Using Krylov-Schwarz methods in an adaptive mesh refinement environment

Kai Germaschewski[1], Amitava Bhattacharjee[1], Rainer Grauer[2], David Keyes[3], and Barry Smith[4]

[1] Center for Magnetic Reconnection Studies, Institute for the Study of Earth, Oceans and Space, University of New Hampshire
[2] Lehrstuhl für Theoretische Physik I, Ruhr-Universität Bochum, Germany
[3] Department of Applied Physics and Applied Mathematics, Columbia University
[4] Mathematics and Computer Science Division, Argonne National Laboratory

Much of the previous work in AMR methods has concentrated on solving hyperbolic equations with explicit timestepping. However, for many problems, either due to their physical nature (e.g., incompressible flows) or for performance reasons (semi-implicit and implicit numerical methods), it becomes necessary to solve global equations.

This paper focuses on the application and performance of well-established preconditioned Krylov-Schwarz solvers in an AMR context, using a Krylov-Schwarz method to accelerate convergence while exploiting the hierarchical structure of AMR grids for multi-level preconditioning using the *fast adpative composite (FAC)* algorithm. We present an implementation that allows us to leverage the powerful supply of preconditioners and linear solvers from the PETSc library.

We apply this method to solve the three-dimensional Euler equations in the search for a finite-time singularity.

1 Introduction

The numerical integration of the three-dimensional Euler equations in order to follow a developing singularity is a highly demanding task which can be achieved only by employing a combination of sophisticated numerical methods.

In the following we describe three key ingredients which have been developed for the *Magnetic Reconnection Code* (MRC), which, while being designed with plasma flows in mind, is applicable to fluid problems as well:

- The integration scheme is a semi-implicit second-order in time and space combined Godunov/projection method.

- Adaptive mesh refinement is used to follow small-scale structures with appropriate high resolution while smooth parts of the domain are treated at lower resolutions. At the same time, parallelization of the algorithms to work in a massively parallel environment is handled at this level.
- Solving the elliptic equations needed to calculate the pressure and maintain a divergence free velocity field is achieved though embracing and extending the PETSc library's Krylov-Schwarz solvers to a multi–level domain through McCormick's *fast adaptive composite (FAC)* algorithm. It should be noted that solving elliptic problems is also a prerequisite for many applications in plasma physics (divergence cleaning of the magnetic field and implicit treatment of the electron inertia in Hall-MHD).

2 Numerical procedure

2.1 Integration scheme

Looking at the incompressible Euler equations as the limiting case of compressible gas dynamics with compressibility κ going to zero reveals some insight into the challenges of numerical treatment.

Most notably, the pressure is an independent variable (related through an equation of state to the internal energy) in compressible gas dynamics, whereas it is just a function of the velocity field in the incompressible Euler equations.

This can easily be seen by taking the curl of the incompressible Euler equations, which makes the pressure term drop out completely:

$$\begin{aligned} \nabla \times (\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} - \nabla p = 0) \quad , \quad \nabla \cdot \mathbf{u} = 0 \\ \Longrightarrow \quad \partial_t \omega + \mathbf{u} \cdot \nabla \omega + \omega \cdot \nabla \mathbf{u} = 0 \end{aligned} \tag{1}$$

with the vorticity $\omega = \nabla \times \mathbf{u}$.

On the other hand, taking the divergence of this equation, we find how to compute the pressure from the velocity field $\mathbf{u}$:

$$\begin{aligned} \nabla \cdot (\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} - \nabla p = 0) \quad , \quad \nabla \cdot \mathbf{u} = 0 \\ \Longrightarrow \quad \nabla \cdot (\mathbf{u} \cdot \nabla \mathbf{u}) = -\nabla^2 p \end{aligned} \tag{2}$$

The speed of sound in a compressible gas is easily determined by considering adiabatic compression and expansion and found to be $c_s = 1/\sqrt{\rho\kappa}$, with density ρ and adiabatic compressibility κ.

Explicit numerical integration of the compressible gas equations needs to resolve sound waves and is thus constrained by a CFL time-step limitation, given by (possibly modified by a constant factor)

$$c_s \Delta t / \Delta x \leq 1 \tag{3}$$

Obviously, it is not possible to simply transfer these explicit methods to the incompressible case, as the sound speed tends to infinity and thus requires the time-step Δt going to zero. In fact, in the incompressible case, a perturbation at one end of the domain will instantaneously affect the field at the other end, whereas in an explicit scheme, the propagation speed is fundamentally limited to the stencil width (possibly multiplied by a small factor) per time-step.

