OPERATOR-BASED PRECONDITIONING OF STIFF HYPERBOLIC SYSTEMS∗

DANIEL R. REYNOLDS†, RAVI SAMTANEY‡, AND CAROL S. WOODWARD§

Abstract. We introduce an operator-based scheme for preconditioning stiff components encountered in implicit methods for hyperbolic systems of PDEs posed on regular grids. The method is based on a directional splitting of the implicit operator, followed by a characteristic decomposition of the resulting directional parts. This approach allows for the solution of any number of characteristic components, from the entire system to only the fastest, stiffness-inducing waves. We apply the preconditioning method to stiff hyperbolic systems arising in magnetohydrodynamics and gas dynamics. We then present numerical results showing that this preconditioning scheme works well on problems where the underlying stiffness results from the interaction of fast transient waves with slowly-evolving dynamics, scales well to large problem sizes and numbers of processors, and allows for additional customization based on the specific problems under study.

Key words. implicit methods, preconditioning, hyperbolic systems

AMS subject classifications. 65F10, 65Y20, 35L60

DOI. 10.1137/080713331

1. Introduction. The numerical solution of hyperbolic systems pervades the modeling of processes ranging from astrophysics and gas dynamics to magnetically confined fusion. A key feature for many of these processes is a significant stiffness of the system as compared with the physical time scales of interest, e.g., plasma refueling in tokamaks, magnetic reconnection in the earth’s magnetotail, and core collapse supernova [16, 20, 21, 31, 35, 37, 38]. As an example, in core collapse supernova, one solves a hyperbolic hydrodynamic system including a shock moving from the stellar core outward. Stability of an explicit scheme applied to this problem requires a time step governed by the CFL condition of motion in the core, which is $\sim 10^3$ times faster than the shock dynamics of interest. Hence, one would like to run the simulation at the larger time step governed by accuracy considerations of modeling the shock rather than the stability constraint in the core [37]. Additionally, while there have been rather significant advances in more diffusive simulations of Navier–Stokes and resistive magnetohydrodynamic (MHD) flows through the use of modern multigrid methods, such approaches fail at high Reynolds and Lundquist numbers. In many of these and other similar problems, such stiffness is induced through only a small number of the hyperbolic waves. Moreover, these stiff components may prohibit scaling of explicit simulations to the very high resolutions required in studying such processes. For these types of problems fully implicit solution techniques have recently

∗Received by the editors January 9, 2008; accepted for publication (in revised form) March 25, 2009; published electronically February 5, 2010. This work was supported by the U.S. DOE under UCSD contract DE-FC02-06ER25785 (DRR), by PPPL and Princeton University contract DE-AC02-76-CH03073, and under the auspices of the U.S. DOE by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. The U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes. Copyright is owned by SIAM to the extent not limited by these rights.
†Department of Mathematics, Southern Methodist University, Dallas, TX 75275-0156 (reynolds@smu.edu).
‡Princeton Plasma Physics Lab, MS 26, PO Box 451, Princeton, NJ 08543 (samtaney@pppl.gov).
§Lawrence Livermore National Lab, Box 808, L-561, Livermore, CA 94551 (cswoodward@llnl.gov).
been gaining favor \[6, 25, 27\]. Of primary importance in the use of fully implicit methods for such problems is that, if implemented with effective and scalable solvers, their lack of a stability restriction gives them the ability to perform such simulations more efficiently and/or more scalably than their explicit counterparts.

Even for nonstiff systems, implicit methods may prove beneficial as simulations increase in scale, since explicit methods can succumb to poor parallel weak scaling. As detailed in \[25\], parallel execution time (\(E\)) for a CFL-constrained explicit calculation is proportional to \(T S^{1+\alpha/d} P^{\alpha/d}\), where \(P\) is the number of processors, \(T\) is the simulation time interval, \(S\) is the problem size per processor, \(d\) gives the spatial dimensions, and \(\alpha\) is determined by the PDE under consideration (hyperbolic gives \(\alpha = 1\), parabolic gives \(\alpha = 2\)). Thus even for three-dimensional (3D) hyperbolic problems (best case scenario), explicit method execution times scale with processor count as \(E \propto P^{1/3}\); two-dimensional (2D) parabolic problems scale dramatically worse, with \(E \propto P\). As a result of this CFL stability dependence, explicit methods have difficulty demonstrating ideal weak scaling (\(E \propto 1\)). While this may not seem severe for moderate problem sizes, target applications on modern petascale computational architectures encounter \(P \approx 10^4 - 10^5\), at which level even explicit methods may not be utilized efficiently.

The application of implicit approaches usually requires the solution of a nonlinear system within each time step, typically solved with Newton’s method. For large-scale simulations on modern petascale machines, these nonlinear systems are solved with a Newton–Krylov method \[25\]. Because convergence of Krylov methods can stall, preconditioning is generally required for efficient solution of the linear systems.

Recent years have seen increased activity in the development and use of implicit solution approaches to hyperbolic systems as part of fluid dynamics applications (see \[19, 27\] and references therein). Generally, the emphasis of these works was to obtain steady state solutions for aerodynamics applications, although more recently the emphasis has shifted to unsteady problems. An excellent example of the latter is the work of Mavriplis \[29\] and Wang and Mavriplis \[40\], who investigated nonlinear multigrid methods as solvers and preconditioners with a Newton or Newton–Krylov technique using unstructured meshes. Another recent example in the aerodynamic literature for implicit solutions to unsteady problems \[5\] combined nonlinear multigrid with a Newton–Krylov approach, using line Jacobi and diagonally dominant alternating direction implicit approaches for preconditioning the linear system. Similarly, in the context of MHD, there has been significant recent work on the development of preconditioning approaches for fully implicit Newton–Krylov simulations, where the notion of “physics-based” preconditioners has been championed by Chacón \[9\] and Chacón, Knoll, and Finn \[6\]. Their approach relies on an approximate Schur factorization of the linear Newton system to “parabolize” the hyperbolic portions of the MHD system, making it amenable to multigrid techniques. The resulting preconditioners have been employed in 2D reduced MHD \[6\], and 3D resistive MHD \[9\]. Our work, described next, is distinguished from other works in that our preconditioner directly attacks the sources of stiffness in the system through an appropriate decomposition of the hyperbolic operator.

As evident in the literature, there are a variety of approaches for implicit solutions of hyperbolic systems arising in gas dynamics and MHD, with the dominant methods being nonlinear multigrid, Newton–Krylov, or a combination of both. In this paper, we present a systematic approach based on a preconditioned Newton–Krylov method to implicitly solve general stiff hyperbolic systems such as those encountered in applications from fluid dynamics, fusion, and astrophysics. The idea behind our
preconditioning approach is as follows: since the stiffness of the system results from
a small number of fast waves, we derive an approximate decomposition of the sys-
tem into its component waves and precondition only the stiffness-inducing parts. We
base this approximate decomposition on an $O(\Delta t^2)$-accurate splitting of the system
into its directional components, followed by a characteristic projection to decouple
the component waves. The result of this decoupling is a set of tightly banded lin-
ear systems, that we solve in parallel using a divide-and-conquer approach similar
to [1]. We note that within this strategy each characteristic equation is solved inde-
dependently, enabling solution of only those components inducing stiffness to the fully
implicit system, while leaving the slower components alone. Upon solution of these
decoupled equations, we project the preconditioned solutions back into their original
conserved variables, resulting in the approximate solution of the original linear
Jacobian system. This preconditioning scheme is valid for any method of lines ap-
proach to hyperbolic systems that employs a linear single or multistep method for
time integration, including those discussed in our earlier work using CVODE (high
order, adaptive, fixed-leading-coefficient BDF) and KINSOL (fixed-step $\theta$ method)
[22]. In this paper we focus on the fixed time-step Crank–Nicolson approach, in order
to better elucidate the effect of preconditioning outside the adaptive time-stepping
context. We note that extensions to the general hyperbolic system (e.g., reaction or
diffusion terms) may be preconditioned independently from the stiff waves using an
operator-split formulation.

This paper is organized as follows. In the next section we describe an implicit
preconditioned Newton–Krylov approach to solving a general system of hyperbolic
equations. In section 3 we show how the implicit approach is applied to the systems
of gas dynamics and MHD. We present numerical results for the preconditioning
strategy on a suite of test problems designed to investigate accuracy and scalability
in section 4 and give concluding remarks in section 5.

