Newton-Krylov-Schwarz Method for a Spherical Shallow Water Model*

Chao Yang[1] and Xiao-Chuan Cai[2]

[1] Institute of Software, Chinese Academy of Sciences, Beijing 100190, P. R. China, `yang@mail.rdcps.ac.cn`

[2] Department of Computer Science, University of Colorado at Boulder, Boulder, CO 80309, USA, `cai@cs.colorado.edu`

1 Introduction

In this paper we study the application of Newton-Krylov-Schwarz method to fully implicit, fully coupled solution of a global shallow water model. In particular, we are interested in developing a scalable parallel solver when the shallow water equations (SWEs) are discretized on the cubed-sphere grid using a second-order finite volume method.

2 Governing Equations

The cubed-sphere grid of gnomonic type [7, 8] is used in this study. The grid is generated by mapping the six faces of an inscribed cube to the sphere surface using gnomonic projection. The six expanded patches are continuously attached together with proper boundary conditions. On each patch, the expressions of the SWEs in local curvilinear coordinates $(x, y) \in [-\pi/4, \pi/4]^2$ are identical. When no bottom topography is involved, the SWEs can be written in the following conservative form:

$$\frac{\partial Q}{\partial t} + \frac{1}{\Lambda}\frac{\partial(\Lambda F)}{\partial x} + \frac{1}{\Lambda}\frac{\partial(\Lambda G)}{\partial y} + S = 0, \tag{1}$$

with

$$Q = \begin{pmatrix} h \\ hu \\ hv \end{pmatrix}, \; F = \begin{pmatrix} hu \\ huu + \frac{1}{2}gg^{11}h^2 \\ huv + \frac{1}{2}gg^{12}h^2 \end{pmatrix}, \; G = \begin{pmatrix} hv \\ huv + \frac{1}{2}gg^{12}h^2 \\ hvv + \frac{1}{2}gg^{22}h^2 \end{pmatrix}, \; S = \begin{pmatrix} 0 \\ S_1 \\ S_2 \end{pmatrix},$$

* The first author was supported in part by NSFC grant 10801125, in part by 973 China grant 2005CB321702, and in part by 863 China grants 2006AA01A125. The second author was supported in part by DOE under DE-FC-02-06ER25784, and in part by NSF under grants CCF-0634894 and DMS 0913089.

and

$$S_1 = \Gamma^1_{11}(huu) + 2\Gamma^1_{12}(huv) + f\Lambda\left(g^{12}hu - g^{11}hv\right),$$
$$S_2 = 2\Gamma^2_{12}(huv) + \Gamma^2_{22}(hvv) + f\Lambda\left(g^{22}hu - g^{12}hv\right).$$

Here h is the fluid thickness, (u, v) are contravariant components of the fluid velocity, g is the gravitational constant and f is the Coriolis parameter due to the rotation of the sphere. The variable coefficients g^{mn}, Λ and Γ^{ℓ}_{mn} are only dependent on the curvilinear coordinates [12].

3 Discretizations

A uniform rectangular $N \times N$ grid is used on each patch. Grid cell $\mathcal{C}_{ij}$ is centered in (x_i, y_j), $i, j = 1, \cdots, N$, with grid size $\Delta x = \Delta y = \pi/2N$. The approximate solution in cell $\mathcal{C}_{ij}$ at time t is defined as

$$Q_{ij} \approx \frac{1}{\Lambda_{ij}\Delta x \Delta y} \int_{y_j-\Delta y/2}^{y_j+\Delta y/2} \int_{x_i-\Delta x/2}^{x_i+\Delta x/2} \Lambda(x,y)Q(x,y,t)\, d\,x\, d\,y,$$

where Λ_{ij} is evaluated at the cell center of $\mathcal{C}_{ij}$. Then we have the following semi-discrete system of the SWEs:

$$\frac{\partial Q_{ij}}{\partial t} + \frac{(\Lambda F)_{i+\frac{1}{2},j} - (\Lambda F)_{i-\frac{1}{2},j}}{\Lambda_{ij}\hbar} + \frac{(\Lambda G)_{i,j+\frac{1}{2}} - (\Lambda G)_{i,j-\frac{1}{2}}}{\Lambda_{ij}\hbar} + S_{ij} = 0. \quad (2)$$

Here the numerical fluxes are approximated using the Osher's Riemann solver [5, 6], i.e.,

$$(\Lambda F)_{i+\frac{1}{2},j} = \Lambda_{i+\frac{1}{2},j} F^{(o)}(Q^-_{i+\frac{1}{2},j}, Q^+_{i+\frac{1}{2},j}) = \Lambda_{i+\frac{1}{2},j} F(Q^*_{i+\frac{1}{2},j}),$$

with

$$h^* = \frac{1}{4gg^{11}}\left[\frac{1}{2}\left(u^- - u^+\right) + \sqrt{gg^{11}h^-} + \sqrt{gg^{11}h^+}\right]^2,$$
$$u^* = \frac{1}{2}\left(u^- + u^+\right) + \sqrt{gg^{11}h^-} - \sqrt{gg^{11}h^+},$$
$$v^* = \begin{cases} v^- + \frac{g^{12}}{g^{11}}\left(u^* - u^-\right), & \text{if } u^* \geq 0 \\ v^+ + \frac{g^{12}}{g^{11}}\left(u^* - u^+\right), & \text{otherwise,} \end{cases}$$

where we assume $|u| < \sqrt{gg^{11}h}$. The calculation of G follows an analogous way, see [12] for details. The following two reconstruction methods for constant states are considered in this study:

