Fully-Implicit Ultrascale Physics Solvers and Application to Ion Source Modelling

Kris Beckwith
and Seth Veitzer
Tech-X Corporation
5621 Arapahoe Ave.
Boulder, CO 80303
Email: beckwith@txcorp.com
and veitzer@txcorp.com

Stephen F. McCormick
and John W. Ruge
Front Range
Scientific Computations, Inc.
Email: stephen.mccormick@Colorado.edu
and John.Ruge@colorado.edu

Luke N. Olson
and Jon C. Cahoun
Dept. of Computer Science
University of Illinois
at Urbana-Champaign
Email: lukeo@illinois.edu
and jccalho2@illinois.edu

***Abstract*—Many problems of interest in plasma modelling are subject to the 'tyranny of scales', specifically, problems that encompass physical processes that operate on timescales that are separated by many orders of magnitude. Investigating such problems therefore requires the use of implicit time-integration schemes, which advance problem solutions on the timescale of interest, while incorporating the physics of the fast-timescales. One promising route to develop these implicit solvers is the combination of Jacobian-Free Newton- Krylov methods (JFNK, [1]), but adapting these methods to work in ultra scale computing environments is a formidable challenge. Here, we describe research on new approaches to adapt Algebraic Mulgrid based solvers (that can be used for provide efficient pre-conditioners for JFNK methods) to ultrascale computing environments, the development and testing of JFNK solvers for coupled plasma-electromagnetics within the USIM framework [2] and the application of these methods to modelling H^- ion sources for the Spallation Neutron Source at ORNL.**

I. Introduction

In the near future, emerging high-performance computers with hundreds of thousands of processing cores will be used to study a range of coupled physics problems. However, at this scale of processor count, a redesign of existing linear and nonlinear solvers is necessary. Many conventional numerical solution methods do not scale well even at the ten-thousand-processor range; the hundred-thousand- processor range poses a major computational challenge to these algorithms. Moreover, processing at this scale is likely to experience soft and hard hardware errors that will lead to disastrous effects on conventional algorithms. Fault-resilient algorithms that maintain reasonable performance and accuracy while managing potentially severe faults in the system represent a critical need as the field approaches more ubiquitous computing at these ultra scales. In this work, we describe an effort to develop and apply solvers that work at ultra scale for a range of plasma-electromagnetics problems. This effort is comprised of three different strands: i) the development and testing of a novel multilevel decomposition methodology that yields ultra-scale performance in terms of efficiency, accuracy, and fault resiliency (described in §II); ii) development and testing of fault tolerance methods for multi grid solvers (described in §III; iii) effective treatment of non-linearities in fluid-plasma systems through Jacobian-Free Newton-Krylov methods (JFNK) (described in §IV) and iv) application of robust solvers to modelling H^- ion sources for the Spallation Neutron Source at ORNL (described in §V).

II. Solving Matrix-based Problems at Ultrascale using Algebraic MultiGrid

In modern large-scale supercomputing applications, Algebraic MultiGrid (AMG [3], [4]) is a leading choice for solving matrix equations because it offers a matrix solver for a wide variety of applications that scales nearly optimally with the size of the simulation. The challenge in general for matrix equation solvers on large parallel machines is that performance can suffer from the high cost of communication relative to that of computation, and this too is a concern for AMG. To ameliorate this concern, two new algebraic multilevel algorithms, Algebraic MultiGrid Domain Decomposition (AMG-DD) and Algebraic MultiGrid Range Decomposition (AMG-RD), have been developed that replace traditional AMG V-cycles with a fully overlapping domain decomposition approach.

While these methods are similar in spirit to the *geometric* methods developed by Brandt and Diskin [5], Mitchell [6], and Bank and Holst [7], their primary distinguishing feature is their purely *algebraic* nature: AMG-RD and AMG-DD trade communication for computation by forming global composite "grids" based only on the matrix, not the geometry. (As is the usual AMG convention, "grids" here should only be taken in the algebraic sense, regardless of whether or not it corresponds to any geometry.) Each processor grid is composite in the sense that it contains the original fine-grid nodes owned by its processor, but becomes increasingly coarse as it extends further away in the algebraic distance sense from these processor subdomain nodes. The important point here is that these global composite grids are constructed based only on components of an existing AMG setup. Once the existing AMG setup phase has determined effective coarse constructs (coarse grids, coarse operators, and intergrid transfer operators), then similarly effective composite grids can be constructed for each processor in a purely algebraic way. AMG-RD and AMG-DD are then able to exploit the full global overlap of the subdomains

978-1-4799-2713-5/14/$31.00

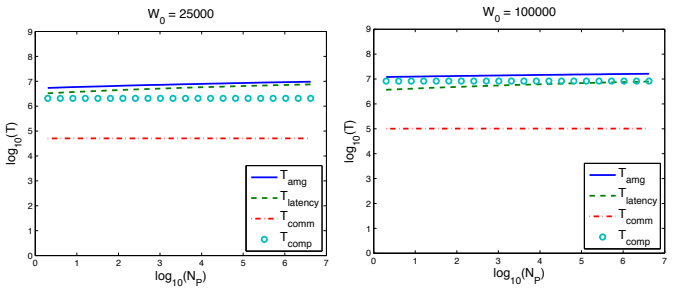

Fig. 1. AMG V-cycle costs for $W_0 = 25,000$ and $W_0 = 100,000$, respectively.

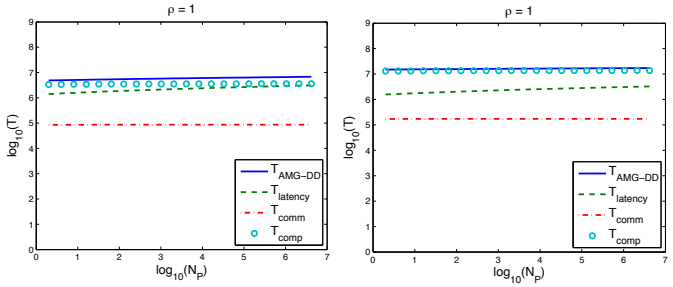

Fig. 2. AMG-DD costs for $W_0 = 25,000$, and $100,000$, $\eta = 2$, and $\rho = 1$.

to substantially reduce inter-processor communication and maintain optimal convergence.

