is critically dependent on both the projection and the smoothers. This study focuses on the latter, specifically the performance of smoothers based on a Krylov type iterative method (GMRES) applied to the fully-coupled Newton-Krylov algebraic multigrid preconditioned solution approach described in [7, 13]. In this context the solution of the discrete system developed by a fully-implicit backward differentiation (BDF) type formulation of a variational multiscale (VMS) finite element spatial discretization of the resistive MHD system is considered [7].

In the context of fully-coupled direct-to-steady-state solution methods for VMS CFD and MHD large linear non-symmetric systems, experience has indicated that robust, and therefore more expensive, smoothing methods are required [14, 15, 16, 5, 17, 7]. For these type systems additive Schwarz domain decomposition (DD) with local incomplete LU factorizations have been shown to be an effective smoother [16, 18, 17, 7]. For these methods robustness can be enhanced by variable overlap between sub-domains and by allowing fill-in for the ILU factors [19, 20, 16, 18, 17]. This robustness however comes at a price since these methods are expensive and require large setup time and larger memory requirements to compute the ILU factors. In an extension to this work we consider the case of transient resistive MHD problems and carry out an initial assessment of AMG preconditioned Krylov methods based on a few well known standard stationary iterative methods (e.g. Jacobi, Gauss-Seidel, etc.) and the recursive application of a Krylov method used as a smoother.

There has been a fair amount of previous work employing Krylov smoothers with multigrid for SPD problems, e.g. scalar elliptic problems and elasticity problems [21, 22, 23, 24, 25]. There has been considerably less previous work

employing Krylov smoothers for Helmholtz problems [26] and for nonsymmetric systems; for example [27] has considered Krylov smooothers for the convection-diffusion equation. Additionally and independently we have considered Krylov smoothers for the MHD equations [28]. This present study is an extension of [28].

This paper is organized as follows. The resistive MHD equations will be presented, followed by the finite element discretization approach. The fully-coupled Newton-Krylov preconditioned with algebraic multigrid solution approach is then presented. Comparison of smoothers, including the Krylov smoothers, will be presented for three transient MHD test cases. Finally, we include some concluding remarks and discussion of future work.

2. Resistive MHD model equations and discretization

The governing equations considered in this study are the 3D resistive isothermal MHD equations including dissipative terms for the momentum and magnetic induction equations [1]. This model provides a continuum description of charged fluids in the presence of electromagnetic fields. The system of equations in residual form is:

$$\mathbf{R}_m = \frac{\partial(\rho\mathbf{u})}{\partial t} + \nabla\cdot[\rho\mathbf{u}\otimes\mathbf{u} - \frac{1}{\mu_0}\mathbf{B}\otimes\mathbf{B} + (P + \frac{1}{2\mu_0}\|\mathbf{B}\|^2)\mathbf{I} - \mu[\nabla\mathbf{u} + \nabla\mathbf{u}^T]] = \mathbf{0} \quad (1)$$

$$R_P = \frac{\partial\rho}{\partial t} + \nabla\cdot[\rho\mathbf{u}] = 0 \quad (2)$$

$$\mathbf{R}_I = \frac{\partial\mathbf{B}}{\partial t} + \nabla\cdot\left[\mathbf{u}\otimes\mathbf{B} - \mathbf{B}\otimes\mathbf{u} - \frac{\eta}{\mu_0}\left(\nabla\mathbf{B} - (\nabla\mathbf{B})^T\right) + \psi\mathbf{I}\right] = \mathbf{0} \quad (3)$$

Here $\mathbf{u}$ is the plasma velocity; ρ is the ion mass density; P is the plasma pressure; $\mathbf{B}$ is the magnetic induction (also termed the magnetic field) that is subject to the divergence-free involution $\nabla\cdot\mathbf{B} = 0$. In this formulation the Lagrange multiplier, ψ is introduced to allow numerical enforcement of the divergence involution as a constraint, $R_\psi = \nabla\cdot\mathbf{B} = 0$ [29, 7] The associated plasma current, $\mathbf{J}$, is obtained from Ampère's law as $\mathbf{J} = \frac{1}{\mu_0}\nabla\times\mathbf{B}$. The physical parameters in this model are the plasma viscosity, μ, the resistivity, η, and the magnetic permeability of free space, μ_0.

This study focuses on the incompressible limit of this system with ($\nabla\cdot\mathbf{u} = 0$). This limit is characteristic of low-flow-Mach number applications for compressible systems as well, and is the most challenging algorithmically because of the presence of the elliptic incompressibility constraint. The methods presented in this work, i.e. the VMS finite element (FE) formulation, Newton-Krylov nonlinear iterative solvers, and the fully-coupled algebraic multilevel preconditioners, also work in the variable density low-Mach-number compressible case.

For the spatial discretization the VMS FE technique is employed, the semi-discretized system is then integrated in time with a methods of lines approach based on BDF methods [7]. Here only a summary of the residual form of the

VMS/stabilized FE formulation is presented. In the weak form residual equations $[\mathbf{w}^h, q^h, \mathbf{C}^h, s^h]$ are the FE weighting functions for velocity, pressure, magnetic field respectively [7]. The sum $\sum_e$ indicates the integrals are taken only over element interiors Ω_e and integration by parts is not performed. ψ denotes the Lagrange multiplier and $\hat{\tau}_m$, $\hat{\tau}_P$, $\hat{\tau}_I$ and $\hat{\tau}_\psi$ the stabilization parameters (stabilization parameters are provided in [7]).

$$\mathbf{F}^h_{\mathbf{u}} = \int_\Omega \mathbf{w}^h \cdot \mathbf{R}^h_m d\Omega + \sum_e \int_{\Omega_e} \rho \hat{\tau}_m \mathbf{R}^h_m \otimes \mathbf{u}^h : \nabla \mathbf{w}^h d\Omega + \sum_e \int_{\Omega_e} \hat{\tau}_P (\nabla \cdot \mathbf{w}^h) R^h_P \, d\Omega$$

$$F^h_P = \int_\Omega q^h R^h_P d\Omega + \sum_e \int_{\Omega_e} \rho \hat{\tau}_m \nabla q^h \cdot \mathbf{R}^h_m d\Omega$$

$$\mathbf{F}^h_I = \int_\Omega \mathbf{C}^h \cdot \mathbf{R}^h_I d\Omega - \sum_e \int_{\Omega_e} \hat{\tau}_I (\mathbf{R}^h_I \otimes \mathbf{u}^h - \mathbf{u}^h \otimes \mathbf{R}^h_I) : \nabla \mathbf{C}^h \, d\Omega + \sum_e \int_{\Omega_e} \hat{\tau}_\psi (\nabla \cdot \mathbf{C}^h) R^h_\psi d\Omega$$

$$F^h_\psi = \int_\Omega s^h R^h_\psi d\Omega + \sum_e \int_{\Omega_e} \hat{\tau}_I \nabla s^h \cdot \mathbf{R}^h_I d\Omega$$