2. Preconditioned Newton–Krylov method for hyperbolic PDEs. In
this section, we describe the fully implicit formulation and Newton–Krylov solution
approach for general hyperbolic conservation laws.

2.1. Implicit solution approach. Consider the system

$$\partial_t U + \nabla \cdot \mathbf{F}(U) = 0,$$

where $U \equiv U(x,t) : \mathbb{R}^3 \times \mathbb{R} \to \mathbb{R}^n$ is a vector of $n$ conserved quantities, and $\mathbf{F}(U) \equiv \{F(U), G(U), H(U)\}^T : \mathbb{R}^n \to \mathbb{R}^{3n}$ is a vector of fluxes.

In numerically approximating solutions of the system (2.1), we follow a method of
lines approach for splitting the time and space dimensions. Under this methodology,
the spatial components are discretized following a spatial semidiscretization involving
a diagonal mass matrix (i.e., finite differences or finite volumes) on a regular spatial
grid. Time discretization of (2.1) is performed using a standard linear multistep
method that defines a nonlinear residual based on the parameters $\alpha_i, \beta_i,$ and $q,$

$$f(U^n) = U^n + \Delta t \beta_0 \nabla \cdot \mathbf{F}(U^n) - \sum_{i=1}^{q} [\alpha_i U^{n-i} + \Delta t \beta_i \nabla \cdot \mathbf{F}(U^{n-i})].$$

In this work we demonstrate results using a second-order Crank–Nicolson method in
which $f(U) = U^n - U^{n-1} + \Delta t \theta (\nabla \cdot \mathbf{F}(U^n) + \nabla \cdot \mathbf{F}(U^{n-1})),$ $\theta = 0.5$. Fully implicit
evolution of (2.1) over a time interval $[t^n, t^{n+1}]$ is then accomplished through solution
to the nonlinear root-finding problem \( f(U^n) = 0 \). To this end, we employ an inexact Newton–Krylov algorithm, as is standard for scalable parallel solution to coupled systems of PDEs [27].

Since all operations in the Newton–Krylov context require only linear complexity operations, the key component required for scalability of fully implicit simulations using this technology is an optimal preconditioning strategy for the inner Krylov linear solver [24, 27]. In Newton–Krylov algorithms, at each Newton iteration a Krylov iterative method is used to solve Jacobian systems of the form

\[
J(U)V = -f(U), \quad J(U) \equiv I + \gamma \frac{\partial}{\partial U} (\nabla \cdot F(U)), \quad \gamma = \Delta t \beta_0.
\]

The number of iterations required for convergence of a Krylov method depends on the eigenstructure of \( J \), where systems with clustered eigenvalues typically result in faster convergence than those with evenly distributed eigenvalues [18, 17, 39]. Unfortunately, for a fixed \( \Delta t \), as the spatial resolution is refined the distribution of these eigenvalues spreads, resulting in increased numbers of Krylov iterations and hence nonscalability of the overall solution algorithm. The role of a preconditioning operator \( P \) is to transform the original Jacobian system (2.3) to either

\[
JP^{-1}PV = -f \quad \text{(right prec.)} \quad \text{or} \quad P^{-1}JV = -P^{-1}f \quad \text{(left prec.).}
\]

The Krylov iteration is then used to solve one of

\[
(JP^{-1})W = -f \quad \text{or} \quad (P^{-1}J)V = X,
\]

where \( X = -P^{-1}f \) is computed prior to the Krylov solve or \( V = P^{-1}W \) is computed after the Krylov solve. Scalable convergence of the method then depends on the spectrum of the preconditioned operator \((JP^{-1} \text{ or } P^{-1}J)\), as opposed to the original Jacobian operator \( J \). Hence, an optimal preconditioning strategy will satisfy the two competing criteria:

1. \( P \approx J \), to help cluster the spectrum of the preconditioned operator.
2. Application of \( P^{-1} \) should be much more efficient than solution to the original system, optimally with linear complexity as the problem is refined and with no dependence on an increasing number of processors in a parallel simulation.

In the next section, we develop an operator-based preconditioning strategy for the system of hyperbolic conservation laws (2.1).

### 2.2. Preconditioner formulation

For linear multistep time integration approaches, we first rewrite the nonlinear problem (2.2) in the form

\[
f(U) = U + \gamma [\partial_x F(U) + \partial_y G(U) + \partial_z H(U)] + g = 0,
\]

where the terms \( F(U), G(U), \) and \( H(U) \) denote the \( x-, y-, \) and \( z- \)-directional hyperbolic fluxes, and the term \( g \) incorporates previous time-level information into the discretized problem. This nonlinear problem has Jacobian

\[
J(U) = I + \gamma [\partial_x (J_F(U)(\cdot)) + \partial_y (J_G(U)(\cdot)) + \partial_z (J_H(U)(\cdot))],
\]

with, e.g., \( J_F(U) = \frac{\partial}{\partial U} F(U) \). We use the notation \( (\cdot) \) to denote the location at which the action of the linear operator takes place, e.g.,

\[
[I + \gamma \partial_x (J_F(U)(\cdot))] V = V + \gamma \partial_x (J_F(U)V).
\]
Omitting the explicit dependence on \( U \) from the notation, and introducing nonsingular matrices \( L_F, L_G, \) and \( L_H \), we may rewrite the Jacobian system (2.5) as

\[
J = I + \gamma \left[ \partial_x \left( J_F L_F^{-1} L_F (\cdot) \right) + \partial_y \left( J_G L_G^{-1} L_G (\cdot) \right) + \partial_z \left( J_H L_H^{-1} L_H (\cdot) \right) \right]
\]

\[
= I + \gamma \left[ J_F L_F^{-1} \partial_x (L_F (\cdot)) + \partial_y \left( J_G L_G^{-1} L_G (\cdot) \right) + \partial_z \left( J_H L_H^{-1} L_H (\cdot) \right) \right]
\]

\[
+ J_G L_G^{-1} \partial_y (L_G (\cdot)) + \partial_y \left( J_G L_G^{-1} L_G (\cdot) \right) + \partial_z \left( J_H L_H^{-1} L_H (\cdot) \right)
\]

\[
+ J_H L_H^{-1} \partial_z (L_H (\cdot)) + \partial_z \left( J_H L_H^{-1} L_H (\cdot) \right)
\]

\[
= I + \gamma \left[ J_F L_F^{-1} \partial_x (L_F (\cdot)) \right] + J_G L_G^{-1} \partial_y (L_G (\cdot)) + J_H L_H^{-1} \partial_z (L_H (\cdot))
\]

\[
+ \partial_x \left( J_F L_F^{-1} \right) L_F (\cdot) + \partial_y \left( J_G L_G^{-1} \right) L_G (\cdot) + \partial_z \left( J_H L_H^{-1} \right) L_H (\cdot)
\]

The preconditioning scheme in this approach is based on the assumption that the majority of the stiffness found in the Jacobian is a result of a small number of very fast hyperbolic waves. To develop an approach for separately treating only these fast waves, we consider the preconditioning matrix, \( P \), constructed using a directional and operator-based splitting of \( J \),

\[
P = \left[ I + \gamma J_F L_F^{-1} \partial_x (L_F (\cdot)) \right] \left[ I + \gamma J_G L_G^{-1} \partial_y (L_G (\cdot)) \right] \left[ I + \gamma J_H L_H^{-1} \partial_z (L_H (\cdot)) \right]
\]

\[
= I + \gamma \partial_x \left( J_F L_F^{-1} \right) L_F (\cdot) + \gamma \partial_y \left( J_G L_G^{-1} \right) L_G (\cdot) + \gamma \partial_z \left( J_H L_H^{-1} \right) L_H (\cdot)
\]

\[
= J + O(\gamma^2).
\]

We denote these components as \( P = P_F P_G P_H P_{local} \). Through constructing the operator \( P \) as a product in this manner, the preconditioner solve consists of three simpler, one-dimensional (1D) implicit advection problems, along with one additional correction for spatial variations in the directional Jacobians \( J_F, J_G, \) and \( J_H \). Hence, \( Pu = b \) may be solved via the steps (i) \( P_F \chi = b \), (ii) \( P_G \omega = \chi \), (iii) \( P_H \psi = \omega \), and (iv) \( P_{local} \eta = \psi \). We note that the splitting (2.6) is not unique, and that in fact these operations can be applied in any order. We discuss our technique for efficient solution of each of the above systems in the following two sections.