This is the reason why for the incompressible equations (and equally as well for the compressible equations as long as one is interested in phenomena on time scales much slower than the speed of sound) an implicit or semi-implicit numerical treatment is needed.

In this work, we adopt a scheme based on work by Bell, Collela and Glaz[4], which is a projection method combined with a second-order accurate Godunov method. The scheme has already been successfully applied in an AMR context, e.g. in [7].

Integration is performed in the primitive variables. However, to integrate this scheme with adaptive mesh refinement (AMR), the fields which are communicated between different levels of resolution are the vorticity fields, allowing us to use only first-order interpolation as those represent the highest order derivatives occurring in the time-stepping calculations. Velocities are then calculated as the curl of a flow potential, obviously ensuring the divergence-free condition, where the flow potential is obtained from the vorticity by solving three elliptic equations. At the same time, a fourth Poisson equation is solved to evaluate the pressure using the relation (2) above.

This scheme, requiring the solution of four Poisson equations per time-step is computationally demanding, but at the same time it ensures that no numerical artifacts (like violation of the divergence-free condition) are introduced. Also, the time-step is limited only by the speed of the flow and is thus fairly large. The associated time scale needs to be resolved for accuracy reasons in any case.

The unsplit second-order Godunov scheme is described in detail in reference [4]. It uses a limiting slopes method in calculating approximations to spatial derivatives to prevent introducing new maxima or minima into the velocity field. In the actual time-stepping procedure, it exploits knowledge from treating the 1D Burgers equation, using an upwind scheme for actively transported quantities (using the solution for the Riemann problem):

$$\begin{aligned} &\partial_t u + u\partial_x u = 0 \\ &\Longrightarrow u_{i+1/2,j} = \begin{cases} u^L \text{ if } u^L \geq 0,\ u^L + u^R \geq 0 \\ 0 \quad \text{otherwise} \\ u^R \text{ if } u^R \leq 0,\ u^L + u^R \leq 0 \end{cases} \end{aligned}$$

For the passively transported quantities, the upwind scheme is

$$\partial_t v + u\partial_x v = 0$$

Level	# grids	# grid points
0	1	70225
1	83	146080
2	103	268666
3	153	545316
4	197	1042132
5	404	1926465
6	600	1967234

Grid points in adaptive simulation:	6976118
Grid points in non-adaptive simulation:	268730449
Ratio:	0.02

Table 1. Comparison of grid points needed with AMR to an equivalent uniform mesh simulation.

$$\Longrightarrow v_{i+1/2,j} = \begin{cases} v^L & \text{if } u_{i+1/2,j} \geq 0 \\ 1/2(v^L + v^R) & \text{otherwise} \\ v^R & \text{if } u_{i+1/2,j} \leq 0 \end{cases}$$

2.2 Adaptive mesh refinement

Looking for singularities, the method of adaptive mesh refinement is an ideal tool. Starting out from a uniform grid, the local truncations errors are monitored and new, refined grids are introduced to cover those regions where the error exceeds a certain threshold. In addition, buffer cells are included in the direction of anticipated movement of the small-scale structures to ensure sufficient resolution until the next refinement procedure. While not being generally useful for, e.g., turbulence studies, where small-scale structures form all over, adaptive mesh refinement proves very efficient for problems where only localized small scales occur, such as in the development of a singularity. It enables us to follow the evolving singular structures with a much higher precision than what a legacy uniform grid code could achieve.

To exemplify the efficency gain obtainable by using AMR, we show an analysis from previous work (current sheets developing in 2D MHD) in Table 1 [6]. Taking into account all the ghost cells needed for the adaptive treatment, the adaptive simulation still uses only 2 percent of the resources needed to solve the problem to the same precision on a uniform grid.