A finite element (FE) discretization of the stabilized equations gives rise to a system of coupled, nonlinear, non-symmetric algebraic equations, the numerical solution of which can be very challenging. These equations are linearized using an inexact form of Newton's method. A formal block matrix representation of these discrete linearized equations is given by

$$\begin{bmatrix} \mathbf{J_u} & \mathbf{B^T} & \mathbf{Z} & \mathbf{0} \\ \mathbf{D} & \mathbf{L_P} & \mathbf{0} & \mathbf{0} \\ \mathbf{Y} & \mathbf{0} & \mathbf{J}_I & \mathbf{B^T} \\ \mathbf{0} & \mathbf{0} & \mathbf{B} & \mathbf{L}_\psi \end{bmatrix} \begin{bmatrix} \delta \hat{\mathbf{u}} \\ \delta \hat{\mathbf{P}} \\ \delta \hat{\mathbf{B}} \\ \delta \hat{\psi} \end{bmatrix} = - \begin{bmatrix} \mathbf{F_u} \\ \mathbf{F_P} \\ \mathbf{F_I} \\ \mathbf{F}_\psi \end{bmatrix}. \tag{4}$$

where the block diagonal contribution of the stabilization procedure has been highlighted by a specific ordering. The block matrix $\mathbf{J_u}$ corresponds to the discrete transient, convection, diffusion and stress terms acting on the unknowns $\delta\hat{\mathbf{u}}$; the matrix $\mathbf{B}$ corresponds to the discrete gradient operator; $\mathbf{D}$, the discrete representation of the continuity equation terms with velocity (note for a true incompressible flow this would be the divergence operator); the block matrix $\mathbf{J}_I$ corresponds to the discrete transient, convection, and diffusion acting on magnetic induction; and the matrices $\mathbf{L_P}$ and $\mathbf{L}_\psi$ correspond to the discrete "pressure Laplacian" and "Lagrange multiplier Laplacian" [7]. The right hand side vectors contain the residuals for Newton's method. The existence of the nonzero matrices, $\mathbf{L_P}$ and $\mathbf{L}_\psi$, in the stabilized FE discretization is in contrast to Galerkin methods using mixed interpolation that produce a zero block on the total mass continuity and solenoidal constraint diagonal. The existence of these block matrices helps to enable the solution of the linear systems with a number of algebraic and domain decomposition type preconditioners that rely on non-pivoting ILU type factorization, or standard stationary iterative methods such

as Jacobi or a DD Gauss-Seidel sub-domain solver [14, 15]. Although the above formal block matrix representation provides insight into the system, the actual linear algebra implementation in the application employs an ordering by FE mesh node with each degree of freedom ordered consecutively.

3. Fully-coupled Newton-Krylov multigrid preconditioned solution approach

Numerical discretization of the governing equations produces a large sparse, strongly-coupled nonlinear system. Although fully-coupled Newton-Krylov techniques [30], where a Krylov solver is used to solve the linear system generated by a Newton's method, are robust, efficient solution of the large sparse linear system that must be solved for each nonlinear iteration is challenging [9, 18]. The performance, efficiency and scalability of the preconditioner is critical [9, 31]. It is well known in the literature that Schwarz domain decomposition preconditioners do not scale due to lack of global coupling [32, 19, 33]. Multigrid methods are one of the most efficient techniques for solving large linear systems [11, 34, 35]. As we have described our Newton-Krylov preconditioned by algebraic multigrid solution method in detail in our previous work [18, 36, 7], we will provide only a very brief description here and refer the reader to our previous work for further details. Although we employ the ML [37] smoothed aggregation [38] library, we employ a nonsmoothed aggregation approach (due to the matrices being nonsymmetric) with uncoupled aggregation. For system of partial differential equations (PDEs), aggregation is performed on the graph where all the PDEs per mesh node is represented by a single vertex. The discrete equations are projected to the coarser level employing a Galerkin fashion with a triple matrix product, $A_{\ell+1} = R_\ell A_\ell P_\ell$, where R_ℓ restricts the residual from level ℓ to level ℓ+1, A_ℓ is the discretization matrix on level ℓ and P_ℓ prolongates the correction from level ℓ+1 to ℓ. We typically employ both pre- and post-smoothing on each level of the multigrid V-cycle (we do not employ W-cycles at scale because it is well known that W-cycles do not scale, e.g. [39]). During the aggregation process, at coarser levels there are fewer and fewer aggregates per MPI process. This will lead to poor quality aggregates as well as the minimum size of the coarsest level being limited by the number of MPI processes. One approach to remedy these problems is to perform a repartitioning of the coarser level matrices by moving them to a subset of the MPI processes, and this approach is employed by the ML library [37, 36]. This allows the construction of better quality aggregates, reduces the communication at coarser levels, and does not constrain the minimum number of aggregates on the coarsest level to be the number of MPI processes. Although at the coarsest levels many MPI processes will be idle, the advantages of this approach significantly outweighs the disadvantages.

The Trilinos framework [40] provides the preconditioned Newton-Krylov method and preconditioners used for this work. Krylov methods are provided by the Aztec [41] library and the multigrid cycles and grid transfers are provided by ML. The Zoltan [42] parallel partitioning, load balancing and data-management

services library provides the recursive coordinate bisection (RCB) partitioning algorithm, an algorithm that partitions a graph based on the coordinates of the vertices. ML provides the data movement when repartitioning the matrix on coarser levels. The Ifpack [43] and Amesos libraries provides the smoothers and coarsest level solve. A serial sparse direct solver is employed on the coarsest level of multigrid. Although Trilinos was employed for this work, other complementary solver packages are available, notably PETSc [44], and other complementary algebraic multigrid packages are available (e.g. HYPRE [45]).

