Acceleration of Time Integration

JB White III, K Evans, R Archibald, J Drake, P Worley, D Kothe

May 16, 2008

Abstract

We outline our strategies for accelerating time integration for long-running simulations, such as those for global climate modeling. The strategies target the Cray XT systems at the National Center for Computational Sciences at Oak Ridge National Laboratory. Our strategies include fully implicit, parallel-in-time, and curvelet methods.

1 Introduction

In his keynote presentation to the user group of the National Center for Computational Sciences in 2007, Lawrence Buja made the following observation [2]. Assessment Report 4 (AR4) of the Intergovernmental Panel on Climate Change [15] marked a change in the role of global climate simulation. Before AR4, the goal of climate simulation was to determine if the Earth's climate is warming and if humans are the primary cause. AR4 established that the answer to these questions is "yes".

Now climate simulation must focus on more-challenging questions, questions of impacts, mitigation, adaptation, and regional details [2]. The answers will require spacial resolution much finer than the 300-km resolution common in current simulations, according to the leading computational geoscientists who wrote *Establishing a Petascale Collaboratory for the Geosciences: Scientific Frontiers* [6].

> More importantly, because the assumptions that are made in the development of parameterizations of convective clouds and the planetary boundary layer are seldom satisfied, the atmospheric component model must have sufficient resolution to dispense with these parameterizations. This would require a horizontal resolution of 1 km.

A critical issue with approaching a resolution of 1 km is what we call the "time barrier". Current climate models use explicit time integration, which means that the time step must go down if the resolution goes up. These models, with 300-km resolution, take time steps on the order of 20 minutes [5]. A similar model with 1-km resolution would need a time step of 4 seconds. The growing parallelism of extreme-scale computers may provide adequate capability to handle the increase in spacial resolution, but single-process performance is stagnating, so the performance of serial time integration is also stagnating. Thus multi-century climate simulations are unrealistic with a 4-second time step.

We have initiated a project at Oak Ridge National Laboratory to attack this issue of time integration for high-resolution simulations. The project, called "Fast Forward", includes three complementary strategies for accelerating time integration: fully implicit time integration, parallel-in-time methods, and new space and time discretizations. In the following sections, we describe our plans and progress for each of these strategies.

Research sponsored by the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory, managed by UT-Battelle, LLC for the US Department of Energy under Contract No. DE-AC05-00OR22725. This research used resources of the National Center for Computational Sciences at Oak Ridge National Laboratory, which is supported by the Office of Science of the US Department of Energy under Contract No. DE-AC05-00OR22725.

2 Fully implicit time integration

Though our eventual goal is full climate models, our current target for fully implicit time integration is the shallow-water equations on the sphere [17].

$$\frac{\partial h^* \mathbf{v}}{\partial t} + \nabla \cdot (\mathbf{v} h^* \mathbf{v}) = -f \hat{\mathbf{k}} \times h^* \mathbf{v} - g h^* \nabla h \tag{1}$$

$$\frac{\partial h^*}{\partial t} + \nabla \cdot (h^* \mathbf{v}) = 0 \tag{2}$$

$$h = h^* + h_s \tag{3}$$

Here h is the water height, h_s is the height of the sea floor, and h^* is the water depth. The water velocity is $\mathbf{v}$, f is the Coriolis parameter, and g is the acceleration from gravity. These equations mimic full equations for atmosphere and ocean models. Our plan is to evaluate our strategies using a series of standardized tests for the shallow-water equations on the sphere [17]. We may then proceed to three-dimensional tests and full models.

To illustrate the potential advantages of implicit methods, we contrast them to explicit methods. Consider y, the state of a system of interest at a known time, and y', the state of the same system at a future time. An explicit method computes the future state y' directly in terms of quantities defined at the current time.

$$y' = f(y) \tag{4}$$

The strength of this method is that it is typically highly parallel, often with only nearest-neighbor communication. The weakness is that it is numerically unstable—it "blows up"—for large time steps. The time step must be proportional to the grid spacing; if the grid spacing decreases, thus increasing resolution, the time step must decrease.

An implicit method typically defines the future state y' in terms of a constant multiple of the previous state y and quantities defined at the future time.

$$y' = ay + f(y') \tag{5}$$

Thus we have a system of equations, often nonlinear equations, that must be solved for each time step. This is the weakness of this method. An important strength, however, is that the method can be stable for arbitrary time steps. The size of the time step still affects accuracy, but it is no longer constrained by numerical instability.