Another important distinguishing feature of AMG-RD and AMG-DD is their novel residual communication process that enables effective parallel computation on composite grids, avoiding the all-to-all communication costs of the geometric methods. The main purpose of this paper is to study the potential of these two algebraic methods as possible alternatives to existing AMG approaches for future parallel machines. To this end, this paper develops some theoretical properties of these methods and reports on serial numerical tests of their convergence properties over a spectrum of problem parameters.

Table I shows sample convergence factors for AMG-DD applied to the model two-dimensional nine-point Poisson equation. For these results, each processor is assigned 5000 points of the original fine grid in its subdomain, Ω^p. The composite grid is constructed by padding Ω^p with the addition of η fine-grid lines just outside of Ω^p, then coarsening that padded region and adding η coarse-grid lines just outside of it, and continuing in this way until the coarse grid covers the whole domain. AMG-DD then proceeds by applying ρ AMG V-cycles to the composite grid problem before the N_p approximations are accumulated. The factors represent the ration of the Euclidean errors between subsequent AMG-DD iterations on the global residual equation. One important take-away from these results is that they appear to be almost independent of the number of processors that we selected. There is mild growth as the number of processors increases, but it is similar to growth exhibited by AMG.

To judge the potential of these methodologies in future architectures, AMG and AMG-DD/AMG-RD were studied

TABLE I
CONVERGENCE OF AMG-DD FOR 9, 16, 25, AND 36 PROCESSORS WITH 5000 POINTS PER PROCESSOR FOR VARYING PADDING (η) AND NUMBER OF CYCLES (ρ).

NumProcessors = 9

	$\rho = 1$	$\rho = 2$	$\rho = 3$	$\rho = 4$
$\eta = 1$	0.119	0.053	0.043	0.054
$\eta = 2$	0.105	0.037	0.031	0.031
$\eta = 3$	0.121	0.036	0.027	0.026
$\eta = 4$	0.098	0.037	0.023	0.028

NumProcessors = 25

	$\rho = 1$	$\rho = 2$	$\rho = 3$	$\rho = 4$
$\eta = 1$	0.147	0.073	0.061	0.062
$\eta = 2$	0.119	0.053	0.038	0.039
$\eta = 3$	0.122	0.049	0.035	0.035
$\eta = 4$	0.111	0.037	0.027	0.027

in the context of a parallel performance model developed by Gahvari et al. [8]. Their idealized model is based on three parameters:

- α: Latency cost per message, assumed to be independent of the length of connections.
- β: Inverse bandwidth cost or, more generically, cost per amount of data sent.
- γ: Flop rate of the machine, that is, the amount of work per computation.

Based on data in Table 2 from [8], α, β, γ are assumed to satisfy $\alpha = 10^4\gamma$ and $\beta = 10\gamma$.

Figure 1 shows the estimated V-cycle time for standard AMG as a function of the number of processors, for two different grid sizes (total fine-grid points of $n^2 = 25,000$ and $100,000$. Figure 2 shows the analogous results for AMG-DD with $\eta = 2$ and $\rho = 1$, which yields convergence factors that compare well to AMG V-cycles. Comparing Figures 1 and 2 confirms that, as desired, AMG-DD achieves the goal of reducing communication for computation. While latency does increase as the number of processors increases, the computation cost is now what drives the overall time per iteration of the algorithm. The amount of computation that has been added to each processor, as shown in Section 5.2.3, is on the same order of magnitude as the standard approach, so these figures also serve to highlight the balance between computation and communication that already exists in traditional AMG implementations.

III. FAULT RESILIENT APPROACHES FOR MULTIGRID SOLVERS

One critical aspect of sparse solvers as they scale to high numbers of processors is their resilience to faults. Emerging high-performance architectures are likely to exhibit higher rates of faults, and we anticipate that silent data corruption (SDC) to be a growing concern — i.e., faults in the system that do no result in node failure but only perturbed. The mean time between faults is expected to increase, which increases the likelihood of a fault disrupting the critical sparse solve.

The most common approaches to fault resilience are independent of the application and focus on checkpoint-restart

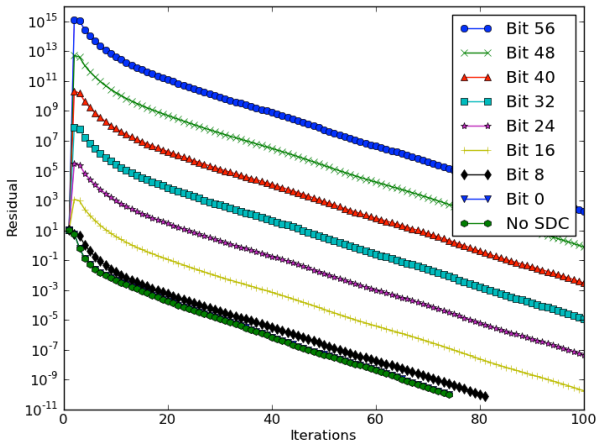

Fig. 3. Impact of a fault in the residual in the second iteration of a multigrid solver.

libraries at the system level. Our approach is to handle certain faults *within* the multigrid algorithm in order to enhance detection and extend reliability. Algorithmic level resilience provides the opportunity to lower the total time to solution in the presence of faults. As an example, consider a standard multigrid cycling algorithm consisting of smooth, residual, restrict, coarsen operations. If we inject a single fault into the residual calculation during the second iteration of the multigrid solve on the second coarsest level, we observe a range of outcomes as depicted in Figure 3. We see, not unexpectedly, that an SDC potentially leads to a drastic increase in the number of iterations (or failure to converge), in which case a restart is necessary at some point in the solver. On the other hand, a fault may have little or no impact in the convergence (for example if a high bit in the mantissa of one number is in error). In this case, we desire an algorithm that does not checkpoint and restart from memory, but instead proceeds with execution.