As mentioned in the introduction, we were interested in evaluating Krylov smoothers compared with our standard ILU smoother for our MHD test cases. For our Newton-Krylov solution approach, a GMRES solver is employed that is preconditioned by multigrid with ILU smoother. Because our test cases matrices are nonsymmetric, the choice of GMRES [46] for the Krylov smoother for our initial evaluations is appropriate. When the Krylov/GMRES smoother is employed, there are two levels of Krylov methods and possibly two levels of preconditioners. Because the preconditioner is changing due to the GMRES smoother, it is necessary to employ a GMRES approach such as flexible GMRES (FGMRES) [47] or GMRESR [48] for the "outer" GMRES Krylov method. This "outer" GMRESR/FGMRES Krylov method is then preconditioned by multigrid. Each level of the multigrid V-cycle employs a Krylov based smoother which in our case is based on GMRES. This is denoted as the "inner" GMRES. This "inner" GMRES can be preconditioned, e.g. by a standard relaxation approach such as point or block Jacobi or Gauss-Seidel, but often this "inner" GMRES will not be preconditioned. For the transient MHD test cases investigated in this study the "outer" GMRES solver is Aztec GMRESR, preconditioned by ML AMG, with "inner" Aztec GMRES smoother on the different multigrid levels, with Ifpack block Jacobi preconditioner (on the "inner" Aztec GMRES). It should be pointed out that the conventional wisdom heading towards exascale era is that algorithms should reduce communication. Unfortunately the Krylov smoother actually increases global communication (e.g. dot products are global operations). In order to avoid this increase in global communication, we have also explored additive Schwarz Krylov smoothers or domain decomposition Krylov smoothers where the GMRES matrix-vector multiplications and dot products are limited to the subdomains. As with typical additive Schwarz smoothers, these subdomains may or may not have overlap. In this study, "Krylov smoother," "GMRES smoother," or "global GMRES smoother" will refer to a global smoother, while additive Schwarz Krylov smoothers or domain decomposition Krylov smoothers will be qualified by adjectives such as "additive Schwarz," "domain decomposition" or "local Krylov" smoother.

An estimate of the fine grid memory usage of an "outer" GMRES using an "inner" GMRES smoother (given in the number of double floating pointer entries) is provided relative to an "outer" GMRES using an ILU smoother algorithm. Here the number of nonzero entries in the Jacobian matrix corresponding to a 3D FE discretization with, dof, degrees of freedom at each of the, N_{nodes}, nodes of a rectangular domain is estimated as $nnz = 27 \times dofs \times dofs \times N_{nodes}$. The estimate of the memory usage for the "outer" and "inner" Krylov subspace

size is, $K_o \times dofs \times N_{nodes}$, and $K_i \times dofs \times N_{nodes}$. This estimate assumes both methods require roughly the same "outer" GMRES Krylov subspace size. Using these estimates the ratio of memory usage of the "inner" GMRES preconditioned method to an additive Schwarz domain decomposition ILU(0) approximate factorization preconditioned method (with no overlap) becomes,

$$r = \frac{(27 \times dofs + K_o + K_i) \times dofs \times N_{nodes}}{(2 \times 27 \times dofs + K_o) \times dofs \times N_{nodes}}.$$

For a sufficiently large number of $dofs$ and a fast converging outer iteration such that, $K_o << 27 \times dofs$, and an "inner" Krylov solve for which $K_i/K_o << 1$, the estimate is approximately $r = \frac{1}{2}$.

4. Results and discussion

Results for three transient resistive MHD test cases are presented. These included (1) the decay of an initialized hydromagnetic vortex that leads to a turbulent MHD flow, (2) a transient magnetic flux tube / island coalescence problem, and (3) a transient MHD hydromagnetic Kelvin-Helmholtz stability problem, all of which are of recent scientific interest. Using these test problems initial results are presented for the scalability of the Krylov smoother in comparison with more traditional stationary iterative methods used as AMG smoothers. This comparison is carried out for an MPI-only implementation of the solution methods.

4.1. Taylor-Green MHD turbulent vortex decay

The first transient MHD test case is a Taylor-Green vortex generalized to MHD as described in [49, 50]. We employ the same domain and initial conditions and for the discretization employ the full VMS formulation as described in [51]. The domain is a periodic box of size $[-\pi, \pi]^3$ with Reynolds number $Re = 1800$ and magnetic Reynolds number $Re_m = 1800$. Tables 1–3 present a weak scaling study for three different mesh sizes: 128^3, 256^3 and 512^3 elements cubed, or 16.8M, 134M and 1.1B DOFs, respectively. These problems are run on 256, 2048 and 16384 cores respectively (one MPI process per core) of a linux cluster that consists of dual-socket Intel Xeon 2.6 GHz oct-core Sandy Bridge processors with QDR InfiniBand fat-tree interconnect (Tri-lab computing cluster "TLCC2" machine at SNL). 20 time steps were used for all the simulations, and weak scaling was performed with fixed CFL (CFL ≈ 0.5). A BDF3 time integration approach was employed. We refer to the linear system solve time or preconditioned iteration time (i.e. not including preconditioner setup) as the "solve" time. The "linear solve time" is the sum of this "solve" time and the preconditioner setup time. Table 1 presents results for the Taylor-Green turbulent vortex decay test case with 128^3 element cube mesh (16.8M DOFs) run on 256 cores for various smoothers: sub-domain Symmetric Gauss-Seidel (SGS) with no overlap, ILU(0) with overlap of 1 ("ILU0ov1"), global GMRES

smoother	preconditioner	GMRESR iter/Δt	Time(s)				mem (MB)
			total	prec	solve	linear solve	
SGS		87.7	1108	18	730	748	1050
ILU0ov1		14.2	732	263	97	360	1440
GMRES	noprec	15.4	697	22.7	267	290	917
	bkJac	13.1	655	35	238	273	927
	ptGS	13.6	828	24	413	437	917
	bkGS	12.0	950	34	539	573	930
DD-GMRES	noprec ov0	21.4	749	21.1	336	357	890
	noprec ov1	15.9	814	102	334	436	1080
	bkJac ov0	20.2	785	32.6	391	424	892
	bkJac ov1	15.5	895	127	371	498	1082
	ptGS ov0	20.2	1173	20.8	793	814	892
	ptGS ov1	12.1	1084	116	611	727	1080
	bkGS ov0	20.2	1365	31.7	975	1007	891
	bkGS ov1	11.9	1218	110	757	867	1081

Table 1: Taylor-Green turbulent MHD vortex decay test case with 128^3 element cube mesh (16.8M DOFs) run on 256 cores of an Intel Xeon Sandy Bridge linux cluster with InfiniBand fat-tree interconnect. Columns 3–8: GMRESR iterations per time step (sum of GMRESR iterations over the Newton steps in a time step), Drekar total wall time, total preconditioner setup time, total linear system solve time or preconditioned iteration time, sum of total preconditioner setup and solve time, maximum high water memory over MPI processes.