Our shallow-water experiments start with the High-Order-Method Modeling Environment (HOMME) [9]. HOMME uses a cubed-sphere grid and supports multiple methods and systems of equations. We start with the spectral-element method and, of course, the shallow-water equations.

We have converted the explicit method in HOMME to implicit using a Jacobian-free Newton-Krylov method [12]. The Newton method solves the nonlinear system using multiple linear solutions. Consider a starting guess y for a nonlinear system of equations; we want $F(y) = 0$, but our starting guess is typically such that $F(y) \neq 0$. To find a better y, we find the change in F as y changes, or the derivative of F with respect to y, also known as the Jacobian, J. We want a correction Δy such that $F(y + \Delta y) = 0$, and we find an approximate Δy using the Jacobian.

$$0 = F(y + \Delta y) \approx F(y) + J \Delta y \tag{6}$$

$$F(y) = -J \Delta y \tag{7}$$

Now we have a linear equation for Δy. Solve this, add Δy to y, and repeat until $F(y) \approx 0$.

The "Krylov" piece of JFNK contributes to solving the linear systems. We solve for Δy using an iterative Krylov-subspace method. We first make a guess for Δy and calculate how bad the guess is, where the difference is called the "residual". We then use the residual to improve the guess. Iterate this process,

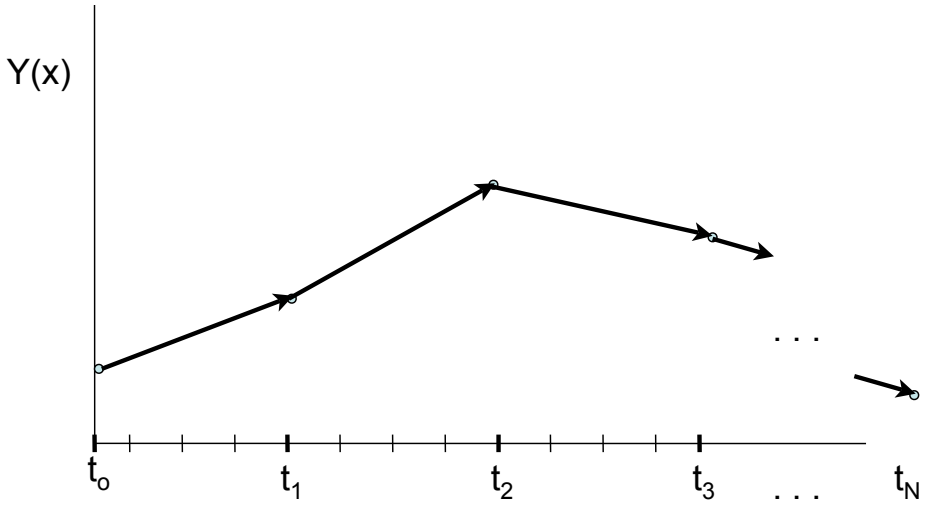

Figure 1: First phase of the parareal algorithm, solving serially at coarse time steps. Dark arrows indicate serial propagation between time steps, with the resulting values indicated by circles.

using clever combinations of past residuals to improve Δy. These "clever combinations" are determined by the details of the particular Krylov method. Iterate until the residual is small enough.

The final piece of JFNK comes from avoiding direct computation of the Jacobian. Instead of building the Jacobian matrix J and then applying it to Δy, we approximate the matrix-vector product by forming a numerical derivative.

$$J\Delta y \approx \frac{F(y + \epsilon\Delta y) - F(y)}{\epsilon} \tag{8}$$

This approximate calculation can be much cheaper than forming J directly, and it requires only the ability to numerically compute F; it does not require analytic knowledge of F.

We have implemented an implicit method as described above in HOMME for the first two test cases of [17]. We use solvers available in the Trilinos package [11]. The first test case is linear advection of a cosine bell around the sphere. After 12 simulated days, the bell returns to its original position. The explicit solver in HOMME uses 2-minute time steps for this case, while the implicit implementation is able to take 2-hour time steps, and the resulting solution at 12 days has similar accuracy. Of course the implicit solver takes multiple iterations for each time step, reducing the performance benefit. Still, the overall runtime is 40% less, and improved preconditioners should increase this advantage.