The approach to detecting SDCs exploits two pieces of the multigrid solver and has the advantage of easily adapting to different multigrid libraries. Given the large number of static loops in the cycling (and their deterministic execution),an inexpensive method for monitoring correct loop termination is to verify that the iteration has fully executed. While this form of loop checking intercepts many types of faults, other faults only manifest in the resulting convergence (cf. Figure 3). One potential indicator is the residual we expect the iterations to converge and since the residual is already available in the cycle, leading to very low overhead in monitoring the norm (which is commonly calculated regardless of resilience). Yet, placing a bound on the residual is likely not sharp, leading to only vague or imprecise detection. As an example, we place bounds of one or two orders on the jump in the residual between iteration. A more effective approach is to monitor the energetic stability of the iterations at *each* level. If A is symmetric, positive-definite, the projection process insists a reduction in the A-norm of the error, $\langle Ae_k, e_k\rangle^{1/2}$. This is not a realistic computation, but the relation ship

$$\langle Ae_k, e_k\rangle = \langle r_k, x^* - x_k\rangle = \langle Ax, x\rangle - 2\langle b, x\rangle + \langle b, x\rangle. \quad (1)$$

Thus a reduction in

$$\langle Ax, x\rangle - 2\langle b, x\rangle \quad (2)$$

is expected and is inexpensive to form after restricting to each level of the cycle and before interpolating.

TABLE II
CLASSIFICATION OF FAULTS INJECTION INTO A MULTIGRID SOLVER.

	Pointer	Control	Arithmetic
Percent of Injections	57.41	23.07	19.52

TABLE III
CONVERGENCE WITH SINGLE INJECTION

Results	No AMG Resilience	Low Cost AMG Resilience	Full AMG Resilience
Converge	66.4	99.8	100
Did Not Converge	1.2	0.2	0
Segfault	32.4	0	0

A. Numerical Tests

To test the impact of these low-overhead schemes, we consider a 2D anisotropic diffusion problem with $1e5$ unknowns on the fine level. A classical AMG hierarchy is constructed, resulting in seven levels and a V-cycle is performed with weighted Jacobi. The fault injection software FlipIt [9] is used with a rate of $1e-9$, which yields $\mathcal{O}(1)$ SDCs per solve for this size. FlipIt classifies its injections into three categories: *Pointer*, *Control*, and *Arithmetic*. The classification *Pointer* refers to all calculations directly related to use of a pointer (loads, stores, address calculation). *Control* refers to all calculations of branching and control flow (comparisons for branches and loop control variables modification). The final category of injected faults, *Arithmetic*, refers to the pure mathematical operations. Table II shows the categories of all injected faults over 1000 independent trials.

It is interesting to see that the majority of faults injected are into *Pointer* instructions. This is due impart to using sparse matrix data structures which requires several operations before the data is used.

With iterative linear solvers convergence can vary dramatically depending upon the initial guess and right hand side. To eliminate this variability we use the same initial guess and right hand side. Since an injected fault can cause an increased time to converge, in Table III we set the maximum number of iterations at 80. This allows 6 more iterations to converge instead of the fault free 74 iterations.

Even just a single injection has drastic impact upon convergence, with almost one-third of all runs segfaulting before converging. A high rate of segfaults is expected since we are injecting most of our faults into *Pointer* instructions. With our

Low Cost SDC detectors, checking loop termination and a bounded residual, and an algorithmic based recovery scheme we are able to recover from these segfaults and converge in almost all cases. The cases that didn't converge were very near convergence, and would have converged within a few extra iterations. The inclusion of (2), *Full*, we converge in all cases.

IV. Fully Implicit Solution Methods for Coupled Problems in Plasma-Electromagnetics

In this section, we describe methods for fully implicit solution of coupled problems in plasma-electromagnetics implemented in the USIM code [2]. In USIM, we have adopted Jacobian-Free Newton-Krylov (JFNK) methods [1] as a solver strategy, due to their high parallel scalability and adaptability to a wide-range of linear and non-linear systems. The adaptability of JFNK methods stems from the idea of writing the entire (non-linear) system as a functional:

$$R\left(Q\right) = 0 \tag{3}$$

For the case of a non-linear, anisotropic Poisson's equation, the functional takes the form:

$$R\left(Q\right) = \nabla\left[\mathbb{T}(Q)\nabla Q\right] \tag{4}$$

where $\mathbb{T}(Q)$ is a tensor diffusion coefficient that depends on Q and $Q = \phi$ is the potential. In USIM such problems are discretized using a moving least-squares method [2]. For a static mesh, this approach allows the computation of matrix coefficients at simulation start-up which can then be reused for the remainder of the simulation. This approach dramatically reduces simulation memory overhead and computation workload. For a hyperbolic problem, the functional takes the form:

$$R\left(Q\right) = \partial_t Q + \nabla\cdot\mathbf{F}\left(Q\right) \tag{5}$$

where Q is the vector of unknowns, e.g. for (inviscid) compressible hydrodynamics, $Q = \{\rho, \mathbf{m}, E\}$ while for (ideal) compressible magnetohydrodynamics, $Q = \{\rho, \mathbf{m}, E, \mathbf{B}\}$. Newton's method applied to this non-linear system gives:

$$\partial_Q R^k\left(Q^k\right)\delta Q^k = -R\left(Q^k\right) \tag{6}$$

where $\delta Q^k = Q^{k+1} - Q^k$ is the update to the vector of unknowns for the iteration k. The system is iterated to convergence, where at each Newton step, the linear system is solved using a Krylov method, e.g. Generalized Minimum Residual (GMRES). Convergence is defined by $||R^k||_2 < \epsilon_a + \epsilon_r||R^0||_2$, where $||\cdot||_2$ is the $\ell - 2$ norm, ϵ_a prevents convergence below roundoff and ϵ_r is the relative convergence tolerance. In this work, we utilize implementations of Jacobian-Free Newton-Krylov methods provided by the NOX Non-linear solver library [10], which is part of the Trilinos framework [11], with $\epsilon_r = 10^{-8}$.