("GMRES") smoother with no preconditioner ("noprec") as well as block Jacobi ("bkJac"), point Gauss-Seidel ("ptGS") and block Gauss-Seidel ("bkGS"), and additive Schwarz/domain decomposition GMRES smoother with the same preconditioners as the global case, but with either no overlap or one level of overlap ("ov0" or "ov1"). The third column is the outer GMRESR iterations per time step, so it is the sum of the GMRESR iterations over the Newton steps within the time step. So there is actually an interaction between the nonlinear Newton solver and linear Krylov solver. Fourth column is the total time of the simulation, including internal mesh generation and Jacobian and residual fill for the 20 time steps. The fifth and sixth columns are the sum of all the preconditioner setup times and sum of all the linear system solve times respectively for the the 20 time steps (each time step has multiple Newton steps). The seventh column is the total linear solve time for the 20 time steps (sum of columns 5 and 6) and is the metric for the comparison of the different smoothers. The eighth

column is the maximum high watermark memory usage over the MPI process[1]. In general, standard relaxation smoothers are not sufficiently robust for our MHD test cases, and we seldom employ them. Here we can see that the outer GMRESR iterations per time for SGS smoother is considerably higher than for any smoother, which makes it uncompetitive compared with the standard ILU smoother (more than double the linear solve time). When the mesh is uniformly refined to the 256^3 element cube mesh (134M DOFs) in Table 2, with SGS smoother the outer GMRESR Krylov solver no longer converges (convergence stalls). Therefore we do not consider the SGS smoother for the 512^3 element cube mesh (Table 3) or for any further test cases in this work. For the standard ILU smoother, the cost for smoother setup (time to compute the ILU factors) is very expensive, and considerably larger than the solve time. For the GMRES smoothers, while the preconditioner setup time is inexpensive compared to ILU, the solve time is considerably more expensive compared to the ILU smoother. GMRES smoother with either no preconditioner or block Jacobi preconditioner is 20% and 25% faster (for linear solve time) than the standard ILU smoother respectively, while requiring only 66% of the memory. Additive Schwarz/domain decomposition GMRES smoother with no overlap without any preconditioner is the same speed as the standard ILU smoother. The additive Schwarz/domain decomposition GMRES smoother with either point or block Gauss-Seidel is not competitive. Note that the additive Schwarz/domain decomposition GMRES smoother with one level of overlap gives similar GMRESR iterations as the global GMRES smoother (with the same preconditioner). Unfortunately constructing the smoother with one level of overlap is considerably more expensive than the no overlap case. For all the cases involving GMRES smoother, five iterations of GMRES are employed (five iterations are employed for all the studies in this paper). Table 2 presents the comparison of smoothers after the mesh has been refined uniformly one level. Other than for the SGS smoother for which the outer GMRESR Krylov solver no longer converges (convergence stalls), the trends are similar to the 128^3 element cube mesh case.

Table 3 presents the comparison of smoothers after the mesh has been refined uniformly another level, to 1.1B DOFs on 16,384 cores. The GMRES smoother with either no preconditioner or block Jacobi preconditioner is 10% and 23% faster (for linear solve time) than the standard ILU smoother respectively, while requiring only 66% of the memory. For ILU smoother, the factorization is expensive, but once the factors are obtained, applying the factors is considerably less expensive. For the GMRES smoother, the preconditioner setup is inexpensive, but the solve time is expensive. It is a trade-off between the expensive ILU factorization for setup versus the expensive solve for GMRES

[1]Note the memory usage shown in all the results in this paper is for a research and development testbed code for the flexible development of multiphysics applications and evaluation of a wide range of iterative nonlinear/iterative linear solver algorithms that is unoptimized. Currently efforts are underway to reduce the overall memory usage outside of the iterative solvers and hence the benefit in the percentage decrease in memory usage for employing the Krylov smoother would actually increase from the levels described in the study.

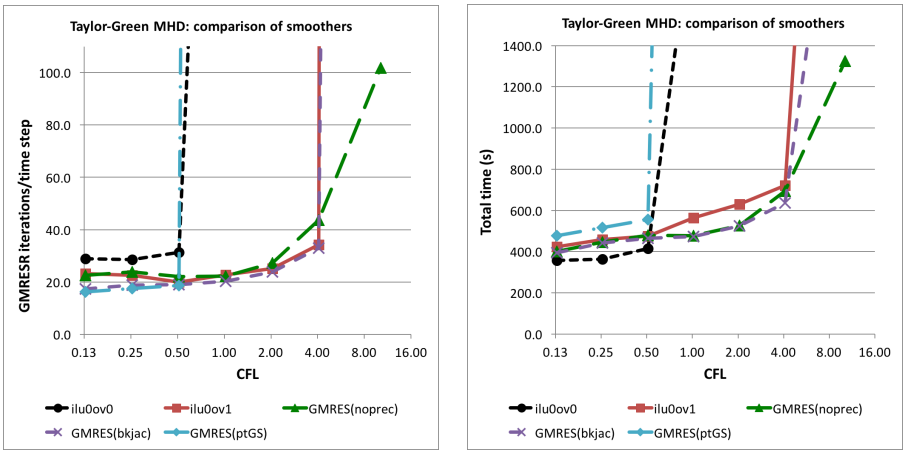
smoother	preconditioner	GMRESR iter/Δt	Time(s)				mem (MB)
			total	prec	solve	linear solve	
SGS		Failed					
ILU0ov1		20.2	862	311	134	445	1440
GMRES	noprec	18.7	741	23.5	328	352	920
	bkJac	15.7	749	41.3	306	348	933
	ptGS	16.9	921	26.8	516	542	920
	bkGS	17.1	1376	40.9	954	995	936
DD-GMRES	noprec ov0	31.4	969	23	565	588	896
	noprec ov1	23.9	1229	138	676	814	1077
	bkJac ov0	28.1	992	38	572	610	901
	bkJac ov1	17.8	1173	147	610	757	1086
	ptGS ov0	27.4	1625	27.0	1212	1239	895
	ptGS ov1	16.9	1700	119	1197	1316	1088
	bkGS ov0	27.6	1963	38	1544	1582	901
	bkGS ov1	16.9	1920	134	1402	1536	1091

Table 2: Taylor-Green turbulent MHD vortex decay test case with 256^3 element cube mesh (134M DOFs) run on 2048 cores of an Intel Xeon Sandy Bridge linux cluster.

smoother	preconditioner	GMRESR iter/Δt	Time(s)				mem (MB)
			total	prec	solve	linear solve	
ILU0ov1		37.3	1515	414	276	690	1520
GMRES	noprec	31.0	1563	34	590	624	1000
	bkJac	21.9	1398	51	482	533	1020
	ptGS	20.0	1660	47	1176	1223	1023
	bkGS	20.0	2089	110	1204	1314	
DD-GMRES	noprec ov0	77.0	2361	57.7	1519	1577	1030
	noprec ov1	54.1	3118	288	1732	2020	
	bkJac ov0	69.5	2519	76.2	1545	1621	1040
	bkJac ov1	34.0	2389	227	1227	1454	
	ptGS ov0	62.6	4044	45.1	3163	3208	1040
	ptGS ov1	33.5	3896	186	2667	2853	
	bkGS ov0	59.1	4745	161	3524	3685	1040
	bkGS ov1	32.1	4073	280	2888	3168	

Table 3: Taylor-Green turbulent MHD vortex decay test case with 512^3 element cube mesh (1.1B DOFs) run on 16,384 cores of an Intel Xeon Sandy Bridge linux cluster.

smoother. The GMRES smoother can considerably lower the number of outer GMRESR iterations, but cost per iteration is not inexpensive. The additive Schwarz/domain decomposition GMRES smoother is not competitive with the standard ILU smoother, especially for the cases with Gauss-Seidel preconditioner (either point or block), although the memory requirement is significantly less than for the standard ILU smoother.