Within our integration with the PETSc [1, 2, 3] library, most of the operations applied during the numerical integration can be written in terms of a sparse matrix multiplication. An advantage here is that we can exploit PETSc's optimized linear algebra operations, which overlap communication costs for passing of non-local boundary data with computation of the processor-local data. It is of course still desirable to keep as much of the work local as possible, which is achieved by using a Hilbert-Peano curve for load-balancing. Our AMR technology uses a so-called oct-tree approach, i.e., we

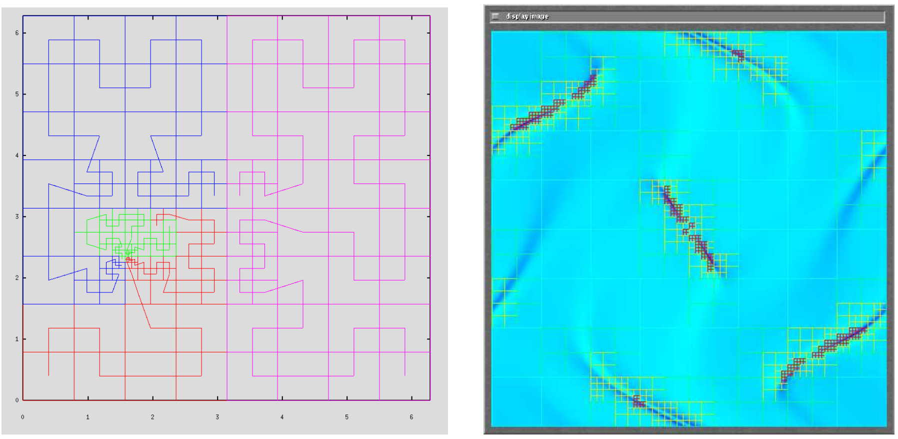

Fig. 1. a) sample adaptive domain coverage and Hilbert-Peano curve, b) Adaptive refinement of current sheets in 2D Hall-MHD.

do not support patches of arbitrary extensions, but instead always break up one box which needs refinement into eight (four in 2D) new boxes of the same shape, which provides a slightly less effective covering but has advantages when it comes to parallelizing and load-balancing.

For visualization purposes, we show the adaptive covering with different levels of resolution for a 2D example, current sheets developing in Hall-MHD simulated within our framework, in Figure 1b. The basic procedure of subdividing grids can clearly be seen there.

The basics of the load-balancing algorithm are visualized in Figure 1a. It shows a sample adaptive domain coverage with adaptive grids at different levels of resolution, as well as the Hilbert-Peano curve mapping those grids and the resulting distribution onto processors.

The Hilbert-Peano curve has the mathematical property that grids that are close in two or three dimensions, respectively, are on average also close on the one-dimensional curve. The one-dimensional curve is divided into equal intervals, the grids on each interval are assigned to a particular processor. In the figure, the colors denote different processors, and it can easily be seen that grids which are kept on a particular processor are generally clustered together, which means the local set of grids typically has a lot of common boundaries not requiring expensive inter-processor communication, but only few boundaries to grids on other processors.

2.3 Multi–level elliptic solvers

A major challenge for the numerical integration of the incompressible Euler equations is the need to solve elliptic equations on the grid hierarchy generated by the AMR framework. The underlying strategy employed here is a variant of the *fast adaptive composite algorithm (FAC)* developed by McCormick [9].

The derivation of FAC algorithm starts with the regular multi–grid algorithm and is best understood considering just two levels, coarse and fine.

Iterative methods, in the simplest case e.g. Gauss-Jacobi or Gauss-Seidel, are called *smoothers* in the multi–level context, since they tend to smooth out the residual – the difference between current approximation and desired right-hand side. Error components with small scales/high frequency are eliminated quickly by these methods, whereas large scales/low frequency modes converge slowly. Smoothers quickly achieve an intermediate approximation which, after removal of high-frequency modes, has only lower frequencies left – it looks *smooth*. The trouble is that from that point on convergence is slow, and one way out of this difficulty are multi-grid methods: the smooth error is restricted onto a coarser domain, where it appears as having higher frequency components, which are quickly removed using the same iterative schemes as on the coarser level. This trick can be applied recursively and will approach $O(N \log N)$, or in the case of *full multigrid* even $O(N)$ scaling.