A variety of temporal and spatial discretizations can be used to solve Eqn. (5). For the work presented here, we adopt the so-called θ-scheme for the temporal discretization:

$$\begin{aligned} R\left(Q^{n+1}\right) = Q^{n+1} - Q^n + \theta\delta t\nabla\cdot\mathbf{F}\left(Q^{n+1}\right) \\ + \left(1-\theta\right)\delta t\nabla\cdot\mathbf{F}\left(Q^n\right) = 0 \end{aligned} \tag{7}$$

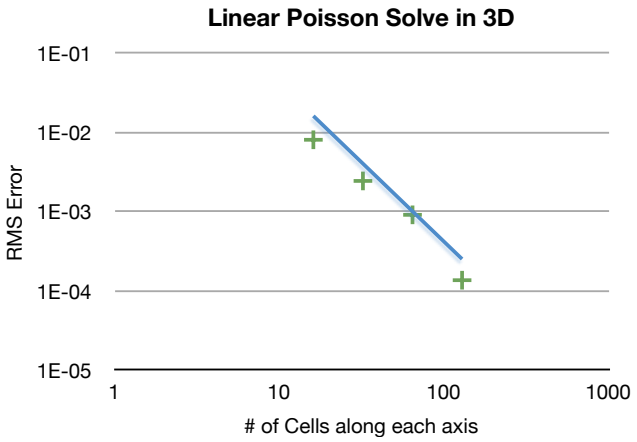

Fig. 4. RMS error for three-dimensional linear poisson problem computed with USIM demonstrating second order convergence

Here, $\theta = 0$ corresponds to a (first-order) forward Euler method, $\theta = 1$ corresponds to a (again first order) backward Euler method, while $\theta = 0.5$ corresponds to a *second-order* Crank-Nicolson method. For the spatial discretization, we adopt the popular finite-volume MUSCL algorithm which provides second-order accuracy for the hyperbolic systems considered here [2]. A key aspect of developing an efficient JFNK-based solver for hyperbolic problems is the adoption of an appropriate pre-conditioner. In the work presented here, no preconditioned was applied. Instead, at each time step, we initialize the JFNK-solver with a solution constructed from a forward-Euler integration ($\theta = 0$) over the time-step. This approach allows us to reach CFL~ 2.0 without difficultly [12].

A. Example Problem: Three-Dimensional Solution to Poisson's Equation

As an initial demonstration of the JFNK solver framework, we consider the solution of a three-dimensional Poisson problem, corresponding to the potential associated with a charged sphere [13], computed without the use of a preconditioned. As such, this problem is linear, In order to test solver performance for a *non-linear* poisson system, we introduce a scalar non-linear coefficient of the form $\kappa = \sqrt{\phi}$. Results for these two Poisson problems are shown in Figure's 4 and 5, which show convergence of RMS error and convergence rate for the linear and non-linear Poisson problems respectively. The RMS error is defined as:

$$\mathrm{RMS} = \frac{1}{V}\int\left|\phi_{\mathrm{compute}} - \phi_{\mathrm{analytic}}\right|^2 dxdydz \tag{8}$$

where $V = \int dxdydz$ is the volume of the mesh, ϕ_{compute} is the converged solution on the mesh and ϕ_{analytic} is the analytic solution to the problem, discretized onto the mesh. The results presented in 4 and 5 demonstrate that we are able to obtain second order convergence for both linear and non-linear Poisson problems using the NOX JFNK framework combined with the moving least squares algorithm implemented in USIM.

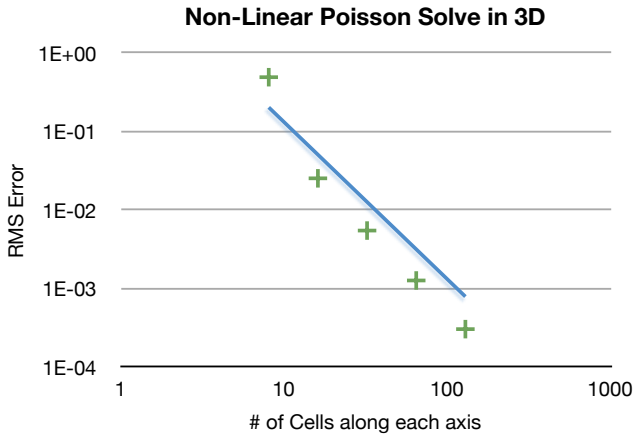

Fig. 5. RMS error for three-dimensional non-linear poisson problem computed with USIM demonstrating second order convergence

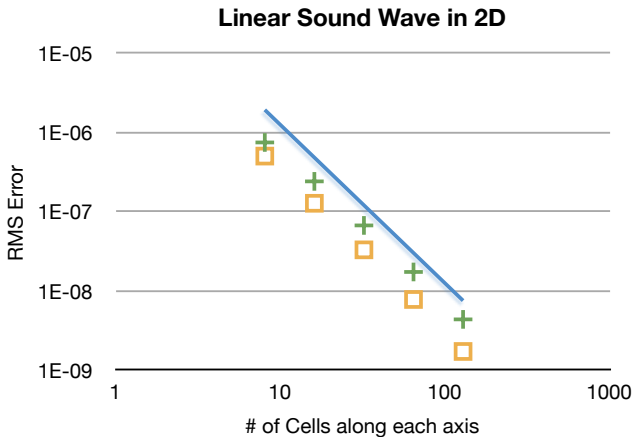

Fig. 6. RMS error for propagation of linearized sound waves, computed in multi-dimensions using USIM. Squares show results from an explicit second-order TVD Runge-Kutta scheme, crosses results from the JFNK solver.