2.3. **Directional preconditioner solves.** We first consider solution of the three preconditioning systems \( P_F, P_G, \) and \( P_H \) from (2.6) of the form, e.g., (x-direction)

\[
P_F \chi = b \quad \Leftrightarrow \quad \chi + \gamma J_F L_F^{-1} \partial_x (L_F \chi) = b.
\]

To this point \( L_F, L_G, \) and \( L_H \) are still unspecified. We take these to be \( n \times n \) matrices whose rows are the left eigenvectors of the respective Jacobians, giving the identities

\[
L_F J_F = \Lambda_F L_F, \quad \Lambda_F = \text{diag}(\lambda^1, \ldots, \lambda^n), \quad J_F R_F = R_F \Lambda_F,
\]

where \( R_F \equiv L_F^{-1} \) are the right eigenvectors (\( n \times n \) column matrix), and \( \lambda^k \) are the eigenvalues of \( J_F \). Through premultiplication of (2.7) by \( L_F \), we have

\[
L_F \chi + \gamma L_F J_F R_F \partial_x (L_F \chi) = L_F b \quad \Leftrightarrow \quad L_F \chi + \gamma \Lambda_F \partial_x (L_F \chi) = L_F b.
\]

Defining the vector of characteristic variables \( w = L_F \chi \), we decouple the equations

\[
w + \gamma \Lambda_F \partial_x w = L_F b \quad \Leftrightarrow \quad w^k + \gamma \lambda^k \partial_x w^k = \beta^k, \quad k = 1, \ldots, n,
\]

where \( w^k \) denotes the \( k \)th element of the characteristic vector \( w \), and \( \beta = L_F b \).
Spatial discretization of each of the characteristic variables \( w^k \) in the same manner as the original PDE (2.1) results in a tightly banded linear system of equations (tridiagonal, pentadiagonal, etc., depending on the method) to solve for the values \( w^k \).

For example the tridiagonal version due to a \( O(\Delta x^2) \) finite-difference discretization is

\[
\frac{\gamma \lambda_j}{2\Delta x} (w^k_{j+1} - w^k_{j-1}) = \beta^k_j.
\]

In the ensuing results, we use a second-order centered finite-volume approximation, with resulting systems for each \( w^k \) that is tridiagonal. Moreover, the above approach results not only in tridiagonal systems for each characteristic variable \( w^k \), but the systems are in fact block tridiagonal, where each block corresponds to only one spatial \(\{x, y, z\} \) row that is decoupled from all other rows through the domain in the same direction. Thus solution of these linear systems can be very efficient, as the computations on each row may be performed independently of one another.

Furthermore, since our initial assumption was that the stiffness of the overall system resulted from a few very fast waves, we need not construct and solve the above systems for each characteristic variable \( w^k \).

In cases where the wave speeds can be estimated, we may set a predefined cutoff to the number of waves included in the preconditioner. This reduction allows for significant savings in preconditioner computation. For those waves that we do not precondition, we approximate them as having wave speed equal to zero, i.e., solving with the approximation \( \Lambda_F = \text{diag}(\lambda^1, \ldots, \lambda^q, 0, \ldots, 0) \). Omission of the \( (n-q) \) slowest waves in this fashion amounts to a further approximation of the preconditioner to the original discretized PDE system. Writing \( P_F \) as the \( x \)-directional preconditioner based on \( q \) waves, we may consider \( \| \chi - \hat{\chi} \|_p \), where \( \chi \) solves \( P_F \chi = b \) and \( \hat{\chi} \) solves \( P_F \hat{\chi} = b \), i.e.,

\[
\chi + \gamma J_F R_F \partial_x (L_F \chi) = b, \quad \hat{\chi} + \gamma \hat{J}_F R_F \partial_x (L_F \hat{\chi}) = b,
\]

where \( \hat{J}_F = R_F \hat{\Lambda}_F L_F \). Left-multiplying by \( L_F \) and proceeding as before, we obtain

\[
\begin{align*}
\hat{w}^k + \gamma \hat{\lambda}_F \partial_x \hat{w} = L_F b, \\
\hat{w}^k + \gamma \hat{\lambda}_F \partial_x \hat{w} = (L_F b)^k, \quad k = 1, \ldots, n, \\
\hat{w}^k + \gamma \hat{\lambda}_F \partial_x \hat{w} = (L_F b)^k, \quad k = 1, \ldots, q, \\
\hat{w}^k = (L_F b)^k, \quad k = q + 1, \ldots, n.
\end{align*}
\]

Measuring the error between \( w \) and \( \hat{w} \) in the \( p \)-norm \( (1 \leq p < \infty) \), we have

\[
\|w - \hat{w}\|_p = \sum_{k=1}^{n} \|w^k - \hat{w}^k\|_p = \sum_{k=q+1}^{n} \|w^k - (L_F b)^k\|_p \\
\leq \sum_{k=q+1}^{n} \|(I + \gamma \lambda^k \partial_x (\cdot))^{-1} - I\|_p \|(L_F b)^k\|_p \\
\leq \sum_{k=q+1}^{n} \left( \frac{\|\gamma \lambda^k \partial_x (\cdot)\|_p}{1 - \|\gamma \lambda^k \partial_x (\cdot)\|_p} \right)^p \|(L_F b)^k\|_p,
\]

where we have assumed for the last inequality that \( \|\gamma \lambda^k \partial_x (\cdot)\|_p < 1 \), \( k = q + 1, \ldots, n \)
where the solution vector to correct for spatial inhomogeneity in component of the split preconditioner (2.6) comprises the local system. Lastly, since $\chi = R_F w$, we obtain the error bound

$$\| \chi - \tilde{\chi} \|_p \leq \| R_F \|_p \left[ \sum_{k=q+1}^{n} \left( \frac{\| \gamma \lambda^k |\partial_x |(\cdot)\|_p}{1 - \| \gamma \lambda^k |\partial_x |(\cdot)\|_p} \right)^p \| (L_F b)^k \|^p \right]^{1/p}.$$ 

Since the eigenvector matrices $L_F$ and $R_F$ may be renormalized as desired, and the eigenvalues are ordered so that $\lambda_i \geq \lambda_j$, for $i < j$, the dominant error from preconditioning only the $q$ fastest waves is approximately

$$\frac{|\gamma \lambda^{q+1} / \Delta x|}{1 - |\gamma \lambda^{q+1} / \Delta x|}.$$ 

Hence omission of waves with small eigenvalues compared to the dynamical time scale (i.e., $\gamma \lambda \ll 1$) will not significantly affect preconditioner accuracy.

2.4. Local nonconstant coefficient correction solve. The remaining component of the split preconditioner (2.6) comprises the local system $P_{local} u = v$,

$$[I + \gamma \partial_x (J_F R_F) L_F + \gamma \partial_y (J_G R_G) L_G + \gamma \partial_z (J_H R_H) L_H] u = v$$

$$\Leftrightarrow [I + \gamma \partial_x (R_F \Lambda_F) L_F + \gamma \partial_y (R_G \Lambda_G) L_G + \gamma \partial_z (R_H \Lambda_H) L_H] u = v.$$ 

We note that for spatially homogeneous Jacobians, $\partial_x (R_F \Lambda_F) = 0$ (similarly for $y$ and $z$), this system reduces to $u = v$. Hence this component may be optionally included to correct for spatial inhomogeneity in $J_F$, $J_G$, and $J_H$. In keeping with the previous discretization approaches, we approximate this system as, e.g.,

$$\gamma \partial_x (R_F \Lambda_F) L_F \approx \frac{2}{\Delta x^2} (R_{F,i+1} \Lambda_{F,i+1} - R_{F,i-1} \Lambda_{F,i-1}) L_{F,i}.$$ 

These solves are spatially decoupled (with respect to $u$), resulting in a block-diagonal matrix whose solution requires only $n \times n$ dense linear solves at each spatial location.

3. Application to compressible gas dynamics and MHD. The preconditioning method developed in the previous section is applicable to any system of multidimensional hyperbolic equations. In this section we apply it to the compressible gas dynamics and magnetohydrodynamics systems of equations. We include a simple 1D MHD example to illustrate the effect of the preconditioning method on the spectrum of the preconditioned Jacobian.