The idea behind FAC is that the quality of the solution obtained from the coarse grid will actually be sufficient on the part of the domain where the field is smooth, whereas on the other part of the domain we have smaller scales that need to be fully resolved on the fine level. We can meet this goal by only executing the fine level iteration on a *local* subset of the domain. While this scheme gives a solution to the desired accuracy, it is not practical, since it requires us to store the fine level data on the entire domain, even in the areas where that resolution is not required.

However, it can easily be shown that the fine grid relaxation will only change the residual transferred to the coarse grid on the iterated domain plus the interface, so it is not necessary to store the fine data everywhere but only on the refined region plus a certain border. This leads to the *bordered multi–level scheme* which is certainly practical, in particular in an AMR context where we need to take provisions to store boundary data for the time evolution in any case [9]:

> *Residuals are computed on the local fine grid including its boundary and first border; these residuals are then transferred to the coarse grid points lying ander the fine grid and at the interface; the correction is the computed and interpolated to the local fine grid including its boundary and both borders; relaxation is then performed at fine grid interior points.*

However, instead of using just a simple relaxation scheme like Gauss-Jacobi, we integrate this algorithm within the PETSc toolkit, giving us access to a large variety of preconditioners and Krylov-accelerators. At the same time, we can also exploit PETSc's optimized computational kernels and integrated parallelization support.

Effectively, the elliptic problem distributed over the grid hierarchy is mapped to a standard linear solve where the sparse matrix is determined

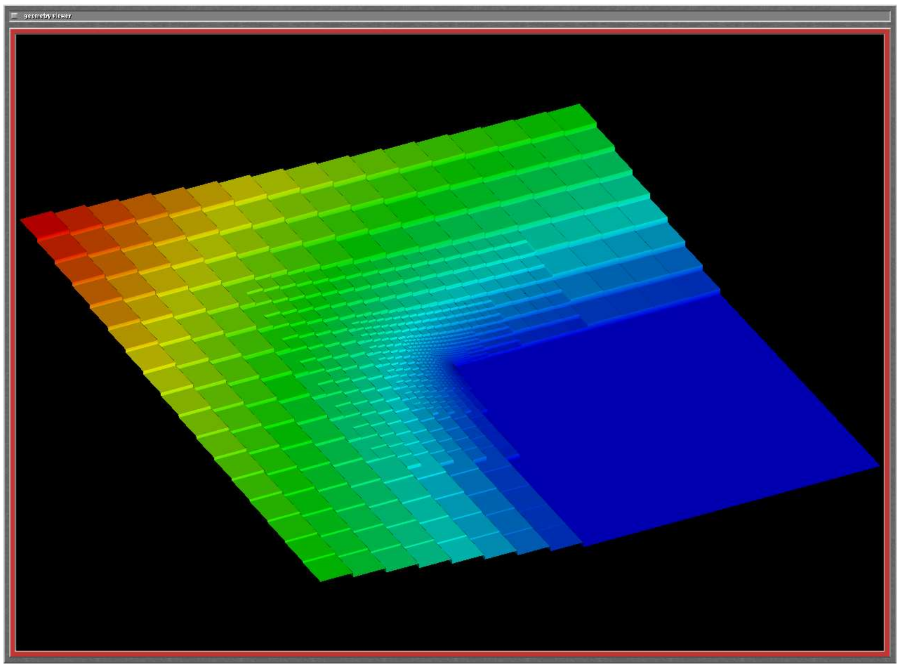

Fig. 2. 2D test problem solution for adaptive elliptic solver.

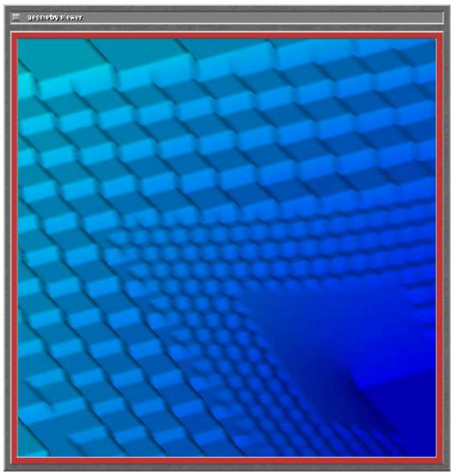

Fig. 3. Zoom into central region of Fig. 2.

by the stencil of the Poisson operator and the mapping of a three-dimensional AMR domain covering to a simple one-dimensional vector.