B. Example Problem: Convergence of Linear Sound Waves in Multidimensions

A simple, yet discriminating test of the JFNK solver applied to hyperbolic problems is the convergence of linear modes of each wave family of inviscid hydrodynamics [14]. In this case, the hyperbolic system is given by:

$$Q = \begin{Bmatrix} \rho \\ \rho v_x \\ \rho v_y \\ \rho v_z \\ E \end{Bmatrix}; \quad F_x(Q) = \begin{Bmatrix} \rho v_x \\ \rho v_x v_x + P \\ \rho v_x v_y \\ \rho v_x v_z \\ (E+P)v_x \end{Bmatrix} \tag{9}$$

where ρ is the density, $\mathbf{v}$ is the velocity, P is the pressure, $E = \frac{P}{\gamma-1} + \frac{1}{2}\rho|\mathbf{v}|^2$ is the total energy, γ is the adiabatic index and the fluxes in the y- and z-directions can be found by permutation of indices. For this test, exact eigenfunctions for sound, contact and shear waves are initialized in a uniform medium with $\rho_0 = 1$, $P_0 = 3/5$ and adiabatic index (ratio of specific heats) $\gamma = 5/3$. To make the test multi-dimensional, the wave is initialized on the grid at an oblique angle, as described in [15]. The test is run until the wave has propagated once around the grid, using a CFL condition of 2.0 ($5\times$ the time step restriction for the explicit method), with $\theta = 0.5$ in order that the method gives overall second order convergence. The data of Figure 6 shows the RMS error for both the JFNK solver and an explicit method, based on MUSCL combined with a second-order total-value-diminishing (TVD) Runge-Kutta (RK) time-discretization (run with a CFL of 0.4). Following Stone et. (2008) [14], the RMS error is defined as:

$$\text{RMS} = \frac{1}{N_x N_y} \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \sum_{k}^{N_{\text{comp}}} \left| Q^{i,j,k}_{\text{final}} - Q^{i,j,k}_{\text{initial}} \right|^2 \tag{10}$$

where N_x, N_y are the number of cells in the x, y directions, N_{comp} is the number of components in the vector of unknowns (e.g. $N_{\text{comp}} = 5$ for inviscid hydrodynamics), $Q^{i,j,k}_{\text{final}}$ is the solution on the mesh at t_{final} and $Q^{i,j,k}_{\text{initial}}$ is the initial condition. The data of this Figure 6 demonstrates that both the JFNK method and the explicit method deliver second-order accuracy for linearized hyperbolic problems, although the RMS error associated with the JFNK method is approx. a factor 2 greater than that of the explicit method at a given grid resolution. We attribute this result to the dispersion error inherent in the Crank-Nicholson method.

C. Example Problem: Advection of a Weakly Magnetized Field Loop

Numerical solution of Maxwell's equations brings a unique set of challenges, due the need to preserve constraints on the electromagnetic fields:

$$\nabla \cdot \mathbf{e} = q/\epsilon_0 \;\; ; \;\; \nabla \cdot \mathbf{b} = 0 \tag{11}$$

While standard algorithms that preserve these constraints to machine precision for explicit approaches, these same algorithms have reduced effectiveness for *implicit* methods (see e.g. [12]). In order to preserve these constraints in a fashion consistent with the JFNK solver described here, we adopt a method based on Generalized Lagrange Multipliers (GLM) [16]. As a specific example, consider the equations of ideal magnetohydrodynamics:

$$Q = \begin{Bmatrix} \rho \\ \rho v_x \\ \rho v_y \\ \rho v_z \\ E \\ b_x \\ b_y \\ b_z \end{Bmatrix}; \; F_x(Q) = \begin{Bmatrix} \rho v_x \\ \rho v_x v_x + P_{\text{t}} - b_x^2 \\ \rho v_x v_y - b_x b_y \\ \rho v_x v_z - b_x b_z \\ (E+P_{\text{t}})v_x - (\mathbf{b}\cdot\mathbf{v})b_x \\ 0 \\ b_y v_x - b_x b_y \\ b_z b_x - b_x v_z \end{Bmatrix} \tag{12}$$

where $E = \frac{P}{\gamma-1} + \frac{1}{2}\rho|\mathbf{v}|^2 + \frac{1}{2}|\mathbf{b}|^2$ and $P_{\text{t}} = P + \frac{1}{2}|\mathbf{b}|^2$. In the GLM approach, an additional evolution equation is introduced that describes the evolution of the solenoidal part of the magnetic field, along with a modification to the induction

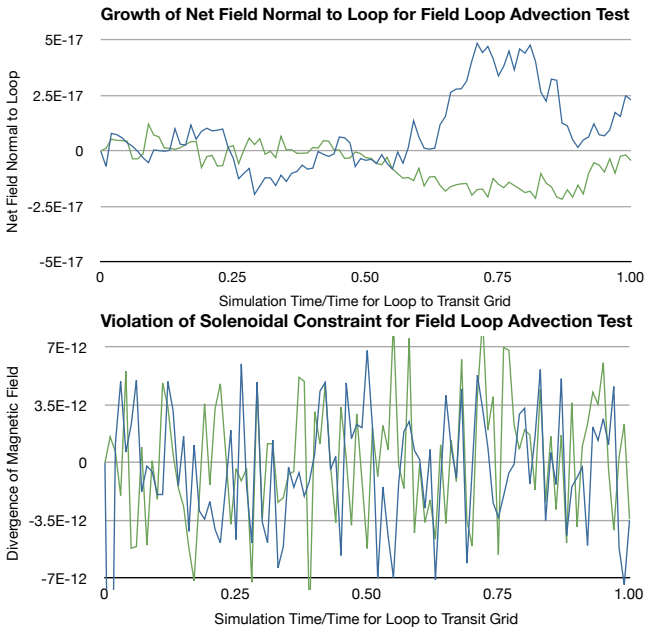

Fig. 7. Results from advection of a weakly magnetized field loop in two-dimensions using USIM. The top panel shows evolution of field normal normal to the loop, while the bottom panel shows divergence errors in the magnetic field. In both panels, green lines show results from an explicit second-order TVD Runge-Kutta scheme, blue lines show results from the JFNK solver.

equation:

$$\mathcal{D}(\psi) + \nabla \cdot \mathbf{b} = 0 \;\; ; \;\; \partial_t \mathbf{b} + \nabla\psi - \nabla \times (\mathbf{v} \times \mathbf{b}) = 0 \quad (13)$$

Dedner et al. [16] proposed adopting an operator, $\mathcal{D}(\psi)$ that causes the evolution of the scalar potential ψ to be described by an advection-diffusion equation:

$$\partial_t \psi + c_h^2 \nabla \cdot \mathbf{b} = -\frac{c_h^2}{c_p^2}\psi \quad (14)$$

which corresponds to advection of divergence errors (e.g. $\nabla \cdot \mathbf{b} \neq 0$) at speed c_h and damping of these errors at a rate c_h^2/c_p^2 [16]. Mignone et al. [17] propose that c_h^2 be chosen to be the fastest wave speed in the simulation domain and c_p determined by dimensional analysis. Compared to the ideal MHD system, this augmented MHD system contains two additional modes, which decouple into a 2×2 linear hyperbolic system which can be solved exactly [17]. For JFNK-based integration methods, this approach is advantageous as the evolution of the scalar potential, ψ can be incorporated into the non-linear residual, R, as opposed to previous, geometric approaches which are designed for explicit integration strategies [12].