3.1. Equations of compressible gas dynamics and MHD. Single fluid magnetohydrodynamics is a mathematical description of a plasma which may be derived from kinetic equations assuming quasineutrality and no distinction between ions and electrons. The equations of gas dynamics form a subset of the MHD equations and may be simply obtained by setting the magnetic field, $\mathbf{B}$, to zero. The MHD equations couple the equations of compressible hydrodynamics with the low-frequency Maxwell’s equations, and may be written in conservation form using rational units,

$$\partial_t U + \nabla \cdot \mathbf{F}(U) = S(U, \nabla \cdot \mathbf{B}) + S(U),$$

where the solution vector $U \equiv U(x, t)$ is given by $U = \{ \rho, \rho u, \rho \mathbf{u}, e \}^T$, and the hyperbolic flux vector $\mathbf{F}(U)$ is given by

$$\mathbf{F}(U) = \left\{ \rho \mathbf{u}, \rho \mathbf{uu} + (p + \frac{\mathbf{B} \cdot \mathbf{B}}{2}) \mathbf{I} - \mathbf{BB}, \mathbf{uB} - \mathbf{Bu}, (e + p + \frac{\mathbf{B} \cdot \mathbf{B}}{2}) \mathbf{u} - \mathbf{B}(\mathbf{Bu}) \right\}^T.$$
In these equations $\rho$ is the density, $\mathbf{u}$ is the velocity, $\mathbf{B}$ is the magnetic field, $p$ is the pressure, and $\epsilon$ is the total energy per unit volume of the plasma. The system is closed by the equation of state, $e = \frac{\rho u^2}{\rho} + \frac{\rho}{2} \mathbf{B} \cdot \mathbf{B}$, with the ratio of specific heats denoted by $\Gamma$ and taken to be $5/3$ throughout this work. The left-hand side of (3.1) comprises the ideal MHD portion of the single-fluid MHD equations, and for completeness we have included a source term $S(U)$ above which allows diffusive terms (usually viscous, resistive, and heat conduction terms; see Reynolds, Samtaney, and Woodward [36] for these). As mentioned above, setting $\mathbf{B} = 0$ results in the compressible Navier–Stokes equations if viscosity and heat conductivity are included, or the compressible Euler equations describing the motion of an ideal compressible gas if diffusion terms are absent. The source term is also a placeholder for reaction terms which may occur, for example, in gas combustion systems.

Following the original work of Godunov [14] and subsequent work by Falle, Komissarov, and Joarder [12] and Powell et al. [34], we have included a nonconservative source term $S(U, \nabla \cdot \mathbf{B}) = -\nabla \cdot \mathbf{B} \{0, \mathbf{B}, \mathbf{u}, \mathbf{B} \cdot \mathbf{u}\}^T$ to symmetrize the ideal MHD system. Inclusion of this source term leads to a nonsingular Jacobian, $J_F$, for the ideal MHD system, and as long as $\nabla \cdot \mathbf{B}$ remains negligible, inclusion of this term will not affect conservation or accuracy of the PDE system. Finally, for the MHD system, a consequence of Faraday’s law is that an initially divergence-free magnetic field must lead to a divergence-free magnetic field for all times, which corresponds to the lack of observations of magnetic monopoles in nature. This solenoidal property is expressed as $\nabla \cdot \mathbf{B} = 0$. The effect of implicit solvers and preconditioning on this property is discussed in the appendix.

For both MHD and compressible gas dynamics, nonlinearities typically occur on the left-hand side of (3.1), i.e., in the hyperbolic part of the system of equations. Examples of physical systems where the stiffness results from the large separation of scales or wave speeds present in the hyperbolic portion of the equations are (a) tokamak fusion MHD in which the dynamical time scales of interest are $10–100$ times slower than the fastest time scales present, and (b) low Mach number combustion in which the advective time scales of interest are nearly two orders of magnitude smaller than the acoustic time scales (although the fastest time scales in combustion may be due to chemical reactions which are not part of the hyperbolic part of the equations and not considered here). Due to this stiffness, numerical simulations of such systems can benefit from the implicit approach presented in this work.

3.2. Preconditioner for the gas dynamics/ideal MHD system. The development of the preconditioner for the gas dynamics or the ideal MHD system proceeds in almost exactly the same fashion as described in section 2.2. For ideal MHD, we base our work on the 8-wave MHD system introduced by Powell et al. [34]. The row matrix (now an $8 \times 8$ matrix) of left eigenvectors for this system is derived as $L_F = Q^{-1} L_F Q$, where $Q$ corresponds to the Jacobian of the conserved variable to primitive variable transformation, $(\rho, \rho \mathbf{u}, \mathbf{B}, e) \rightarrow (\rho, \mathbf{u}, \mathbf{B}, p)$, and $L_F$ corresponds to the matrix of left eigenvectors of $J_F$. This 8-wave formulation is equivalent to the ideal MHD system (3.1), so long as the solenoidal constraint $\nabla \cdot \mathbf{B} = 0$ is satisfied exactly. Hence, for such constraint-preserving states we have the identities

$$J_F = Q^{-1} \tilde{J}_F Q, \quad L_F = Q^{-1} \tilde{L}_F Q, \quad R_F = Q^{-1} \tilde{R}_F Q,$$

where $R_F \equiv L_F^{-1}$, $L_F J_F = \Lambda_F L_F$, $J_F R_F = R_F \Lambda_F$.

The eight eigenvalues for the ideal MHD system are $\Lambda_F = \text{diag}(u_x \pm c_f, u_x \pm c_a, u_x \pm c_s, u_x, u_x)$, where $u_x$ is the $x$-velocity, and $c_f, c_a, c_s$ are, respectively, the fast magne-
tosonic, Alfvén, and slow magnetosonic speeds. Typically in tokamak fusion MHD, \( c_s \ll c_a < c_f \). Of the two identical eigenvalues \( u_x \), one corresponds to the entropy wave while the other corresponds to advection of \( \nabla \cdot \mathbf{B} \) in the 8-wave formulation.

In 3D gas dynamics, the left and right eigenvectors are 5 \( \times \) 5 matrices and can be easily generalized from a 1D system presented in Chapter 3 of [28]. The eigenvalues of inviscid gas dynamics are \( \Lambda_p = \text{diag}(u_x \pm c, u_x, u_x, u_x) \), where \( c \) is the sound speed given by \( c^2 = \Gamma p/\rho \).

### 3.3. An illustrative example

Before applying the preconditioner to multidimensional nonlinear examples, we illustrate its benefit on a simple example. We consider the 1D linearized ideal MHD system,

\[
\partial_t U + J^0_F \partial_x U = 0,
\]

where \( J^0_F = \frac{dF(U)}{dU} |_{U=U^0} \), under the following background state: \( \{\rho_0 = 1.0, \rho u_0 = 0, B_x = 0.1 \cos(\alpha), B_y = 0.1 \sin(\alpha), B_z = 1, p_0 = 0.01\} \), with \( \alpha = 60^\circ \). This corresponds to a low-\( \beta \) tokamak plasma with \( B_z \) playing the role of the toroidal magnetic field, and \( B_x \) and \( B_y \) corresponding to components of the poloidal magnetic field. The wave speeds for this state are \( c_s = 0.006, c_a = 0.05 \), and \( c_f = 1.013 \) (i.e., \( c_f \approx 10c_a \approx 100c_s \), conditions similar to those encountered in tokamak fusion plasmas). We discretize (3.2) uniformly over the domain \( x \in [0, L] \) with \( N \) finite volume cells and mesh spacing \( \Delta x = L/N \). We further assume periodic boundary conditions, a fourth-order finite-difference discretization of the spatial derivative, and a \( \theta \)-scheme time discretization. The spatial derivative is approximated as \( (\partial_x U)_i \approx a(U_{i+1} - U_{i-1}) + b(U_{i+2} - U_{i-2}) \), where in the fourth-order method \( a = (1.5\Delta x)^{-1} \) and \( b = (-12\Delta x)^{-1} \). We then solve the linear system \( JU^{n+1} = g(U^n) \) at every time step, where \( g(U) = U - (1-\theta)\Delta t J_F \left[ a(U_{i+1} - U_{i-1}) + b(U_{i+2} - U_{i-2}) \right] \).

\[
J = \begin{bmatrix}
I & K & M & 0 & \cdots & 0 & -M & -K \\
-K & I & K & M & 0 & \cdots & 0 & -M \\
-M & -K & I & K & M & 0 & \cdots & 0 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & 0 & \cdots & 0 & -M & -K & I & K \\
M & 0 & \cdots & 0 & -M & -K & I & K \\
K & M & 0 & \cdots & 0 & -M & -K & I \\
\end{bmatrix},
\]

\( K = a \theta \Delta t J_F \), and \( M = b \theta \Delta t J_F \). The preconditioner is defined by setting all but the \( q \)-stiffest eigenvalues to zero; we define \( \Lambda = \text{diag}\{\lambda_1, \ldots, \lambda_q, 0, \ldots, 0\} \) and then solve only for the corresponding characteristic variables. This results in a preconditioner matrix that is identical to (3.3), albeit with the subblocks \( K \) and \( M \) replaced with the approximate blocks \( \tilde{K} = a \theta \Delta t R \Lambda \) and \( \tilde{M} = b \theta \Delta t R \Lambda \), respectively.

