

## A fully implicit 3D extended magnetohydrodynamics algorithm

L. Chacón<sup>1</sup>

<sup>1</sup>*Los Alamos National Laboratory, Los Alamos, NM 87545, USA*

### Introduction

The extended magnetohydrodynamics model (XMHD) includes effects such as nonlinear transport and two-fluid (Hall) effects. XMHD supports multiple, disparate time scales that make explicit time-differencing approaches extremely inefficient. While a fully implicit implementation promises efficiency without sacrificing numerical accuracy [1], the nonlinear nature of the XMHD system and the numerical stiffness of its fast waves make this endeavor very difficult.

Newton-Krylov methods are, however, ideally suited for such a task. They combine Newton's method for nonlinear convergence and Krylov techniques to solve the associated Jacobian (linear) systems. Krylov methods can be implemented in a Jacobian-free fashion (avoiding forming and storing the Jacobian matrix), and can be preconditioned for efficiency [2, 3]. The preconditioning step considers an inexpensive approximation to the Jacobian inverse to enhance the Krylov iteration convergence rate, and can be easily incorporated into the Krylov algorithm.

The focus of this document is to discuss our preconditioning strategy for the 3D primitive-variable resistive MHD formalism (details of the spatial discretization employed can be found in Ref. [4]). It is based on "physics-based" ideas [2, 3], in which knowledge of the physics is exploited to derive well-conditioned (diagonally dominant) approximations to the original system that are amenable to optimal multilevel solver technologies. Grid convergence studies will demonstrate that CPU time scales scale optimally as  $\mathcal{O}(N)$ , where  $N$  is the number of unknowns, and that the number of Krylov iterations scales as  $\mathcal{O}(N^0)$ .

### Implicit Time Integration and Newton-Krylov methods

Computationally, MHD has been dominated by explicit methods. While conceptually simple, explicit techniques are inappropriate for multiple-time-scale problems, since they must resolve the fastest time scale supported by the model (which may be orders of magnitude faster than the dynamical time scale of interest). Implicit methods can step over fast time scales to accurately resolve the dynamical time scale, but a large-scale system of nonlinear equations must be solved in order to advance the solution in time. Nevertheless, large CPU speedups over explicit methods are possible [2, 3], as will be shown here.

In the context of fully implicit time integration, we employ Newton-Krylov methods [5], which are an implementation of Newton's method in which a Krylov subspace method is used

to approximately solve the required Jacobian linear systems. Specifically, let  $\mathbf{G} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ , where  $\mathbf{v} \in \mathbb{R}^n$  is a current approximate solution to  $\mathbf{G}(\mathbf{v}) = \mathbf{0}$ . The Krylov method is applied to determine a solution of the Newton equation  $\mathbf{J}(\mathbf{v})\mathbf{s} = -\mathbf{G}(\mathbf{v})$ , where  $\mathbf{J} \in \mathbb{R}^{n \times n}$  is the Jacobian matrix at  $\mathbf{v}$ . In this work, we employ GMRES [5] as the Krylov method of choice, which is suitable for non-definite, non-symmetric systems. For efficiency, an inexact Newton method [6] is usually employed, whereby the linear tolerance is adjusted as the nonlinear iteration proceeds to avoid over-solving early in the iterative procedure.

Since Krylov methods only require matrix-vector products to proceed, a Jacobian-free implementation (JFNK) is possible whereby the required matrix-vector products are approximated by a finite-difference approximation  $\mathbf{J}\mathbf{s} \approx [\mathbf{G}(\mathbf{v} + h\mathbf{s}) - \mathbf{G}(\mathbf{v})]/h$ , for a suitably chosen differencing parameter  $h$ . This avoids forming and storing the Jacobian matrix.

Another advantage of Newton-Krylov methods is the possibility of preconditioning. Preconditioning consists in operating on the system matrix  $\mathbf{J}$  with an operator  $\mathbf{P}^{-1}$  (preconditioner) such that  $\mathbf{J}\mathbf{P}^{-1}$  (right preconditioning) or  $\mathbf{P}^{-1}\mathbf{J}$  (left preconditioning) is well-conditioned (i.e.,  $\mathbf{P}^{-1} \approx \mathbf{J}^{-1}$ ). Here, we focus on right preconditioning, where the equivalent system  $(\mathbf{J}\mathbf{P}^{-1})(\mathbf{P}\delta\mathbf{v}) = -\mathbf{G}(\mathbf{v})$  is considered. Thus, GMRES will solve  $(\mathbf{J}\mathbf{P}^{-1})\mathbf{z} = -\mathbf{G}(\mathbf{v})$ , and the Newton update  $\delta\mathbf{v}$  is found upon obtaining  $\mathbf{z}$  as  $\delta\mathbf{v} = \mathbf{P}^{-1}\mathbf{z}$ . Notice that the preconditioned system is equivalent to the original system for any nonsingular operator  $\mathbf{P}^{-1}$ . Thus, the choice of  $\mathbf{P}^{-1}$  does not affect the accuracy of the final solution, but crucially determines the efficiency of the algorithm.