A rigorous test of the ability of a scheme to evolve the MHD system without introducing divergence errors in the magnetic field is the advection of a weakly magnetized field loop [18]. In this test, the magnetic field normal to the plane of the loop (e.g. b_z) should remain zero for all time, which can only be maintained through balance of terms in the induction equation and preservation of $\nabla \cdot \mathbf{b} = 0$ [14]. For this test, we initialize the problem as described in [18] and evolve the loop for one crossing time using both the JFNK scheme (with CFL$= 2.0$) and a second-order accurate explicit scheme (with CFL$= 0.4$). The data of the top panel of Figure 7 shows the evolution of the magnetic field normal to the loop for both cases; while the data of the bottom panel of Figure 7 shows the evolution of divergence of the magnetic field. From these figures, we conclude that the JFNK method is able to keep divergence errors of the magnetic field to a similar magnitude as the explicit method and that little growth of the component of the field normal to the loop is observed. These results suggest that the combination of GLM methods with JFNK solver provides a robust way to minimize divergence errors for coupled plasma-electromagnetics simulations.

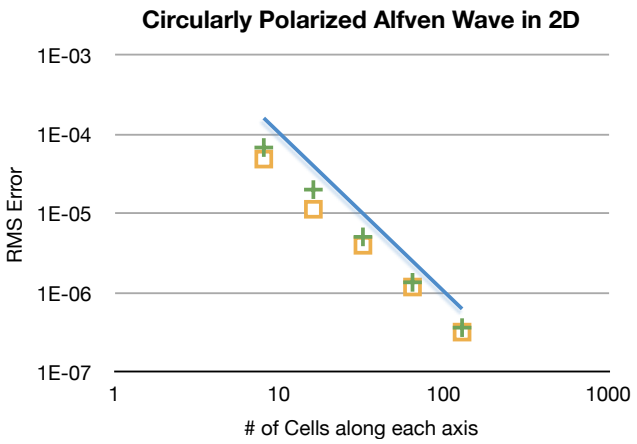

Fig. 8. RMS error for propagation of circularly polarized Alfven waves, computed in multi-dimensions using USIM. Squares show results from an explicit second-order TVD Runge-Kutta scheme, crosses results from the JFNK solver.

D. Example Problem: Convergence of Circularly Polarized Alfven Waves in Multidimensions

The final problem that we demonstrate here is the propagation of a circularly polarized Alfven wave. Such a wave are an exactly solution of the equations of MHD and were introduced by [19] as a sensitive test of the dispersion properties of algorithms for magnetohydrodynamics. For the purposes of this test, we initialize the wave as described in [18] at an oblique angle to the grid and evolve the wave for one crossing around the grid using both the JFNK solver (with CFL$= 2.0$) and a second-order accurate explicit scheme (with CFL$= 0.4$). We then measure the RMS error (defined as in Eqn. 10 with $N_{\rm comp} = 8$) compared to the initial solution discretized on the mesh, the results of which are shown in Figure 8. Note that the data of this figure *includes* divergence errors in the magnetic field. Both the JFNK-solver and the explicit scheme show overall second order convergence for the range of resolutions considered here and the errors are comparable between the two approaches. That is, for this non-linear problem, we do not see significant introduction of dispersion error by the use of a Crank-Nicholson time-discretization at the CFL considered here compared to a second-order Runge-Kutta method.

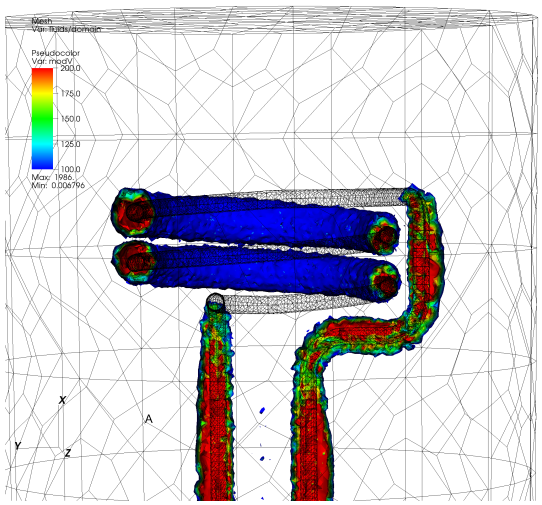

Fig. 9. Bulk plasma velocity near the antenna surface after $7.35\mu s$. High-velocity plasma near the antenna legs is parallel to the surface, but perpendicular near the bends.

V. Application: Modeling Ion Source Antenna Failure Using USim

Robust H^- ion sources are a key component for the Spallation Neutron Source at ORNL. Antennas used to produce H^- ions often fail when plasma heats the protective insulating coating on the antennas, exposing the bare metal and causing structural failure. Reducing antenna failures, which are expensive and reduce the operating capability of the source, is the top priority of the SNS H^- Source Program at ORNL [21]. Ablation of the ceramic dielectric coating at the antenna bends, where the antenna is exposed to the plasma is a common pattern of failure [20]. We are developing unique numerical models of plasma interactions with antenna surfaces using USim in order to provide optimizations of antenna design that will reduce antenna failure for the SNS.