A main objective of the PETSc library is to efficiently solve this kind of distributed sparse matrix linear problem, using optimal Krylov-Schwarz

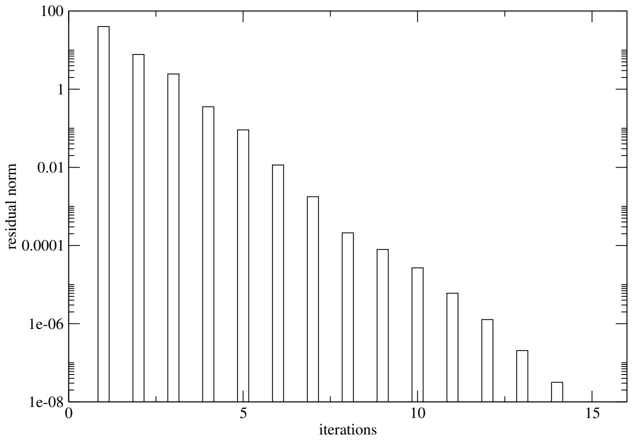

Fig. 4. Convergence of the multi-level elliptic solver for the 2-D test problem.

solvers with multi–level preconditioning which approach $O(N)$ efficiency. The run-time behavior of those solvers is basically determined by recurring matrix-vector-products. The implementation of this important operation overcomes communication bottlenecks by overlapping the communication time needed to obtain non-local data with processor-local computation work, followed by completion of the job through computation involving the by then-arrived non-local data.

A demonstration of the adaptive elliptic solver can be seen in Figures 2 and 3, where the method described is used to solve a standard two-dimensional test problem. The problem at hand is an L-shaped domain, solving Laplace's equation $\nabla^2 u = 0$ with boundary conditions so chosen that the solution is $u = r^{2/3}\sin(2\phi/3)$. This solution has a singular derivative at the origin, making it a good problem to test a multi–level AMR domain covering, with grids of finer resolution added as one approaches the singularity at the origin.

Figure 4 shows the convergence of the solver, in spite of the asymmetry/singular behavior in the problem, we obtain a reasonably fast convergence rate.

3 Tests

To validate the new code, preliminary test runs have been performed on NERSC's seaborg IBM SP machine, using up to 1024 processors.

We use Kida's [8] initial condition for a high-symmetry periodic flow, originally designed to facilitate turbulence studies in Navier-Stokes.

$$u_x(x,y,z) = \sin x(\cos 3y \cos z - \cos y \cos 3z) \tag{4}$$
$$u_y(x,y,z) = \sin y(\cos 3z \cos x - \cos z \cos 3x) \tag{5}$$
$$u_z(x,y,z) = \sin z(\cos 3x \cos y - \cos x \cos 3y) \tag{6}$$

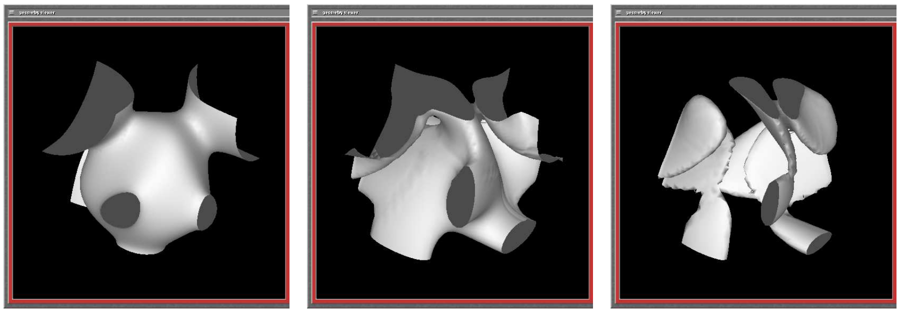

Fig. 5. Isosurfaces of the absolute vorticity value, at times 0, 0.525, 0.735, shown is one octant of the domain