The second test case is a steady-state solution integrated for 12 days. The explicit implementation uses a 4-minute time step, resulting in a run time of 28 seconds. Because the solution is a steady state, the implicit implementation can take a single 12-day time step, resulting in a run time of just 3.6 seconds.

3 Parallel in time

The "parareal" algorithm was first published in 2001 by Jacques-Louis Lions, Yvon Maday, and Gabriel Turincini [13]. The original algorithm and variants have proven successful for a range of applications, including Navier-Stokes [8], structural dynamics [7], and reservoir simulation [10].

The first phase of parareal is to propagate serially using coarse time steps, as depicted in Figure 1. The middle phase, shown in Figure 2, is to perform fine-scale time integrations starting from each coarse step, where the integrations occur in parallel. The final phase, as shown in Figure 3, is to serially propagate and accumulate the correction at each coarse step, where the corrections come from the fine-scale integrations.

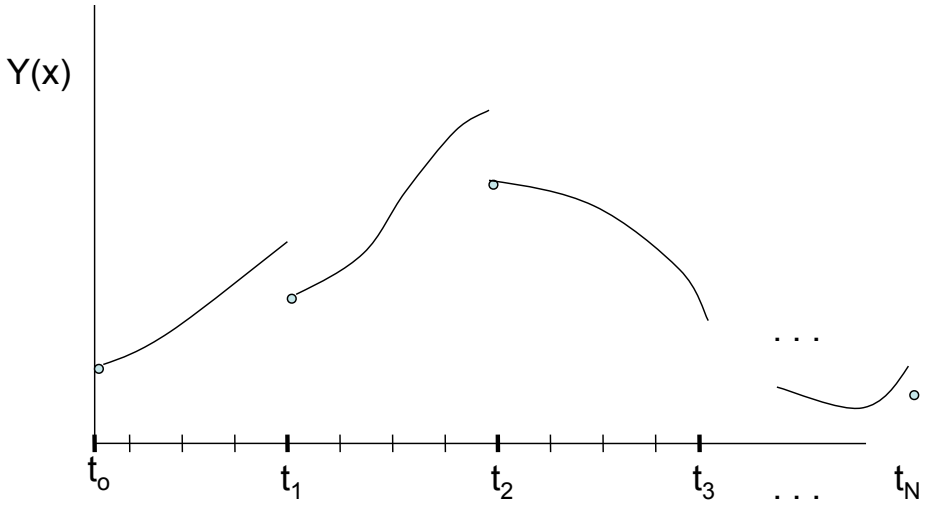

Figure 2: Middle phase of the parareal algorithm, parallel fine-scale time integrations starting at each coarse time step. Circles indicate coarse-time points, and lines indicate fine-scale time integration from each coarse-time point.

Iterate these three phases until the final correction is small enough. Note that the parallel efficiency of this algorithm is limited by the number of iterations. Published results [13, 8, 7, 10] suggest that two or three iterations may be adequate.

Thus far we have only implemented the parareal algorithm for pure advection in one dimension, with constant velocity v.

$$\frac{\partial u}{\partial t} + v\frac{\partial u}{\partial x} = 0 \tag{9}$$

Even using fully implicit time integration that is unconditionally stable when used alone, our implementation of the parareal algorithm is unstable for pure linear advection. Our results match the theoretical predictions of [14], and, based on this analysis, we expect nonlinear problems to not suffer from this instability. We next plan to investigate parareal with the one-dimensional Burgers' equation to strengthen this expectation.

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - \nu\frac{\partial^2 u}{\partial x^2} = 0 \tag{10}$$

4 New discretizations

Different discretizations in space and time can complement implicit methods and parallelism in time to improve time integration. New discretizations we are investigating include curvelets, multi-wavelets, and novel applications of finite differences.

Curvelets [3, 4, 16] are a specialized basis that combines the benefits of compact and shape-preserving bases. Curvelets are compact, or have limited spacial extent, like finite elements, and they also preserve their shape for long periods of time, like Fourier waves. This shape-preserving property may allow stability for time steps of order $\sqrt{\Delta x}$ instead of Δx. And their compactness makes them an optimally sparse representation of wave propagation [3]. A difficulty with curvelets is that they require a periodic domain, so we are currently investigating periodic mappings of the cubed sphere.