The Spallation Neutron Source (SNS) is a state-of-the-art pulsed accelerator facility that pushes the intensity frontier for neutron beam production, providing unprecedented tools for researchers in physics, chemistry, biology, and materials science. Intense neutron beams are produced through spallation of high-energy protons on mercury, whereas the protons are accelerated through the driving linac as H^- ions, up to an energy of 1 GeV. The linac routinely operates at $\sim$ 40 mA and 1 MW, and is fed by the ion source with pulses of $\sim$ 1ms, $\sim$ 50mA H^- at a rate of 60 Hz. Given the high duty cycle and large amount of H^- required to drive the accelerator, the design requirements of the ion source include high-power components, intense plasmas, and high fields. Failures of the ion source under these conditions, particularly failures of the rf antenna used to produce H^- in the source, significantly affect the ability of the SNS to produce neutrons for science, and are costly to repair. Typically ion sources such as the SNS H^- source can only be accurately modeled if a large range of spatial and temporal scales are simulated. For instance, the SNS H^- source device is on the order of tens of cm in extent. The Debye length can be as low as $100\mu m$, meaning that there must be on the order of 1×10^9 computational cells in order to resolve plasma dynamics. rf frequencies for these types of sources are typically on the order of tens of MHz, requiring a time step of $\sim 1 \times 10^{-09} s$. If electromagnetic waves are to be resolved, the Courant-Friedrich-Lewy (CFL) criterion must be met in order to preserve numerical stability. This requires that dt $<$ dx/c, where dt is the time step, dx is the typical cell size, and c is the speed of light. So in order to resolve $10\mu m$ cells, the time step must be on the order of $1 \times 10^{-14} s$. Ion motion time scales are typically a thousand times greater than electron motions because of the mass difference. In addition, simulation time scales of many hundreds to thousands of rf periods are required in order to achieve steady state and quantify the plasma pressure on antenna surfaces.

There are a number of methods for relaxing simulation requirements, primarily utilizing different ways of increasing the time step and reducing the physical models to not include electron motion, while maintaining physical models that are accurate enough to resolve the primary effects of plasma interactions with antenna surfaces.

We have performed simulations in USim [22] using a single-fluid MHD model with an imposed magnetic field (due to the rf antenna current and static confining magnets) for the SNS Ion Source. We integrate the system given in Eqn. 12, with the magnetic field defined to be the sum of the magnetic field induced by plasma motion and the imposed magnetic field from the antenna, i.e. $\mathbf{b} = \mathbf{b}_{plasma} + \mathbf{b}_{antenna}$. The antenna magnetic field is computed by making the ansatz that the space-time evolution of the SNS H^- ion source antenna is separable, i.e. $\mathbf{J}(t, \mathbf{r}) = f(t)\mathbf{j}(\mathbf{r})$ This allows us, for example, to solve following equation for the magnetic vector potential at time $t = 0$, $\nabla^2 \mathbf{A}(\mathbf{r}) = \mathbf{j}(\mathbf{r})$. We solve this equation using the Jacobian-Free Newton-Krylov method described previously. The magnetic field generated by the SNS H^- ion source antenna is then derived according to: $\mathbf{B}_{antenna} = f(t)\nabla \times \mathbf{A}(\mathbf{r})$. The antenna field distribution is initialized according to the procedure described above and we choose $f(t)$ to be given by $f(t) = A \sin(\omega t)$ with A specified such that the current flowing through the antenna is $0.1A$ and $\omega = 2\pi f$ with $f = 13.6MHz$. We limit the simulation time step such that each antenna cycle is covered by 20.33 points in time; initial testing revealed this choice to represent a sensible compromise between physical accuracy and simulation cost, while using a non-integer number of points helps to precent aliasing. We evolve the system for 100 RF cycles (7.35×10^{-6} seconds) using the second order finite volume algorithms for solving the equations of single fluid magnetohydrodynamics implemented in USim The divergence of the induced magnetic field within the plasma is controlled by standard divergence cleaning techniques due to [16], [17]. For the choice of mesh, physical model and antenna design considered here, the simulation took approx. 1 hour on 8 Intel i7 3.4GHz cores. Note that the wall clock time necessary for the simulation is determined by the need to capture the frequency

of the antenna current.

Simulation results

Using this reduced physics model we observe increased bulk plasma velocity of $> 200m/s$ at the antenna surfaces where ablation of the ceramic coating is often seen in the SNS ion source. This is shown in Figure (9). We simulated 100 rf cycles, or $7.35\mu s$, using second-order finite volume algorithms for solving the single-fluid MHD equations (above). These simulations were performed on eight Intel i7 3.4 GHz cores and took about one hour of wall clock time. By comparison, it takes over one day of wall clock time to simulate 10 rf periods using explicit Finite Difference Time Domain Particle-In-Cell (FDTD-PIC) algorithms on a small cluster of 32 cores. Since single-fluid MHD equations assume quasi-neutrality, we are able to average over the plasma frequency and the Debye scale, and hence increase the time step. In these simulations we set the time step such that the rf period was sufficiently resolved. Our simulation results indicate that even reduced fluid models can provide insight into plasma dynamics at work in the SNS ion source.

VI. Conclusion

Jacobian-Free Newton-Krylov (JFNK) methods provide a promising route for fully-implicit solution strategies for coupled plasma-electromagnetic models. By combining JFNK methods with new approaches to fault tolerant Algebraic-Multigrid solvers, it is possible to develop solvers that are ready for ultra scale computing resources. We have implemented JFNK solvers within the USIM tool for coupled plasma-electromagnetics simulation and demonstrated second order convergence for a range of linear and non-linear standard plasma problems. Use of Generalized Lagrange Multipliers to control divergence errors in the magnetic field were demonstrated to produce physically accurate solutions when combined with JFNK solvers. Finally, these solvers were used to study ion source antenna failure for H^- ion sources that are a key component for the Spallation Neutron Source at ORNL. Simulation results demonstrated increased bulk plasma velocity of $> 200m/s$ at the antenna surfaces where ablation of the ceramic coating is often seen in the SNS ion source, suggesting that coupled fluid-electromagnetic models can provide valuable insights into H^- ion source failure.