We choose the time step \( \Delta t \) to be 500 times the explicit time step constrained by the usual CFL condition and use \( \theta = 0.5 \). The domain length and discretization size are \( L = 2 \) and \( N = 512 \), respectively. For this example, the real parts of the eigenvalues of \( J \) are all 1, and its condition number is approximately 343. In Figure 1 we plot the eigenvalues of the preconditioned operator, \( JP^{-1} \), where \( P \) uses \( q = 0 \) (no preconditioning), \( q = 2 \) (fast magnetosonic), \( q = 4 \) (Alfvén and fast magnetosonic), and \( q = 8 \) (all waves). For these values of \( q \), the condition number decreases from 343 to 16 to 2 and finally to 1, leading to the observation that for a linear system of hyperbolic conservation laws, the preconditioner is exactly equal to the Jacobian.
4. Computational results. Unless otherwise noted, all tests were performed using identical solver parameters: a relative nonlinear residual tolerance $\epsilon = 10^{-7}$, a maximum of 20 Newton iterations, at most 200 GMRES iterations per Newton step, an inexact Newton tolerance of 0.01 $\|f(U)\|_{\text{WRMS}}$, and a time-centered implicit discretization (i.e., $q = 1, \beta_0 = \beta_1 = 0.5$ from (2.2)). All preconditioned tests used the full 8-wave formulation without the splitting correction $P_{\text{local}}$ (labeled FW Prec), and were compared against an implicit but unpreconditioned solver (labeled No Prec). Of these, all nonscaling studies were performed in serial, on a dual 3.0 GHz Linux workstation, while all parallel scaling tests were performed on the Thunder Linux cluster at Lawrence Livermore National Laboratory, consisting of 1024 1.4 GHz Itanium-2 nodes (4 processors/node). The computational tests are performed using an implicit resistive MHD code described in [36], in which we have added the preconditioning approach. We examine this approach on a suite of stiff test problems from MHD and gas dynamics. We first examine a MHD linear wave propagation problem on which we expect the preconditioner to solve to be nearly exact. This problem is then rotated to exercise the directionally-split nature of the preconditioner. The third test is a 2D Rayleigh–Taylor instability in compressible gas dynamics. The fourth problem is the Kelvin–Helmholtz instability, a stringent nonlinear resistive MHD test that has been used to assess the efficacy of preconditioners for implicit MHD solvers [7]. We further provide parallel weak scaling results, as well as some investigation of various modifications to this preconditioning approach that may offer increased computational efficiency.

4.1. Linear wave propagation tests. We begin with the “best case scenario” for this preconditioner, consisting of an ideal MHD linear wave advection test problem. It involves application of a small amplitude perturbation to an initially constant equilibrium, where the perturbation consists of an eigenfunction corresponding to the slow magnetosonic wave [36]. Hence implicit integration of this problem consists of advection of this stiff, slow wave across the domain.

We set the computational domain to $[0, 2] \times [0, 2]$, with periodic boundary conditions in both the $x$- and $y$-directions, and initialize the equilibrium state $\hat{U} = (\rho, \rho u, B, e)^T$, where $\rho = 1, u = 0, B = (\cos \alpha \cos \theta, \sin \alpha \sin \theta, 0)^T$, and $e = 0.1$. Here, $\theta = \tan^{-1} \frac{k_y}{k_x}$, $\alpha = -44.5^\circ$, and the ratio $\frac{k_y}{k_x}$ gives the direction of wave propagation. We project these to characteristic variables via $W = L\hat{U}$, where $L$ is the left
eigenvector matrix of the linearized MHD system, and then perturb the sixth-fastest characteristic, \( w_6 = w_6^0 + 10^{-5} \cos(\pi k_x x + \pi k_y y) \). The initial condition is then set as \( U^0 = R W \), where \( R \) is the right eigenvector matrix. All runs are taken to a final time of \( T_f = 10 \). We note that even though the simulation code is fully nonlinear, the nonlinear effects are second order in the perturbation amplitude, and thus simulations still operate within the linearized MHD regime.

We examine two scenarios: in the first the wave propagates parallel to the \( x \)-axis (i.e., \( k_x = 1 \), \( k_y = 0 \)); hence the directional splitting used in the preconditioner should not affect its accuracy. In the second the propagation is at a 45° angle oblique to the \( x \)-axis (i.e., \( k_x = k_y = 1 \)), exercising the directionally-split nature of the approach.

Results for both tests are shown in Figure 2. The total number of Krylov iterations is plotted for various time step sizes (different curves) and spatial discretizations (horizontal axis). For the \( x \)-directional test, the preconditioner is nearly exact, and hence the Krylov iterations remain nearly constant as the mesh is refined, as compared with the nonpreconditioned tests that increase rapidly. For the oblique test, the directional splitting does not appear to significantly affect the preconditioner accuracy, again resulting in nearly constant Krylov iterations with mesh refinement. Table 1 shows the CPU timings, which suggest that for small problem sizes the reduction in linear iterations does not outweigh the computational cost of preconditioning; though once the mesh is refined the preconditioning benefit becomes increasingly apparent.

### 4.2. Rayleigh–Taylor test

The Rayleigh–Taylor instability (RTI) occurs whenever fluids of different density are subjected to acceleration in a direction opposite...
that of the density gradient [10]. RTI is encountered in a variety of contexts, such as combustion, inertial-confinement fusion, supernova explosions, and geophysics. Simulations of the compressible RTI are quite common (e.g., see Table 1 of [11] for a list of compressible RTI investigations and other references therein). Furthermore, in a variety of such simulations the Mach number is typically low (O(0.1) or less). This begs for an implicit treatment of the fast acoustic time scales, as they evolve much more quickly than the advective time scales of physical interest. For this test, we augment the compressible gas dynamics equations with gravitational acceleration and viscous stress terms in the momentum equations, and the work done due to gravity and shear stress in the energy equation. Initially, the fluids are at rest under hydrostatic equilibrium. The density field is then perturbed according to the equations

\[
\rho(x, y) = \frac{1}{2} (\rho_1 + \rho_2) + \frac{1}{2} (\rho_2 - \rho_1) \tanh[\psi(x, y)],
\]

\[
\psi(x, y) = y - \frac{1}{2} \sum_{k=1}^{N} \exp[-\sigma(k - 2)^2] [a_1 \cos(kx) + a_2 \sin(kx)].
\]

We set \(\rho_1 = 1\) and \(\rho_2 = 2\) (i.e., Atwood ratio is one-third), where \(a_1, a_2\) are random amplitudes in the interval \([-0.5, 0.5]\) and \(\sigma = 0.01\). The domain of investigation is \([-\pi/2, \pi/2] \times [-6, 6]\), with boundary conditions as outflow in \(y\) and periodic in \(x\). We set the Froude number to 10 and Reynolds number to \(10^5\). Snapshots of the density field at \(t = \{0, 400\}\) are shown in Figure 3 for a \(256^2\) mesh simulation computed with a time step of \(\Delta t = 0.25\). In this simulation the peak Mach number varied from 0 at \(t = 0\) to approximately 0.011 at \(t = 400\). In Figure 4 we present solver results, in which the preconditioned approach takes considerably fewer linear iterations for all time step sizes and spatial discretizations used. The CPU data in Table 2 demonstrates that for very small time steps and coarse meshes, the cost of preconditioning outweighs the runtime benefit resulting from the reduced iterations. For larger time steps or better-refined meshes, the preconditioned simulation again significantly outperforms the unpreconditioned solver.

### 4.3. Kelvin–Helmholtz tests

We now investigate a more strenuous MHD problem, the Kelvin–Helmholtz test [10, 26]. This instability exhibits a large growth rate, and the dynamics quickly become nonlinear with tightly coupled, highly inhomogeneous fields. We therefore consider this to be a “worst-case” test problem...
Mesh size & Krylov iterations (2D Rayleigh–Taylor) & Fig. 4. Krylov iterations for the 2D Rayleigh–Taylor tests.