Multi-wavelets [1] are another specialized basis with useful properties. Multi-wavelets are adaptive and designed specifically for spacial refinement, so they efficiently represent features at different spacial scales. They have strong error bounds that facilitate adaptive refinement and coarsening. Multi-wavelets work best

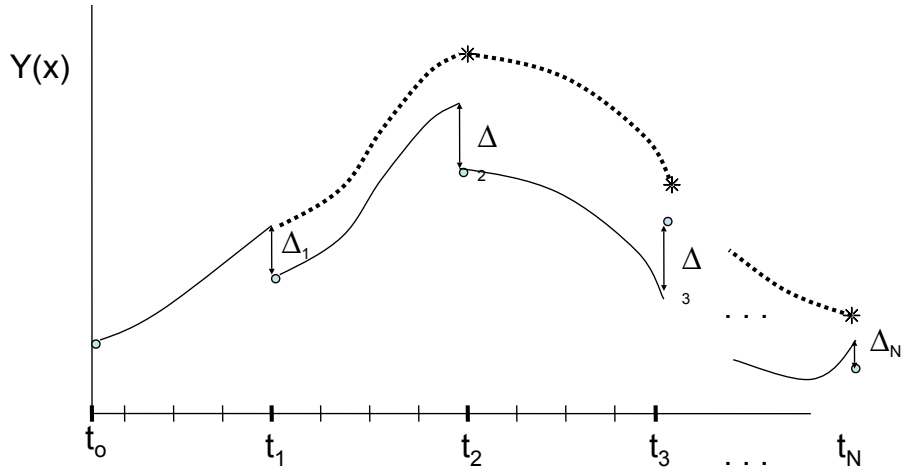

Figure 3: Final phase of the parareal algorithm, serial propagation and accumulation of fine-scale corrections at each coarse step. The Δ terms indicate corrections of the coarse-time points given by the fine-time integrations. The "*" points connected by dotted lines indicate the resulting coarse-time points with accumulated corrections.

with an integral formulation of the target problem, and we have just started development of an integral formulation of the shallow-water equations on the sphere.

In addition to these new bases, we are looking at novel applications of traditional finite differences. One such application is high-order single-step time integration. To illustrate this method, consider the simple example of advection in one dimension.

$$\frac{\partial u}{\partial t} + v\frac{\partial u}{\partial x} = 0 \tag{11}$$

Here v is a constant velocity.

We can approximate the time derivative using a Taylor series in the small time step Δt. Let u be the state at a known time and u' be the future state we want to compute.

$$u = u' - \Delta t\frac{\partial u'}{\partial t} + \frac{\Delta t^2}{2}\frac{\partial^2 u'}{\partial t^2} - \frac{\Delta t^3}{6}\frac{\partial^3 u'}{\partial t^3} + O(\Delta t^4) \tag{12}$$

Note that this series is in an implicit form, written in terms of the future state u'.

We can replace the time derivatives with space derivatives using Equation 11. This is useful because we typically have many grid points in space, allowing us to form high-order space derivatives. But we want the minimal number of points in time: two, the most-recent time with known state and the next time with unknown state. To change time derivatives to space derivatives, use Equation 11 as follows.

$$\frac{\partial u'}{\partial t} = -v\frac{\partial u'}{\partial x} \tag{13}$$

By taking derivatives of this transformation and applying it recursively, we can change Equation 12 to the following.

$$u = u' + v\Delta t\frac{\partial u'}{\partial x} + \frac{v^2\Delta t^2}{2}\frac{\partial^2 u'}{\partial x^2} + \frac{v^3\Delta t^3}{6}\frac{\partial^3 u'}{\partial x^3} + O(\Delta t^4) \tag{14}$$

Thus, if we have high-order space derivatives, we can get high accuracy in time "for free", without additional points in time, but with additional floating-point operations. Such operations are relatively inexpensive on modern computers. The ability to use just two points in time has multiple advantages, including memory

savings, savings in I/O storage space and bandwidth for checkpoints and restarts, and simple start ups from a single set of initial conditions.

We have explicit and implicit implementations of high-order single-step time integration for one-dimensional advection, and an explicit implementation for Burgers' equation. We are currently developing implicit and semi-implicit implementations for Burgers' equation, with the later goal of the shallow-water equations on the sphere.