(a) (b)

Figure 1: Taylor-Green turbulent MHD vortex decay 128^3 element test case comparing (a) GMRESR iterations per time step and (b) total time for different smoothers as CFL is increased (Δt is increased).

For transient simulations, the ability to take larger time steps significantly reduces the time for a simulation (for the case where numerical stability constrains the time step). Figure 1 compares the robustness of five of the smoothers with increasing CFL, i.e. increasing time step, for the 128^3 element cube mesh case. Figure 1(a) plots outer GMRESR iterations per time step while Figure 1(b) plots total time (total time for entire simulation, i.e. including mesh generation, Jacobian and residual construction, etc.). One can observe that the ILU(0) with no overlap smoother and GMRES preconditioned by point Gauss-Seidel smoother no longer converges for CFL > 0.5. ILU(0) with one level of overlap smoother, GMRES preconditioned by block Jacobi smoother and GMRES with no preconditioner smoother still converge for CFL $= 4$, which is a time step $8\times$ larger. Similar trends hold for the refined meshes.

4.2. Island Coalescence

The island coalescence problem follows the unstable evolution of two 3D current tubes (in the cross plane - islands) embedded in a sheered magnetic field Harris sheet [7]. The structure of this equilibrium can be seen in the upper left plot of Figure 2 with and iso-surface of $\mathbf{J}$. The combined magnetic

smoother	preconditioner	GMRESR	Time(s)				mem
		iter/Δt	total	prec	solve	linear solve	(MB)
ILU0ov1		11.9	918	410	62	472	917
GMRES	noprec	17.5	719	25	244	270	631
	bkJac	12.9	705	42	204	246	631
DD-GMRES	noprec ov1	17.1	905	122	331	454	735
	bkJac ov1	14.7	905	139	324	463	735

Table 4: Island coalescence test case with 64^3 element cube mesh (2.1M DOFs) run on 256 cores of an Intel Xeon Sandy Bridge linux cluster.

field produced by the flux tubes produces Lorentz forces that pull the flux tubes together, and at finite resistivity the islands coalesce (join) to form one island. Figure 2 shows iso-surfaces in the 3D volume and iso-lines and filled color contours on intersecting planes of J_z during the reconnection event. Clearly evident is the formation of the the x-point in the intersecting planes between the islands (see images at t = 2 and 3), the development of thin current sheets at that same x-point location (and the corresponding 3D surface), and the movement of the center of the tubes (island o-points) towards the x-point [52, 53]. The dynamics of island coalescence changes as a function of resistivity. For larger resistivities, the x- and o-points monotonically approach each other. For low resistivities, fluid-plasma pressure builds up as the islands approach and a sloshing or bouncing of the o-point position is encountered that leads to lower reconnection rates (for more details on the physics see e.g. [52]).

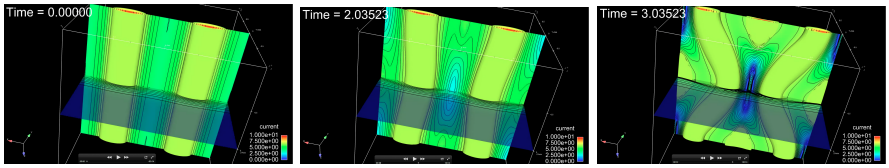


Figure 2: Structure of the current tubes in 3D island coalescence problem with $S = 2 \times 10^4$ for the initial condition, and two times in the evolution of the problem $t = 2, 3$. The 3D current tubes have bent in the z-direction and form current sheets as indicated by the iso-lines and filled color contours of plasma current at five locations.

Tables 4–6 present a weak scaling study for the problem described in [7] for 64^3 element cube mesh (2.1M DOFs), 128^3 element cube mesh (16.9M DOFs) and 256^3 element cube mesh (135M DOFs) with Lundquist number 10^3. For this study, the time step is fixed at 0.1 and run to simulation time 4.0 (40 time steps). Each time the mesh is refined, the CFL is doubled. For the 256^3 element cube mesh the time scale for Alfven wave is associated with $CFL_A = (\frac{B}{\sqrt{\rho\mu_0}}\Delta t)/\Delta x \approx 15$. Table 4 presents a comparison of smoothers for the 64^3 element cube mesh (2.1M DOFs) run on 64 cores of the SNL TLCC2 linux cluster. For both the global GMRES smoother, or additive Schwarz/domain decomposition GMRES smoother, when either a point or block

smoother	preconditioner	GMRESR	Time(s)				mem
		iter/Δt	total	prec	solve	linear solve	(MB)
ILU0ov1		15.1	1022	455	96	552	907
GMRES	noprec	21.6	871	30	378	407	645
	bkJac	14.6	792	50	362	412	632
DD-GMRES	noprec ov1	20.0	1261	154	638	793	738
	bkJac ov1	16.1	1182	169	551	721	736

Table 5: Island coalescence test case with 128^3 element cube mesh (16.9M DOFs) run on 512 cores of an Intel Xeon Sandy Bridge linux cluster.

smoother	preconditioner	GMRESR	Time(s)				mem
		iter/Δt	total	prec	solve	linear solve	(MB)
ILU0ov1		15.5	1125	528	110	638	922
GMRES	noprec	21.8	911	67	380	448	666
	bkJac	15.5	865	88	317	406	653
DD-GMRES	noprec ov1	20.4	1442	191	768	959	753
	bkJac ov1	15.9	1311	206	648	854	753

Table 6: Island coalescence test case with 256^3 element cube mesh (135M DOFs) run on 4096 cores of an Intel Xeon Sandy Bridge linux cluster.

Gauss-Seidel preconditioner (subdomain preconditioner) is employed the outer GMRESR stalled. For the 64^3 element cube mesh, all the Krylov smoothers are faster than the standard ILU smoother, with the global GMRES smoother with no preconditioner or block Jacobi preconditioner being 40–50% faster. Table 5 presents a comparison of smoothers for the 128^3 element cube mesh (16.9M DOFs) run on 512 cores. Here the global GMRES smoother is faster than the standard ILU smoother, but the additive Schwarz/domain decomposition GMRES smoothers are slower. Table 6 presents a comparison of smoothers for the 256^3 element cube mesh (135M DOFs) run on 4096 cores. As with the previous two coarser meshes, the global GMRES smoother is faster than the standard ILU smoother (30–35% reduction in time) while only requiring 65% of the memory, but as with the previous coarser mesh the additive Schwarz/domain decomposition GMRES smoothers are slower (but requires 20% less memory than ILU).