Table 2

<table>
<thead>
<tr>
<th>Mesh size</th>
<th>Δt</th>
<th>NP time</th>
<th>FW time</th>
</tr>
</thead>
<tbody>
<tr>
<td>64²</td>
<td>0.1</td>
<td>119.3</td>
<td>134.7</td>
</tr>
<tr>
<td>64²</td>
<td>0.2</td>
<td>139.8</td>
<td>106.8</td>
</tr>
<tr>
<td>128²</td>
<td>0.1</td>
<td>1129.5</td>
<td>1365.5</td>
</tr>
<tr>
<td>128²</td>
<td>0.2</td>
<td>1157.8</td>
<td>678.4</td>
</tr>
<tr>
<td>128²</td>
<td>0.4</td>
<td>1502.7</td>
<td>581.5</td>
</tr>
<tr>
<td>256²</td>
<td>0.1</td>
<td>9791.4</td>
<td>7594.5</td>
</tr>
<tr>
<td>256²</td>
<td>0.2</td>
<td>13866.3</td>
<td>3773.3</td>
</tr>
<tr>
<td>256²</td>
<td>0.4</td>
<td>18995.9</td>
<td>3525.9</td>
</tr>
</tbody>
</table>

(2D Rayleigh–Taylor test runtimes. We compare solvers with and without preconditioning (FW, NP) on varying time step sizes (Δt). Comparable explicit simulation times were 225.39, 3168.8, and 25494.0 seconds for the 64², 128², and 256² meshes, respectively.

Snapshots of the x and z components of the (initially homogeneous) magnetic field at t = 2 for a 256² mesh simulation computed with a time step of Δt = 0.0025. Throughout this simulation, the number of nonlinear iterations ranged from 1 to 3, with the associated preconditioned Krylov iteration counts in the range of 6–13 per time step. Solver results for these tests are shown in Figure 6 and Table 3. The 3D nonpreconditioned solver could not converge within the allowed
Fig. 5. Snapshots of $B_x$ (left) and $B_z$ (right) in the 2D Kelvin–Helmholtz test at $t = 2$.

Fig. 6. Krylov iterations for the 2D (left) and 3D (right) Kelvin–Helmholtz tests. The 3D nonpreconditioned solver could not converge within the allowed iterations for many of the $\Delta t = 0.01$ runs, or for nearly all of the $\Delta t = 0.02$ runs, so those portions of the curves are omitted.

iterations at $\Delta t = 0.01$ for larger mesh sizes and could not converge at $\Delta t = 0.02$ for any but the coarsest mesh. For all time step sizes and all spatial discretizations used, the preconditioner results in significantly fewer linear iterations, with the disparity growing as the mesh is refined. However, for small time steps and coarse spatial meshes we again see that the cost of preconditioning outweighs the runtime benefit due to the reduced iteration count. At larger time step sizes and for more refined meshes the preconditioned simulation is the clear winner.

Table 3

<table>
<thead>
<tr>
<th>Mesh</th>
<th>$\Delta t$</th>
<th>NP time</th>
<th>FW time</th>
</tr>
</thead>
<tbody>
<tr>
<td>64$^2$</td>
<td>0.0025</td>
<td>267.9</td>
<td>359.5</td>
</tr>
<tr>
<td>64$^2$</td>
<td>0.005</td>
<td>220.4</td>
<td>204.1</td>
</tr>
<tr>
<td>64$^2$</td>
<td>0.01</td>
<td>226.9</td>
<td>117.2</td>
</tr>
<tr>
<td>128$^2$</td>
<td>0.0025</td>
<td>1727.8</td>
<td>1724.4</td>
</tr>
<tr>
<td>128$^2$</td>
<td>0.005</td>
<td>1742.9</td>
<td>980.8</td>
</tr>
<tr>
<td>128$^2$</td>
<td>0.01</td>
<td>2179.6</td>
<td>731.7</td>
</tr>
<tr>
<td>256$^2$</td>
<td>0.0025</td>
<td>16277.3</td>
<td>9952.3</td>
</tr>
<tr>
<td>256$^2$</td>
<td>0.005</td>
<td>19822.8</td>
<td>6174.5</td>
</tr>
<tr>
<td>256$^2$</td>
<td>0.01</td>
<td>26090.8</td>
<td>8052.4</td>
</tr>
</tbody>
</table>

Kelvin–Helmholtz test runtimes: left is 2D, right is 3D. We compare runs with and without preconditioning (FW, NP) at various time step sizes ($\Delta t$); nonconvergent runs are denoted “—–.”
4.4. Weak scaling tests. We also consider the scalability of this approach as both the problem size and number of processors are increased (i.e., weak scaling). Here, all problem specifications match those in earlier tests, except that the final time is reduced to $T_f = 0.5$. We plot total wall-clock time and total linear iterations required as we increase the computational mesh proportionately with the number of processors. Perfect weak scaling of these quantities would be represented by a horizontal line.

Results for the obliquely propagating linear wave test problem and both 2D and 3D Kelvin–Helmholtz test problems are presented in Figure 7. On the linear wave tests we note that the requisite linear iterations for convergence scales almost perfectly ($P^{0.005}$) for all time step factors. While the linear iterations scale ideally, we note that CPU scaling depends on both a constant number of linear iterations and the scalability of the inner parallel periodic tridiagonal solver. The plot therefore investigates the scalability of this inner linear solver, showing modest increases in CPU time as the problem size is increased, scaling as $P^{0.15}$ for up to $P = 576$ for these 2D tests.

For the more strenuous Kelvin–Helmholtz tests we see how the preconditioner accuracy comes into play for moderate to large problem sizes. Here the 2D tests exhibit a slow growth in linear iterations, scaling ($P^{0.24}$) up through $P = 256$, while the 3D tests show good scalability in linear iterations ($P^{0.13}$) for all but the largest $\Delta t$. Such behavior could be due to either an increase in splitting inaccuracy or an increase in the second-order (viscous and resistive) terms that are left unpreconditioned with this formulation. These increases in linear iterations are borne out in the CPU scalings, where we notice a steady increase in required CPU time, exhibiting 2D scaling of approximately $P^{0.28}$ (up to $P = 256$) and 3D of $P^{0.24}$ (up to $P = 1000$). Even with those increases, however, we point out that in three dimensions the required CPU time only goes up by about a factor of 6 in scaling up from 1 to 1000 processors.

4.5. Preconditioner optimizations. In the preconditioner formulation from section 2, a number of customizations are readily apparent. We examine three such changes here: preconditioning only a subset of the MHD waves, including the local correction solve $P_{\text{local}}$, and freezing the eigendecomposition used in the preconditioner to the initial condition. Each may provide increased efficiency over the approach used in the previous results. For problems whose stiffness results from only the fastest MHD waves, preconditioning the slow waves may not contribute to the accuracy of the overall approach and may therefore be omitted. For problems that are highly spatially inhomogeneous, the correction matrix $P_{\text{local}}$ may significantly improve the accuracy of the approach. Lastly, for problems that do not deviate far from their initial conditions, the eigendecompositions may remain nearly unchanged throughout a simulation, and therefore the initial decomposition could be reused.

We investigate such options on both the oblique linear wave test and the 2D Kelvin–Helmholtz test problems in Tables 4 and 5. Here, the preconditioners include the standard 8-wave approach above with the correction solve (Prec = 8), the 8-wave approach without the correction solve (Prec = 8 NC), the 8-wave approach without correction that is frozen on the initial condition (Prec = 8 NC-f), the noncorrected 4-wave formulation (Prec = 4 NC), and the noncorrected 2-wave formulation (Prec = 2 NC). We tested each of these approaches on a variety of spatial discretizations and time step factors ($C = \Delta t / \Delta t_{\text{CFL}}$). All simulations were performed for 50 time steps, and we present values of both the total linear iterations (Krylov) and the total runtime (CPU). As demonstrated in these results, such optimizations can indeed provide benefits over the full preconditioner formulation; however, it is difficult to a
priori determine the best formulation for a given problem, as the optimal approach appears to be problem dependent. However, in most circumstances it appears that the splitting correction solve $P_{\text{local}}$ may both slow convergence of the Krylov iteration and contribute to unnecessary overhead in the preconditioner evaluation.