5 Summary

We have outlined some of the strategies we are investigating in the Fast Forward project, where our goal is to accelerate time integration for important science applications that need higher resolution and longer simulated times, applications like climate simulation. Implicit methods will allow us to choose time steps based on our accuracy requirements, not based on the much tighter constraint of numerical stability for explicit methods. Parallel-in-time methods may allow us to take advantage of the largest dimension in our applications of interest, the time dimension, for parallel execution on extreme-scale computers. And improved discretizations could compound the improvements provided by the other strategies, accelerating time integration further still.

Which of these strategies will prove beneficial? Time will tell.

References

[1] B Alpert, G Beylkin, D Gines, and L Vozovoi. Adapative solution of partial differential equations in multiwavelet bases. *Journal of Computational Physics*, 182:149–190, 2002.

[2] L Buja. Climate models: From IPCC to the petascale. Keynote to NCCS User Group meeting, March 2007.

[3] E Candes and L Demanet. The curvelet representation of wave propagators is optimally sparse. *Communications on Pure and Applied Mathematics*, 11:1472–1528, 2005.

[4] E Candes, L Demanet, D Donoho, and LX Ying. Fast discrete curvelet transforms. *Multiscale Modeling and Simulation*, 5(3):861–899, 2006.

[5] WD Collins, PJ Rasch, BA Boville, JJ Hack, DL Williamson, JT Kiehl, B Briegleb, C Bitz, SJ Lin, M Zhang, and Y Dai. Description of the NCAR Community Atmosphere Model (CAM 3.0). Technical Note 464+STR, NCAR, June 2004.

[6] Ad Hoc Committee and Technical Working Group for a Petascale Collaboratory for the Geosciences. *Establishing a Petascale Collaboratory for the Geosciences: Scientific Frontiers.* UCAR/JOSS, 2005.

[7] C Farhat, J Cortial, C Dastellung, and H Bavestrello. Time-parallel implicit integrators for the near-real-time prediction of linear structural dynamics responses. *International Journal for Numerical Methods in Engineering*, 67:697–724, 2006.

[8] PF Fischer, F Hecht, and Y Maday. A parareal in time semi-implicit approximation of the navier-stokes equations. *Proceedings of the 15th International Conference on Domain Decomposition Methods*, 2003.

[9] A Fournier, MA Taylor, and JJ Tribbia. The Spectral Element Atmosphere Model (SEAM): High-resolution parallel computation and localized resolution of regional dynamics. *Monthly Weather Review*, 132(10):726–748, 2004.

[10] I Garrido, B Lee, GE Fladmark, and MS Espedal. Convergent iterative schemes for time parallelization. *Mathematics of Computation*, 75(255):1403–1428.

[11] M Heroux, R Bartlett, V Howle, R Hoekstra, J Hu, T Kolda, R Lehoucq, K Long, R Pawlowski, E Phipps, A Salinger, H Thornquist, R Tuminaro, J Willenbring, and A Williams. An overview of trilinos. Technical Report 2003-2927, Sandia National Laboratories, 2003.

[12] DA Knoll and DE Keyes. Jacobian-free newton-krylov methods: A survey of approaches and applications. *Journal of Computational Physics*, 193(2):357–397, 2004.

[13] JL Lions, Y Maday, and G Turinici. A "parareal" in time discretization of PDEs. *Computes Rendus de l'Académie des Sciences—Series I—Mathematics*, 332(7):661–668, April 2001.

[14] Y Maday, E Rønquist, and GA Staff. The parareal-in-time algorithm: Basics, stability analysis, and more. In preparation.

[15] S Solomon, D Quin, M Manning, M Marquis, K Averyt, MMB Tignor, HL Miller, Jr., and Z Chen, editors. *Climate Change 2007: The Physical Science Basis*. Cambridge University Press, 2007.

[16] J Starck, M Nguyen, Y Moudden, and P Abrail. Wavelets, ridgelets, and curvelets on the sphere. *Astronomy and Astrophysics*, 446(3):1191–1204, 2006.

[17] DL Williamson, JB Drake, JJ Hack, R Jakob, and PN Swarztrauber. A standard test set for numerical approximations to the shallow water equations in spherical geometry. *Journal of Computational Physics*, 102(1):211–224, September 1992.