4.3. 3D hydromagnetic Kelvin-Helmholtz shear layer

The final test problem is a 3D hydromagnetic Kelvin-Helmholtz unstable shear layer [7] (Figure 3). The computational domain is $[-2,2]\times[0,4]\times[-1,1]$ and the initial condition is defined by two counter flowing conducting fluid streams with constant velocities $U(x, y > 0, z, 0) = (1, 0, 0)$ and $U(x, y < 0, z, 0) = (-1, 0, 0)$ and a Harris sheet sheared magnetic field defined by

$$\mathbf{B}(x, y, z, 0) = (0, B_0 tanh(y/\delta), 0)\,.$$

The boundary conditions are periodic on the right and left as well as the front and back and no flow through the top and bottom. The parameters were chosen so that $Re = 10^4$, the magnetic Reynolds number $Re_m = 10^4$ and the magnetic field strength is selected so that the Alfvenic Mach number $M_A = v/v_A = 3$.

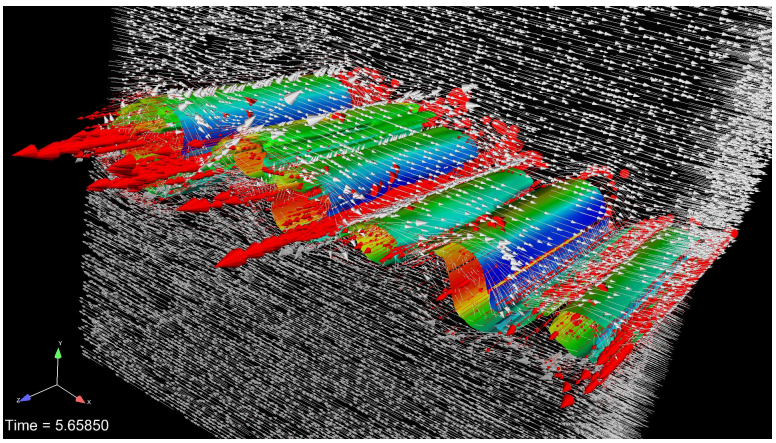


Figure 3: 3D hydromagnetic Kelvin-Helmholtz shear layer with $Re = 10^4, Re_m = 10^4, M_A = 3$. The white vectors show magnetic field, the red vectors the plasma current, and the surface is of constant vorticity colored by the plasma current.

Figure 4 presents a weak scaling study for the 3D hydromagnetic Kelvin-Helmholtz shear layer test case with 532K DOFs (run on 32 cores of an Intel Xeon Sandy Bridge linux cluster), 4.23M DOFs (256 cores) and 33.7M DOFs (2048 cores). All simulations were run to the same simulation time while the velocity CFL was kept fixed, which halved the time step for each uniform refinement of the mesh. The number of time steps were doubled as the mesh was refined, therefore the 532K DOFs, 4.23M DOFs and 33.7M DOFs test cases were run with 480, 960 and 1920 time steps, respectively. Figure 4(a) presents preconditioner setup time and linear system solve time per Newton step. The leftmost three columns are with ILU(0) with one level of overlap smoother for 32, 256 and 2048 MPI processes, respectively. The middle three columns are with GMRES smoother with no preconditioner for 32, 256 and 2048 MPI processes, respectively. The rightmost three columns are with GMRES smoother with block Jacobi preconditioner for 32, 256 and 2048 MPI processes, respectively. For the ILU smoother, the time to construct the multigrid preconditioner (mostly ILU smoother factorization time) is significantly greater than the solve time. For the GMRES smoother, the time to construct the preconditioner is significantly less than the solve time. Figure 4(b) presents the outer GMRESR iterations per Newton step for the 9 runs. The outer GMRESR iterations per Newton step for the GMRES smoother with block Jacobi preconditioner is similar to ILU smoother, while the GMRES smoother with no preconditioner requires more outer GMRESR iterations. Figure 4(c) presents maximum high water mark memory over MPI processes for the 9 runs. The standard ILU

smoother requires 40–50% more memory than the GMRES smoothers.

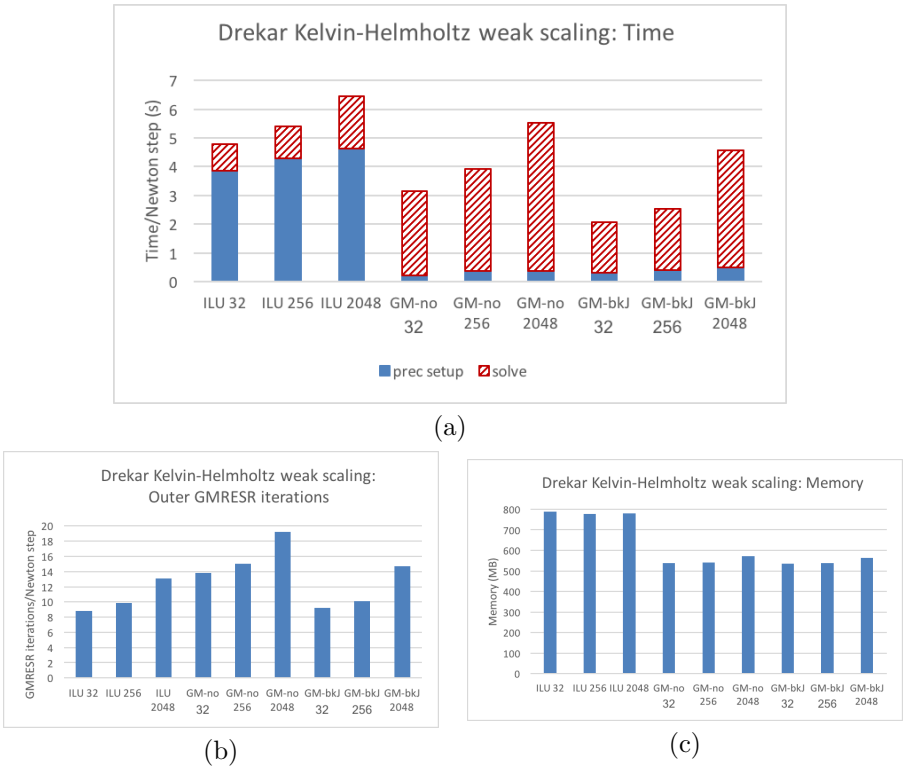


Figure 4: 3D hydromagnetic Kelvin-Helmholtz shear layer test case weak scaling study with 532K DOFs (run on 32 cores of an Intel Xeon Sandy Bridge linux cluster), 4.23M DOFs (256 cores) and 33.7M DOFs (2048 cores). (a) Preconditioner setup time and linear system solve time per Newton step. The leftmost three columns are with ILU smoother for 32, 256 and 2048 MPI processes, respectively. The middle three columns are with GMRES smoother with no preconditioner for 32, 256 and 2048 MPI processes, respectively. The rightmost three columns are with GMRES smoother with block Jacobi preconditioner for 32, 256 and 2048 MPI processes, respectively. (b) Outer GMRESR iterations per Newton step for the 9 runs. (c) Maximum over MPI processes of high water mark memory for the 9 runs.