5. Conclusions. We have introduced a preconditioner designed to alleviate stiffness induced by fast hyperbolic waves for problems discretized on structured grids. Through splitting the implicit operator into its directional components along each axial direction and then using the characteristic decomposition of the linearized operator appropriately, we approximate the solution to a multivariable 3D hyperbolic operator with a sequence of single-valued banded linear solves. Moreover, through using the characteristic decomposition we may choose to precondition any number of components, namely only those contributing the greatest stiffness to the problem.

Such an approach is not limited to uniform grid calculations. Block-structured
Table 4
Comparisons of average linear iterations and CPU time per time step for various “optimizations” on the oblique linear advection test. Preconditioners solve for varying numbers of waves, ignore \( P_{\text{local}} \) (NC), and/or freeze the initial eigendecomposition (f).

<table>
<thead>
<tr>
<th>Mesh</th>
<th>Prec</th>
<th>( C = 50 ) Krylov</th>
<th>( C = 50 ) CPU</th>
<th>( C = 100 ) Krylov</th>
<th>( C = 100 ) CPU</th>
<th>( C = 250 ) Krylov</th>
<th>( C = 250 ) CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>128 x 128</td>
<td>8</td>
<td>1.0</td>
<td>0.6</td>
<td>1.0</td>
<td>0.6</td>
<td>1.0</td>
<td>0.6</td>
</tr>
<tr>
<td>128 x 128</td>
<td>8 NC</td>
<td>1.0</td>
<td>0.4</td>
<td>1.0</td>
<td>0.4</td>
<td>1.0</td>
<td>0.4</td>
</tr>
<tr>
<td>128 x 128</td>
<td>8 NC-f</td>
<td>1.0</td>
<td>0.4</td>
<td>2.7</td>
<td>0.7</td>
<td>1.2</td>
<td>0.4</td>
</tr>
<tr>
<td>128 x 128</td>
<td>2 NC</td>
<td>1.0</td>
<td>0.5</td>
<td>2.7</td>
<td>0.6</td>
<td>1.2</td>
<td>0.4</td>
</tr>
<tr>
<td>256 x 256</td>
<td>8</td>
<td>1.0</td>
<td>3.0</td>
<td>1.0</td>
<td>2.9</td>
<td>8.8</td>
<td>8.9</td>
</tr>
<tr>
<td>256 x 256</td>
<td>8 NC</td>
<td>1.0</td>
<td>2.1</td>
<td>1.0</td>
<td>2.2</td>
<td>3.8</td>
<td>3.7</td>
</tr>
<tr>
<td>256 x 256</td>
<td>8 NC-f</td>
<td>1.0</td>
<td>2.1</td>
<td>1.0</td>
<td>2.1</td>
<td>3.8</td>
<td>3.7</td>
</tr>
<tr>
<td>256 x 256</td>
<td>4 NC</td>
<td>1.1</td>
<td>2.0</td>
<td>1.0</td>
<td>2.0</td>
<td>3.6</td>
<td>3.5</td>
</tr>
<tr>
<td>256 x 256</td>
<td>2 NC</td>
<td>1.1</td>
<td>2.1</td>
<td>1.0</td>
<td>2.1</td>
<td>3.6</td>
<td>3.7</td>
</tr>
<tr>
<td>512 x 512</td>
<td>8</td>
<td>1.0</td>
<td>13.6</td>
<td>1.0</td>
<td>13.0</td>
<td>7.7</td>
<td>34.9</td>
</tr>
<tr>
<td>512 x 512</td>
<td>8 NC</td>
<td>1.0</td>
<td>10.1</td>
<td>1.0</td>
<td>9.7</td>
<td>7.8</td>
<td>27.9</td>
</tr>
<tr>
<td>512 x 512</td>
<td>8 NC-f</td>
<td>1.0</td>
<td>10.1</td>
<td>1.0</td>
<td>9.7</td>
<td>7.8</td>
<td>28.0</td>
</tr>
<tr>
<td>512 x 512</td>
<td>4 NC</td>
<td>1.4</td>
<td>11.4</td>
<td>3.2</td>
<td>14.2</td>
<td>7.8</td>
<td>25.5</td>
</tr>
<tr>
<td>512 x 512</td>
<td>2 NC</td>
<td>1.4</td>
<td>11.7</td>
<td>3.2</td>
<td>16.4</td>
<td>7.8</td>
<td>30.2</td>
</tr>
</tbody>
</table>

Table 5
Comparisons of average linear iterations and CPU time per time step for various “optimizations” on the 2D Kelvin–Helmholtz test problem. Preconditioners solve for varying numbers of waves, ignore \( P_{\text{local}} \) (NC), and/or freeze the initial eigendecomposition (f).

<table>
<thead>
<tr>
<th>Mesh</th>
<th>Prec</th>
<th>( C = 25 ) Krylov</th>
<th>( C = 25 ) CPU</th>
<th>( C = 50 ) Krylov</th>
<th>( C = 50 ) CPU</th>
<th>( C = 100 ) Krylov</th>
<th>( C = 100 ) CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>128 x 64</td>
<td>8</td>
<td>16.2</td>
<td>2.0</td>
<td>28.6</td>
<td>3.0</td>
<td>101.2</td>
<td>10.0</td>
</tr>
<tr>
<td>128 x 64</td>
<td>8 NC</td>
<td>14.7</td>
<td>1.3</td>
<td>27.3</td>
<td>2.0</td>
<td>65.9</td>
<td>4.8</td>
</tr>
<tr>
<td>128 x 64</td>
<td>8 NC-f</td>
<td>13.9</td>
<td>1.1</td>
<td>21.0</td>
<td>1.6</td>
<td>42.6</td>
<td>2.8</td>
</tr>
<tr>
<td>128 x 64</td>
<td>4 NC</td>
<td>14.7</td>
<td>1.1</td>
<td>27.2</td>
<td>1.8</td>
<td>66.2</td>
<td>4.2</td>
</tr>
<tr>
<td>128 x 64</td>
<td>2 NC</td>
<td>14.8</td>
<td>1.0</td>
<td>26.3</td>
<td>1.6</td>
<td>65.1</td>
<td>3.7</td>
</tr>
<tr>
<td>256 x 128</td>
<td>8</td>
<td>13.7</td>
<td>9.0</td>
<td>24.9</td>
<td>11.9</td>
<td>83.1</td>
<td>34.7</td>
</tr>
<tr>
<td>256 x 128</td>
<td>8 NC</td>
<td>12.4</td>
<td>5.9</td>
<td>22.6</td>
<td>7.3</td>
<td>56.2</td>
<td>17.0</td>
</tr>
<tr>
<td>256 x 128</td>
<td>8 NC-f</td>
<td>11.8</td>
<td>5.3</td>
<td>17.9</td>
<td>5.9</td>
<td>29.9</td>
<td>8.7</td>
</tr>
<tr>
<td>256 x 128</td>
<td>4 NC</td>
<td>12.4</td>
<td>4.3</td>
<td>22.5</td>
<td>6.5</td>
<td>56.8</td>
<td>14.7</td>
</tr>
<tr>
<td>256 x 128</td>
<td>2 NC</td>
<td>11.4</td>
<td>3.4</td>
<td>21.1</td>
<td>5.5</td>
<td>54.5</td>
<td>13.0</td>
</tr>
<tr>
<td>512 x 256</td>
<td>8</td>
<td>10.6</td>
<td>29.6</td>
<td>21.7</td>
<td>51.4</td>
<td>74.9</td>
<td>147.2</td>
</tr>
<tr>
<td>512 x 256</td>
<td>8 NC</td>
<td>9.0</td>
<td>18.9</td>
<td>18.8</td>
<td>33.5</td>
<td>48.1</td>
<td>84.6</td>
</tr>
<tr>
<td>512 x 256</td>
<td>8 NC-f</td>
<td>10.0</td>
<td>19.3</td>
<td>14.4</td>
<td>25.8</td>
<td>26.7</td>
<td>44.8</td>
</tr>
<tr>
<td>512 x 256</td>
<td>4 NC</td>
<td>9.0</td>
<td>16.0</td>
<td>18.1</td>
<td>27.2</td>
<td>49.9</td>
<td>69.1</td>
</tr>
<tr>
<td>512 x 256</td>
<td>2 NC</td>
<td>9.4</td>
<td>14.9</td>
<td>17.3</td>
<td>23.9</td>
<td>45.8</td>
<td>55.4</td>
</tr>
</tbody>
</table>

Adaptive mesh refinement methods discretize PDEs with a hierarchy of regular grids to adaptively refine around regions of interest [3, 4, 38]. For such problems, fast adaptive composite linear solvers are often used, which employ fast uniform-grid algorithms on each level iteratively to achieve the solution on the hierarchical composite mesh, and have proven beneficial in radiation diffusion and reduced MHD applications [30, 32, 33]. This preconditioning approach could easily be utilized in such methods.