Acknowledgment

This work supported by AFOSR under contract #FA95501210478; Program Officer: Dr Fariba Fahroo & Dr John Luginsland and under the auspices of the Department of Energy, Office of Basic Energy Sciences Award #DE-SC0009585.

References

[1] D. A. Knoll and D. E. Keyes, "Jacobian-free Newton-Krylov methods: a survey of approaches and applications," *Journal of Computational Physics*, vol. 193, pp. 357–397, Jan. 2004.

[2] J. Loverich, S. C. D. Zhou, K. Beckwith, M. Kundrapu, M. Loh, S. Mahalingam, P. Stoltz, and A. Hakim, *Nautilus: A Tool For Modeling Fluid Plasmas*. American Institute of Aeronautics and Astronautics, 2014/07/08 2013. [Online]. Available: http://dx.doi.org/10.2514/6.2013-1185

[3] A. Brandt, S. McCormick, and J. Ruge, "Algebraic multigrid (AMG) for sparse matrix equations." in *In Sparsity and its Applications, D.J, Evans (ed.).*, 1984, pp. 257–284.

[4] J. Ruge and K. Stüben, *Algebraic multigrid (AMG). In Multigrid Methods, vol. 5, McCormick SF (ed.).* SIAM: Philadelphia, PA., 1986.

[5] A. Brandt and B. Diskin, "Multigrid solvers on decomposed domains," in *Domain Decomposition Methods in Science and Engineering: The Sixth International Conference on Domain Decomposition*, ser. Contemporary Mathematics, vol. 157. Providence, Rhode Island: American Mathematical Society, 1994, pp. 135–155.

[6] W. Mitchell, "A parallel multigrid method using the full domain partition," *Electron. Trans. Numer. Anal.*, vol. 6, pp. 224–233, 1998.

[7] R. E. Bank and M. J. Holst, "A new paradigm for parallel adaptive meshing algorithms," *SIAM J. Sci. Stat. Comp.*, vol. 22, pp. 1411–1443, 2000.

[8] H. Gahvari, W. Gropp, K. E. Jordan, M. Schulz, and U. M. Yang, "Modeling the performance of an algebraic multigrid cycle using hybrid mpi/openmp," *2012 41st International Conference on Parallel Processing*, vol. 0, pp. 128–137, 2012.

[9] J. Calhoun, L. Olson, and M. Snir, "FlipIt: An LLVM based fault injector for HPC," in *Euro-Par 2014*, 2014, accepted.

[10] R. Pawlowski, J. Shadid, J. Simonis, and H. Walker, "Globalization techniques for newton–krylov methods and applications to the fully coupled solution of the navier–stokes equations," *SIAM Review*, vol. 48, no. 4, pp. 700–721, 2006. [Online]. Available: http://dx.doi.org/10.1137/S0036144504443511

[11] M. A. Heroux, R. A. Bartlett, V. E. Howle, R. J. Hoekstra, J. J. Hu, T. G. Kolda, R. B. Lehoucq, K. R. Long, R. P. Pawlowski, E. T. Phipps, A. G. Salinger, H. K. Thornquist, R. S. Tuminaro, J. M. Willenbring, A. Williams, and K. S. Stanley, "An overview of the trilinos project," *ACM Trans. Math. Softw.*, vol. 31, no. 3, pp. 397–423, 2005.

[12] M. F. Adams, R. Samtaney, and A. Brandt, "Toward textbook multigrid efficiency for fully implicit resistive magnetohydrodynamics," *Journal of Computational Physics*, vol. 229, pp. 6208–6219, Sep. 2010.

[13] J. M. Stone and M. L. Norman, "ZEUS-2D: A radiation magnetohydrodynamics code for astrophysical flows in two space dimensions. I - The hydrodynamic algorithms and tests." *ApJSS*, vol. 80, pp. 753–790, Jun. 1992.

[14] J. M. Stone, T. A. Gardiner, P. Teuben, J. F. Hawley, and J. B. Simon, "Athena: A New Code for Astrophysical MHD," *ApJSS*, vol. 178, pp. 137–177, Sep. 2008. [Online]. Available: http://adsabs.harvard.edu/abs/2008ApJS..178..137S

[15] T. A. Gardiner and J. M. Stone, "An unsplit Godunov method for ideal MHD via constrained transport in three dimensions," *Journal of Computational Physics*, vol. 227, pp. 4123–4141, Apr. 2008. [Online]. Available: http://adsabs.harvard.edu/abs/2008JCoPh.227.4123G

[16] A. Dedner, F. Kemm, D. Kröner, C.-D. Munz, T. Schnitzer, and M. Wesenberg, "Hyperbolic Divergence Cleaning for the MHD Equations," *Journal of Computational Physics*, vol. 175, pp. 645–673, Jan. 2002.

[17] A. Mignone and P. Tzeferacos, "A second-order unsplit Godunov scheme for cell-centered MHD: The CTU-GLM scheme," *Journal of Computational Physics*, vol. 229, pp. 2117–2138, Mar. 2010.

[18] T. A. Gardiner and J. M. Stone, "An unsplit Godunov method for ideal MHD via constrained transport," *Journal of Computational Physics*, vol. 205, pp. 509–539, May 2005. [Online]. Available: http://adsabs.harvard.edu/abs/2005JCoPh.205..509G

[19] G. Tóth, "The $\nabla \cdot B = 0$ Constraint in Shock-Capturing Magnetohydrodynamics Codes," *Journal of Computational Physics*, vol. 161, pp. 605–652, Jul. 2000.

[20] M. P. Stockli, "Wide-leg antennas," *Configuration Controls Committee's Pre-Installation Review*, 2011.

[21] M. P. Stockli, B. X. Han, and R. Welton, "The sns h- hm source program: Scope, finding, issues, puzzles, and plans," in *Proc. 2nd rf H- Ion Source Workshop*, Giardini Naxos, Italy, 2011.

[22] J. Loverich and A. Hakim, "Two-dimensional modeling of ideal merging plasma jets," *J. Fusion Energy*, vol. 29, no. 6, 2010.