More test cases need to be considered to build more confidence in the Krylov smoother. But it can be faster due to reduction in outer GMRESR solver iterations and requires less memory (typically 35% less memory for global GMRES smoother) than the standard ILU smoother.

5. Conclusions

In this study, we evaluated the use of Krylov smoothers for multigrid as an alternative smoother to our robust but expensive in terms of time and memory

standard ILU smoother for our fully-coupled Newton-Krylov algebraic preconditioned multigrid solution approach for large-scale VMS resistive MHD simulations. Our study considered three transient MHD simulations, but more test cases need to be considered to build more confidence in the Krylov smoother. The GMRES smoother can be faster due to reduction in outer GMRESR solver iterations and requires less memory (typically 35% less memory for global GMRES smoother) than our standard ILU smoother. However, the conventional wisdom heading towards the exascale era is that algorithms should reduce communication. Unfortunately the global Krylov smoother actually increases communication. Hence our interest in additive Schwarz/domain decomposition Krylov smoothers. Our next step is to evaluate the Krylov smoother at very large scales to see how the global Krylov smoother compares with the domain decomposition Krylov smoother and standard ILU smoother. Even if it does not perform as well as ILU for very large scales, it holds promise for being a competitive and robust alternative to ILU at scales in the range of 100M to 1 billion DOFs.

6. Acknowledgments

The authors would like to thank R. Pawlowski and E. Cyr for their collaborative effort in developing the Drekar MHD code and D. Sondak and T. Smith for the collaborative development of the VMS LES turbulent MHD modeling capability. The authors gratefully acknowledge the generous support of the DOE NNSA ASC program and the DOE Office of Science AMR program at Sandia National Laboratories. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy, National Nuclear Security Administration under contract DE-NA-0003525.

[1] H. Goedbloed, S. Poedts, Principles of Magnetohydrodynamics with Applications to Laboratory and Astrophysical Plasmas, Cambridge Univ. Press, 2004.

[2] E. Priest, T. Forbes, Magentic Reconnection, MHD theory and applciations, Cambridge Univ. Press, 2000.

[3] L. Chacón, A non-staggered, conservative, $\nabla \cdot \mathbf{B} = \mathbf{0}$, finite-volume scheme for 3D implicit extended magnetohydrodynamics in curvilinear geometries, Comput. Phys. Comm. 163 (2004) 143–171.

[4] L. Chacón, An optimal, parallel, fully implicit Newton-Krylov solver for three-dimensional visco-resistive magnetohydrodynamics, Phys. Plasmas 15 (2008) 056103.

[5] J. N. Shadid, R. P. Pawlowski, J. W. Banks, L. Chacón, P. T. Lin, R. S. Tuminaro, Towards a scalable fully-implicit fully-coupled resistive MHD

formulation with stabilized fe methods, Journal of Computational Physics 229 (20) (2010) 7649–7671.

[6] S. C. Jardin, Review of implicit methods for the magnetohydrodynamic description of magnetically confined plasmas, J. Comp. Physics 231 (2012) 822–838.

[7] J. Shadid, R. Pawlowski, E. Cyr, R. Tuminaro, L. Chacon, P. Weber, Scalable implicit incompressible resistive MHD with stabilized FE and fully-coupled Newton–Krylov–AMG, Computer Methods in Applied Mechanics and Engineering 304 (2016) 1–25.

[8] Y. Saad, Iterative methods for sparse linear systems, SIAM, 2003.

[9] D. A. Knoll, D. E. Keyes, Jacobian-free Newton–Krylov methods: a survey of approaches and applications, J. Comput. Phys. 193 (2004) 357–397.

[10] K. Stüben, A review of algebraic multigrid, Journal of Computational and Applied Mathematics 128 (1) (2001) 281–309.

[11] U. Trottenberg, C. Oosterlee, A. Schüller, Multigrid, Academic Press, London, 2001.

[12] M. Brezina, J. Hu, R. Tuminaro, Algebraic multigrid, in: D. Padua (Ed.), Encyclopedia of Parallel Computing, Springer, 2011, pp. 23–33.

[13] P. Lin, J. Shadid, J. Hu, R. Pawlowski, E. Cyr, Performance of fully-coupled algebraic multigrid preconditioners for large-scale VMS resistive MHD, Journal of Computational and Applied Mathematics, *accepted* 0 (2017) 0–0.

[14] J. Shadid, A fully-coupled Newton-Krylov solution method for parallel unstructured finite element fluid flow, heat and mass transfer simulations, Int. J. CFD 12 (1999) 199–211.

[15] J. Shadid, R. Tuminaro, K. Devine, G. Henningan, P. Lin, Performance of fully-coupled domain decomposition preconditioners for finite element transport/reaction simulations, J. Comput. Phys. 205 (1) (2005) 24–47.

[16] J. Shadid, A. Salinger, R. Pawlowski, P. Lin, G. Hennigan, R. Tuminaro, R. Lehoucq, Large-scale stabilized FE computational analysis of nonlinear steady-state transport/reaction systems, Computer methods in applied mechanics and engineering 195 (13) (2006) 1846–1871.

[17] P. T. Lin, J. N. Shadid, R. S. Tuminaro, M. Sala, G. L. Hennigan, R. P. Pawlowski, A parallel fully coupled algebraic multilevel preconditioner applied to multiphysics PDE applications: Drift-diffusion, flow/transport/reaction, resistive MHD, International Journal for Numerical Methods in Fluids 64 (10-12) (2010) 1148–1179.

[18] P. T. Lin, M. Sala, J. N. Shadid, R. S. Tuminaro, Performance of fully coupled algebraic multilevel domain decomposition preconditioners for incompressible flow and transport, Int. J. Num. Meth. Eng. 67 (2) (2006) 208–225.

[19] B. Smith, P. Bjorstad, W. Gropp, Domain decomposition: parallel multilevel methods for elliptic partial differential equations, Cambridge university press, 2004.

[20] A. Quarteroni, A. Valli, Numerical approximation of partial differential equations, Vol. 23, Springer Science & Business Media, 2008.

[21] R. Bank, C. Douglas, Sharp estimates for multigrid rates of convergence with general smoothing and acceleration, SIAM J. Numer. Anal. 22 (1983) 617–633.