The linear Jacobian systems that result from coupled multi-physics, multiple-time-scale systems are very ill-conditioned due to the presence of fast-time-scale phenomena (e.g., waves). An important new class of preconditioners for JFNK methods that addresses stiff hyperbolic systems (such as XMHD) is referred to as physics-based or PDE-based (e.g., see [2, 3]). Hyperbolic systems are generally unsuitable for multilevel approaches, which are desired because they result in a number of Krylov iterations virtually independent of the problem size (see [2, 3]). Physics-based methods attempt to reformulate such hyperbolic systems into parabolic ones, which are better suited for multilevel methods. We term this procedure ‘‘parabolization’’. The procedure for the resistive MHD system is described in the next section.

### Preconditioning strategy for resistive MHD

We consider the compressible resistive MHD model, given by:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0, \quad (1)$$

$$\frac{\partial \mathbf{B}}{\partial t} - \nabla \times (\mathbf{u} \times \mathbf{B}) + \nabla \times \left( \frac{\eta}{\mu_0} \nabla \times \mathbf{B} \right) = 0, \quad (2)$$

$$\frac{\partial(\rho\mathbf{u})}{\partial t} + \nabla \cdot \left[ \rho\mathbf{u}\mathbf{u} - \frac{\mathbf{B}\mathbf{B}}{\mu_0} + \overleftrightarrow{T} \left( p + \frac{B^2}{2\mu_0} \right) - \rho\nu\nabla\mathbf{u} \right] = 0, \quad (3)$$

$$\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T + (\gamma - 1)T\nabla \cdot \mathbf{u} = 0, \quad (4)$$

where the plasma is assumed polytropic  $p \propto n^\gamma$ , with  $\gamma$  the polytropic constant,  $p = (n_i + n_e)T = 2nT$  the pressure,  $T = T_i = T_e$  the ion/electron temperature, and  $n = n_i = n_e$  the ion/electron density. In these equations,  $\rho = m_i n$  is the ion mass density,  $\mathbf{u}$  is the plasma velocity,  $\mathbf{B}$  is the magnetic field,  $\eta$  is the resistivity,  $\nu$  is the kinematic viscosity. The linearized resistive MHD model in Eqs. 1–4 has the following coupling structure:

$$\mathbf{J}\delta\mathbf{x} = \begin{bmatrix} \mathbf{D}_\rho & 0 & 0 & \mathbf{U}_{u\rho} \\ 0 & \mathbf{D}_T & 0 & \mathbf{U}_{uT} \\ 0 & 0 & \mathbf{D}_B & \mathbf{U}_{uB} \\ \mathbf{L}_{\rho u} & \mathbf{L}_{Tu} & \mathbf{L}_{Bu} & \mathbf{D}_u \end{bmatrix} \begin{pmatrix} \delta\rho \\ \delta T \\ \delta\mathbf{B} \\ \delta\mathbf{u} \end{pmatrix} = - \begin{pmatrix} \mathbf{G}_\rho \\ \mathbf{G}_T \\ \mathbf{G}_B \\ \mathbf{G}_u \end{pmatrix},$$

where subscripts label the relevant dependent variables. The diagonal blocks  $\mathbf{D}$  contain advection/diffusion contributions, and are “easy” to invert using multilevel techniques. Off-diagonal blocks  $\mathbf{L}$  and  $\mathbf{U}$  contain all hyperbolic couplings, and are the source of hyperbolic stiffness in the MHD model. Note the Jacobian matrix has an “arrow” structure. To derive an approximate inverse of this Jacobian matrix, we consider the following  $2 \times 2$  block structure:

$$\mathbf{J}\delta\mathbf{x} = \begin{bmatrix} \mathbf{M} & \mathbf{U} \\ \mathbf{L} & \mathbf{D}_u \end{bmatrix} \begin{pmatrix} \delta\mathbf{y} \\ \delta\mathbf{u} \end{pmatrix} = - \begin{pmatrix} \mathbf{G}_y \\ \mathbf{G}_u \end{pmatrix},$$

with  $\mathbf{M}$  containing the diagonal blocks for  $\rho$ ,  $\mathbf{B}$ , and  $T$ . The Schur decomposition (factorization) of the  $2 \times 2$  block Jacobian  $\mathbf{J}$  yields:

$$\begin{bmatrix} \mathbf{M} & \mathbf{U} \\ \mathbf{L} & \mathbf{D}_u \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{I} & 0 \\ -\mathbf{L}\mathbf{M}^{-1} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{M}^{-1} & 0 \\ 0 & \mathbf{P}_{\text{Schur}}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\mathbf{M}^{-1}\mathbf{U} \\ 0 & \mathbf{I} \end{bmatrix},$$

with  $\mathbf{P}_{\text{Schur}} = \mathbf{D}_u - \mathbf{L}\mathbf{M}^{-1}\mathbf{U}$ . The *exact* Jacobian inverse only requires  $\mathbf{M}^{-1}$  and  $\mathbf{P}_{\text{Schur}}^{-1}$ . The term  $\mathbf{L}\mathbf{M}^{-1}\mathbf{U}$  in  $\mathbf{P}_{\text{Schur}}$  is effectively a parabolic operator obtained from the hyperbolic couplings  $\mathbf{L}$ ,  $\mathbf{U}$ , thereby “parabolizing” the MHD equation. However, a direct treatment of  $\mathbf{P}_{\text{Schur}}$  is impractical from the efficiency standpoint due to the presence of  $\mathbf{M}^{-1}$ , and we need suitable approximations. One such simplification, valid in the limit of small transport coefficients and low-speed flows, is  $\mathbf{M}^{-1} \approx \Delta t \mathbf{I}$  (i.e., we neglect advection and diffusion effects) in  $\mathbf{P}_{\text{Schur}}$  and  $-\mathbf{M}^{-1}\mathbf{U}$  above, which results in the following three-step preconditioning algorithm:

$$\begin{aligned} \delta\mathbf{y}^* &= -\mathbf{M}^{-1}\mathbf{G}_y \\ \delta\mathbf{u} &\approx \mathbf{P}_{\text{SI}}^{-1}[-\mathbf{G}_u - \mathbf{L}\delta\mathbf{y}^*]; \mathbf{P}_{\text{SI}} = \mathbf{D}_u - \Delta t\mathbf{L}\mathbf{U} \\ \delta\mathbf{y} &\approx \delta\mathbf{y}^* - \Delta t\mathbf{U}\delta\mathbf{u} \end{aligned}$$

We employ multigrid methods (MG) to approximately invert  $\mathbf{P}_{\mathbf{SI}}$  and  $\mathbf{M}$ , since both are block-diagonally dominant by construction. We demonstrate the performance of such a preconditioner with the tearing mode problem in 2D Cartesian geometry described in Ref. [4]. We employ two classical MG V(4,4) cycles (with under-damped Jacobi as a smoother) in the preconditioning stage to approximately obtain  $\mathbf{M}^{-1}$  and  $\mathbf{P}_{\mathbf{SI}}^{-1}$ . Grid scaling results are presented in Table 1 for an implicit time step  $\approx 1100$  larger than the explicit CFL condition. Notice from the table that the CPU scaling is optimal. Table 2 shows the results of a time-step scaling study, demonstrating favorable scaling of the number of GMRES iterations per time step with time step size, and implicit CPU speedups vs. explicit approaches of an order of magnitude.

Grid	$\Delta t$	Newton/ $\Delta t$	GMRES/ $\Delta t$	CPU
32x32	6	6	34.7	222
64x64	3	5.8	22.9	593
128x128	1.5	5.6	14.8	1700

Table 1: Grid convergence study. Computations span 10 time steps.

$\Delta t$	Newton/ $\Delta t$	GMRES/ $\Delta t$	CPU (s)	$CPU_{exp}/CPU$	$\Delta t/\Delta t_{CFL}$
0.5	4.9	8.4	1189	5.1	380
0.75	5.7	10.2	1361	6.7	570
1.0	5.0	11.5	1400	8.7	760
1.5	5.6	14.8	1700	10.7	1140

Table 2: Time step convergence study for a  $128 \times 128$  grid. Computations span 10 time steps.

## References

- [1] D. A. Knoll, L. Chacón, L. Margolin, and V. A. Mousseau, J. Comput. Phys. **185**, 583 (2003).
- [2] L. Chacón, D. A. Knoll, and J. M. Finn, J. Comput. Phys. **178**, 15 (2002).
- [3] L. Chacón and D. A. Knoll, J. Comput. Phys. **188**, 573 (2003).
- [4] L. Chacón, Comput. Phys. Comm. **163**, 143 (2004).
- [5] P. N. Brown and Y. Saad, SIAM J. Sci. Stat. Comput. **11**, 450 (1990).
- [6] R. Dembo, S. Eisenstat, and R. Steihaug, J. Numer. Anal. **19**, 400 (1982).