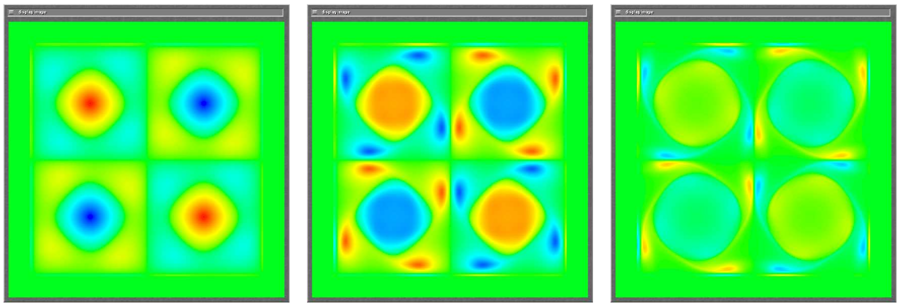

Fig. 6. y-component of the vorticity, cut through $y = 0$, at times 0, 0.5, 1.1

This initial condition is known to form six vortex dipoles at the origin, which may evolve into a finite-time singularity [5, 10].

Figure 5 shows isosurfaces of the absolute value of the vorticity at times and magnitudes comparable to those given in [5] in one octant of the domain.

Figure 6 shows the y-component of the vorticity at the $y = 0$ plane evolving in time. The formation and beginning collapse of two vortex dipoles near the origin is clearly observed (the remaining 4 dipoles form in the x and z-component of the vorticity, not shown here).

For validation purposes, we compared numerical data from the pseudo-spectral code described in [5] and from the code described in [7], run with Kida's initial condition, to preliminary runs of the MRC code described in this paper and find good agreement in the vorticity isosurfaces as well as in the time evolution of maximum vorticity.

4 Conclusion

In this paper, we introduced the numerical methods used in the application of the Magnetic Reconnection Code (MRC) to solve the three-dimensional Euler equations.

We focused on presenting how the integration of Krylov-Schwarz methods with a hierarchical set of grids, obtained using block-structured adaptive mesh refinement, can provide a performant method to solve elliptic problems, as they occur in many problems in computational fluid dynamics / plasma flows.

Exploiting and extending existing technology, the PETSc library, enabled us to use a wide variety of well established sophisticated solver technologies and kept the implementation effort manageable.

At this point of writing, we have been able to perform only preliminary test runs of our new code. The results show promise for following an evolving singularity farther in time than has been possible previously with fixed uniform resolution.

References

1. Satish Balay, Kris Buschelman, William D. Gropp, Dinesh Kaushik, Matt Knepley, Lois Curfman McInnes, Barry F. Smith, and Hong Zhang. PETSc home page. http://www.mcs.anl.gov/petsc, 2001.
2. Satish Balay, Kris Buschelman, William D. Gropp, Dinesh Kaushik, Matt Knepley, Lois Curfman McInnes, Barry F. Smith, and Hong Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 2.1.5, Argonne National Laboratory, 2002.
3. Satish Balay, William D. Gropp, Lois Curfman McInnes, and Barry F. Smith. Efficient management of parallelism in object oriented numerical software libraries. In E. Arge, A. M. Bruaset, and H. P. Langtangen, editors, *Modern Software Tools in Scientific Computing*, pages 163–202. Birkhauser Press, 1997.
4. J. B. Bell, P. Colella, and H. M. Glaz. A second-order projection method for the incompressible Navier–Stokes equations. *J. Comput. Phys.*, 85:257–283, 1989.
5. O. N. Boratav and R. B. Pelz. Direct numerical simulation of transition to turbulence from a high-symmetry initial condition. *Phys. Fluids*, 6:2757–2784, 1994.
6. H. Friedel, R. Grauer, and C. Marliani. Adaptive mesh refinement for singular current sheets in incompressible magnetohydrodynamic flows. *J. Comput. Phys.*, 134:190–198, 1997.
7. R. Grauer, C. Marliani, and K. Germaschewski. Adaptive mesh refinement for singular solutions of the incompressible Euler equations. *Phys. Rev. Lett.*, 80:4177–4180, 1998.
8. S. Kida. Three-dimensional periodic flows with high-symmetry. *J. Phys. Soc. Jpn.*, 54:2132, 1985.
9. S.F. McCormick. *Multilevel adaptive methods for partial differential equations*, volume 6 of *Frontiers in Applied Mathematics*. SIAM, 1989.
10. R.B. Pelz. Locally self-similar, finite-time collapse in a high-symmetry vortex filament model. *Phys. Rev. E*, 55(2):1617–1626, 1997.