As a result of the splittings on which this preconditioner is based, its accuracy may be limited by the time step size, the strength of propagation in directions oblique to the coordinate axes, the spatial inhomogeneity of the fluxes that give rise to the local correction matrix \( P_{\text{local}} \), and the number of waves that are treated with the preconditioner. However, even with such approximations, we have demonstrated its
effectiveness on a variety of problems that have been designed to exercise each of these error-inducing terms. All of these tests have one similarity in common—they are numerically stiff, having dynamical time scales that occur orders of magnitude more slowly than the fastest wave effects. In fact these are precisely the type of problems that require implicit methods; otherwise standard explicit-time approaches prove more efficient. We note that in a previous paper, we have shown that nonpre-conditioned implicit methods can result in tremendous efficiency improvements over explicit methods on stiff MHD problems [36]; through this preconditioning approach we have further improved the efficiency of such implicit simulations.

We have also investigated the weak scaling performance of the preconditioned implicit solver on a number of tests. In each case, the preconditioned solver scaled well as the problem sizes (and processor counts) were increased, for some problems even demonstrating nearly ideal weak scaling in linear iterations for moderate processor counts ($p \leq 256$ in 2D and $p \leq 1728$ in 3D). In those tests where the weak CPU scaling deteriorated, the number of linear iterations required for convergence did not increase dramatically, indicating that improvements could still be made in the parallel implementation of the preconditioning approach (the parallel periodic tridiagonal solver), extending its effectiveness to increasingly larger processor counts. We intend to investigate this issue, as well as approaches for reducing the splitting error, in future work.

**Appendix. Solenoidal preservation in implicit MHD solutions.** In this section we show that the use of fully implicit methods based on matrix-free inexact Newton–Krylov algorithms retains the solenoidal condition on the magnetic field for MHD calculations. To demonstrate this preservation, we first summarize the structure of these algorithms before moving on to the theoretical result.

In solving the nonlinear root-finding problem $f(U^{n}) = 0$, given in (2.2), inexact Newton methods generate a sequence of iterates $\{U^{i}\}$ that converge to the time-evolved solution $U^{n}$ as follows [24]. Given an initial guess $U_{0}$ (typically chosen as some convex combination of previous states, $U_{0} = \sum \alpha_{k}U^{n-k}$, $\sum \alpha_{k} = 1$) and nonlinear and linear tolerances $\varepsilon$ and $\delta$, the sequence $\{U^{i}\}$ is generated through the following steps:

(i) Solve $J(U^{i})V_{i} = -f(U^{i})$ for $V_{i}$ such that $\|J(U^{i})V_{i} + f(U^{i})\| < \delta$.

(ii) Set $U^{i+1} = U^{i} + V_{i}$, and if $\|f(U^{i+1})\| < \varepsilon$, stop.

In solving the system (i), matrix-free Krylov methods [15] generate another sequence of iterates $\{S_{k}\}$ converging to $V_{i}$, where each iterate is chosen from the subspace

$$K_{k}(\tilde{J}, f) = \text{span}\{f, \tilde{J}f, \tilde{J}^{2}f, \ldots, \tilde{J}^{k}f\}$$

and where $\tilde{J}$ approximates the Jacobian via a finite difference,

$$\tilde{J}(U)V = [f(U + \sigma V) - f(U)]/\sigma = J(U)V + O(\sigma).$$

For solution algorithms of this type, we have the following result.

**Theorem A.1.** If the spatial semidiscretization of $(\nabla \cdot F(U))$ is commutative, i.e., $\partial_{xy}^{2} = \partial_{yx}^{2}$, then the above algorithm applied to (2.2) exactly preserves an initially solenoidal magnetic field, for any nonlinear and linear tolerances $\varepsilon$ and $\delta$.

**Proof.** We first define the space of all constraint-preserving discrete states,

$$\mathcal{V} = \{U = (\rho, \rho u, B, e)^{T} \in \mathbb{R}^{8N} \mid \nabla \cdot B = 0\}.$$

Here $N$ is the spatial discretization size. Due to the homogeneous linear constraint $\nabla \cdot B = 0$, $\mathcal{V}$ is a vector space, since for any $U, V \in \mathcal{V}$ and $\alpha, \beta \in \mathbb{R}$, $\nabla \cdot (\alpha U + \beta V) = \text{null matrix}$. We note that for a previous paper, we have shown that nonpre-conditioned explicit methods can result in tremendous efficiency improvements over explicit methods on stiff MHD problems [36]; through this preconditioning approach we have further improved the efficiency of such implicit simulations.
\( \alpha \nabla \cdot U + \beta \nabla \cdot V = 0 \); hence \((\alpha U + \beta V) \in \mathcal{V} \), and by the vector space structure of \(\mathbb{R}^{8N} \), the vector space structure of \(\mathcal{V} \) follows.

Let \( F_B(U) \) correspond to the \( \mathbf{B} \) flux components, \( F_B(U) = \nabla \times \mathbf{E} \), where from Ohm’s law the electric field \( \mathbf{E} = -\mathbf{u} \times \mathbf{B} + \eta \nabla \times \mathbf{B} \), so \( \nabla \cdot (\nabla \cdot F_B(U)) = \nabla \cdot (\nabla \times \mathbf{E}) \).

While this identically equals zero on the continuum level, at the discrete level

\[
\nabla \cdot (\nabla \times \mathbf{E}) = \partial_x(\partial_y E_z - \partial_z E_y) + \partial_y(\partial_z E_x - \partial_x E_z) + \partial_z(\partial_x E_y - \partial_y E_x) \\
= (\partial^2_{xy} - \partial^2_{yx}) E_z + (\partial^2_{yz} - \partial^2_{zy}) E_x + (\partial^2_{zx} - \partial^2_{xz}) E_y,
\]

which equals zero only if the spatial discretization is commutative, which holds by assumption. Hence, given any element \(U \in \mathcal{V} \), \((\nabla \cdot F(U)) \in \mathcal{V} \).

Therefore if \(U \in \mathcal{V} \), \(f(U) \in \mathcal{V} \) since \(f(U)\) is formed as a linear combination of elements in \(\mathcal{V} \). Moreover, the Krylov subspace \(K_k(\tilde{J}, f) \subset \mathcal{V} \), due to the finite-difference approximation \(\tilde{J}\) that recursively forms \(K_k(\tilde{J}, f)\) out of elements in \(\mathcal{V} \). Additionally, the updated solution is formed as the linear combination \(U^n = U_0 + \sum V_i\), where \(V_i \in K_k(\tilde{J}, f)\). Lastly, since the initial guess \(U_0 = \sum \alpha_k U^{n-k} \in \mathcal{V} \) is the starting point for the iteration, and we begin the simulation with an initially solenoidal magnetic field, the updated solution \(U^n \in \mathcal{V} \) as well.

The salient features of this argument require that (a) the spatial semidiscretization is commutative and (b) the Krylov algorithm is both matrix-free and not preconditioned.

The first of these requirements is not met when upwind spatial discretizations based on Riemann or approximate Riemann solvers are used [2, 13]. For preconditioned simulations, the Krylov subspace is formed with the preconditioned operator, i.e., \(K_k(P \tilde{J}, f)\), and hence a preconditioner may push the solution off of the constraint manifold. While we do not show that our preconditioned system satisfies the constraints, we have observed minimal deviation from the solenoidal property in our tests to date, as shown in Figure 8.

We note that a proof demonstrating similar behavior regarding the global conservation of mass, momentum, and energy (exact conservation without preconditioning for any \(\varepsilon\) and \(\delta\)) may be found in [8].

**Fig. 8.** \(\nabla \cdot \mathbf{B}\) results due to preconditioning. Oblique wave propagation on a 256^2 mesh with \(\Delta t \approx 0.15\) (left); 2D Kelvin–Helmholtz on a 512^2 mesh with \(\Delta t \approx 0.006\) (right). While preconditioning increases \(|\nabla \cdot \mathbf{B}|\) error (worst with \(P_{local}\), it remains well below simulation accuracy.

**Acknowledgments.** The authors wish to acknowledge the many insightful discussions with Luis Chacón, Michael Pernice, Bobby Philip, Steve Jardin, David Keyes, and Phillip Colella during the course of this work.
REFERENCES