[22] F. Bornemann, P. Deuflhard, The cascadic multigrid method for elliptic problems, Numerische Mathematik 75 (1996) 135–152.

[23] V. Shaidurov, Some estimates of the rate of convergence for the cascadic conjugate-gradient method, Computers & Mathematics with Applications 31 (1996) 161–171.

[24] D. Braess, On the combination of the multigrid method and conjugate gradients, in: Multigrid Methods II, Vol. 1228 of Lecture Notes in Mathematics, 2006, pp. 52–64.

[25] Y. Notay, P. Vassilevski, Recursive Krylov-based multigrid cycles, Numer. Linear Algebra Appl. 15 (2008) 473–487.

[26] H. Elman, O. Ernst, D. O'Leary, A multigrid method enhanced by krylov subspace iteration for discrete Helmholtz equations, SIAM Journal on Scientific Computing 23 (2001) 1291–1315.

[27] P. Birken, J. Bull, A. Jameson, A study of multigrid smoothers used in compressible CFD based on the convection diffusion equations, in: M. Papadrakakis, V. Papadopoulos, G. Stefanou, V. Plevris (Eds.), Proceedings of the VII European Congress on Computational Methods in Applied Sciences and Engineering ECCOMAS Congress 2016, TU Vienna, Crete Island, Greece, June 2016.

[28] P. Lin, J. Shadid, J. Hu, P. Tsuji, Performance of smoothers for algebraic multigrid preconditioners for finite element variational multiscale incompressible magnetohydrodynamics, in: Proceedings of the SIAM Conference on Parallel Processing for Scientific Computing (PP16), SIAM, April 2016.

[29] R. Codina, N. Hernández-Silva, Stabilized finite element approximation of the stationary magento-hydrodyanamics equations, Comput. Mech. 38 (2006) 344–355.

[30] P. N. Brown, Y. Saad, Hybrid Krylov methods for nonlinear systems of equations, SIAM J. Sci. Stat. Comput. 11 (1990) 450–481.

[31] M. Benzi, Preconditioning techniques for large linear systems: a survey, Journal of Computational Physics 182 (2002) 418–477.

[32] A. Quarteroni, A. Valli, Domain Decomposition Methods for Partial Differential Equations, Oxford University Press, Oxford, 1999.

[33] X.-C. Cai, M. Sarkis, A restricted additive Schwarz preconditioner for general sparse linear systems, SIAM J. Sci. Comput. 21 (1999) 792–797.

[34] W. Briggs, V. Henson, S. McCormick, A Multigrid Tutorial, Second Edition, SIAM, Philadelphia, 2000.

[35] W. Hackbusch, Multi-grid Methods and Applications, Springer-Verlag, Berlin, 1985.

[36] P. T. Lin, J. N. Shadid, M. Sala, R. S. Tuminaro, G. L. Hennigan, R. J. Hoekstra, Performance of a parallel algebraic multilevel preconditioner for stabilized finite element semiconductor device modeling, Journ. Comp. Physics 228 (2009) 6250–6267.

[37] M. Gee, C. Siefert, J. Hu, R. Tuminaro, M. Sala, ML 5.0 smoothed aggregation user's guide, Tech. Rep. SAND2006-2649, Sandia National Laboratories (2006).

[38] P. Vaněk, J. Mandel, M. Brezina, Algebraic multigrid based on smoothed aggregation for second and fourth order problems, Computing 56 (1996) 179–196.

[39] P. Lin, Improving multigrid performance for unstructured mesh drift-diffusion simulations on 147,000 cores, International Journal for Numerical Methods in Engineering 91 (2012) 971–989.

[40] M. Heroux, R. Bartlett, V. Howle, R. Hoekstra, J. Hu, T. Kolda, R. Lehoucq, K. Long, R. Pawlowski, E. Phipps, A. Salinger, H. Thornquist, R. Tuminaro, J. Willenbring, A. Williams, An Overview of Trilinos, Tech. Rep. SAND2003-2927, Sandia National Laboratories (2003).

[41] R. S. Tuminaro, M. Heroux, S. A. Hutchinson, J. N. Shadid, Aztec user's guide–version 2.1, Tech. Rep. SAND99-8801J, Sandia National Laboratories, Albuquerque NM, 87185 (Nov. 1999).

[42] K. Devine, E. Boman, R. Heaphy, B. Hendrickson, C. Vaughan, Zoltan data management services for parallel dynamic applications, Computing in Science & Engineering 4 (2) (2002) 90–97.

[43] M. Sala, M. Heroux, Robust algebraic preconditioners with IFPACK 3.0, Tech. Rep. SAND2005-0662, Sandia National Laboratories (2005).

[44] S. Balay, W. D. Gropp, L. C. McInnes, B. F. Smith, Efficient management of parallelism in object oriented numerical software libraries, in: E. Arge, A. M. Bruaset, H. P. Langtangen (Eds.), Modern Software Tools in Scientific Computing, Birkhäuser Press, 1997, pp. 163–202.

[45] R. Falgout, A. Baker, E. Chow, V. Henson, E. Hill, J. Jones, T. Kolev, B. Lee, J. Painter, C. Tong, P. Vassilevski, U. Yang, Users manual, HYPRE high performance preconditioners, Tech. rep., Lawrence Livermore National Laboratory (2002).

[46] Y. Saad, M. Schultz, GMRES: a generalized minimial residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comp. 7 (1986) 856–869.

[47] Y. Saad, A flexible inner-outer preconditioned GMRES algorithm, SIAM J. Sci. Comput. 14 (12) (1993) 461–469.

[48] H. V. der Vorst, GMRESR: a family of nested GMRES methods, Numerical Linear Algebra with Applications 1 (1994) 369–386.

[49] A. Pouquet, E. Lee, M. Brachet, P. Mininni, D. Rosenberg, The dynamics of unforced turbulence at high reynolds number for taylor–green vortices generalized to mhd, Geophysical and Astrophysical Fluid Dynamics 104 (2010) 115–134.

[50] E. Lee, M. Brachet, A. Pouquet, P. Mininni, D. Rosenberg, Lack of universality in decaying magnetohydrodynamic turbulence, Physical Review E 81 (2010) 016318.

[51] D. Sondak, J. Shadid, A. Oberai, R. Pawlowski, E. Cyr, T. Smith, A new class of finite element variational multiscale turbulence models for incompressible magnetohydrodynamics, Journal of Computational Physics 295 (2015) 596–616.

[52] D. Biskamp, Magnetic Reconnection in Plasmas, Cambridge University Press, Cambridge, UK, 2000.

[53] D. A. Knoll, L. Chacón, Coalescence of magnetic islands, sloshing, and the pressure problem., Phys. Plasmas 13 (3) (2006) 32307 – 1.