- Piecewise constant method (first order):

$$Q^-_{i+\frac{1}{2},j} = Q^+_{i-\frac{1}{2},j} = Q_{ij}. \quad (3)$$

- Piecewise linear method (second order):

$$Q^-_{i+\frac{1}{2},j} = Q_{ij} + \frac{Q_{i+1,j} - Q_{i-1,j}}{4}, \quad Q^+_{i-\frac{1}{2},j} = Q_{ij} - \frac{Q_{i+1,j} - Q_{i-1,j}}{4} \tag{4}$$

On each patch interface, one layer of ghost cells is needed and the numerical fluxes are calculated symmetrically across the interface to insure the numerical conservation of total mass, see [11] for details.

Given a semi-discrete system

$$\frac{\partial Q}{\partial t} + \mathcal{L}(Q) = 0,$$

we use the following second-order backward differentiation formula (BDF-2) for the temporal integration:

$$\frac{1}{2\Delta t}\left(3Q^{(m)} - 4Q^{(m-1)} + Q^{(m-2)}\right) + \mathcal{L}(Q^{(m)}) = 0. \tag{5}$$

Here $Q^{(m)}$ denotes Q evaluated at m-th time step with a fixed time step size Δt. Only at the first time step, a first-order backward Euler (BDF-1) is used.

4 Nonlinear Solver

Fully implicit method enjoys an advantage that the time-step size is no longer constrained by the CFL condition. The price to pay is that a large sparse nonlinear algebraic system has to be solved at each time step. In this study, we use Newton-Krylov-Schwarz (NKS) algorithm as the nonlinear solver.

In the NKS algorithm, to solve a nonlinear system $\mathcal{F}(X) = 0$, an inexact Newton's method is used in the outer loop. Let X_n be the approximate solution for the n-th Newton iterate, we find the next solution X_{n+1} as

$$X_{n+1} = X_n + \lambda_n s_n, \quad n = 0, 1, ... \tag{6}$$

where λ_n is the steplength decided by a linesearch procedure and s_n is the Newton correction. We then use the right-preconditioned GMRES (restarted every 30 iterations) method to solve the Jacobian system

$$J_n M^{-1}(M s_n) = -\mathcal{F}(X_n), \quad J_n = \mathcal{F}'(X_n)$$

until the linear residual $r_n = J_n s_n + \mathcal{F}(X_n)$ satisfies

$$\|r_n\| \leq \eta \|\mathcal{F}(X_n)\|.$$

We implement a hand-coded analytic method to generate the Jacobian J_n in the calculation. The accuracy (relative tolerance) of the Jacobian solver is determined uniformly by the nonlinear forcing terms $\eta = 10^{-3}$. Some more flexible methods such as that of [2] may be used to get more efficient or more robust solutions. The Newton iteration (6) ends when the following stopping condition is satisfied

$$||\mathcal{F}(X_{n+1})|| \leq \max\{\varepsilon_r||\mathcal{F}(X_0)||, \varepsilon_a\},$$

where $\varepsilon_r, \varepsilon_a \geq 0$ are nonlinear tolerances.

To achieve uniform residual error at each time step, we use adaptive stopping conditions with both lower and upper adjustments in the NKS method. To do a lower adjustment, we do not let the iteration stop until

$$||\mathcal{F}(X_{n+1})|| \leq 1.0 \times 10^{-5}$$

even when the relative tolerance of $\varepsilon_r = 10^{-7}$ is satisfied. An upper adjustment can be done by setting the absolute tolerance to be $\varepsilon_a^{(0)} = 10^{-8}$ at the first time step and then lettting it adaptively be decided by

$$\varepsilon_a^{(m)} \leftarrow \max\{\varepsilon_a^{(m-1)}, ||\mathcal{F}(X^{(m-1)})||\},$$

where $X^{(m-1)}$ is the converged solution of previous time step.

The preconditioner M^{-1} is obtained by using the restricted additive Schwarz (RAS, [1, 9]) method based on the domain decomposition of the cubed-sphere described briefly here. The six patches of the cubed-sphere can be either simultaneously [12] or independently divided into non-overlapping subdomains. In this study, the six patches are treated in a separated way, i.e., the six patches are respectively decomposed into p non-overlapping rectangular subdomains. Each subdomain is then mapped onto one processor. Thus $6p$ is the total number of processors and subdomains as well. An overlapping decomposition can be obtained by extending each subdomain with δ layers of grid points in all directions. It should be noted that the overlapping area might lie on other patches and directions might also change.

In practice we use a point-wise ordering for both unknowns and the nonlinear equations, resulting in Jacobian matrices with 3×3-block entries. Subdomain solves are done by LU factorizations or incomplete LU (ILU) factorizations with fill-in level ℓ. Here the LU and the ILU factorizations are done in a point-wise manner, i.e., fill-ins are always 3×3 blocks rather than scalars.

5 Numerical Results

Our numerical tests are carried out on an IBM BlueGene/L supercomputer with $4{,}096$ nodes. Each node has a dual-core IBM PowerPC 440 processor running at 700 MHz and with 512 MB of memory. We use the 4-wave Rossby–Haurwitz problem in [10] as the test case in this study. The characteristic time and length scale is one day and the Earth's radius. The result on day 14 is provided in Fig. 1, consistent with the reference solutions in [3, 4].

To test the performance of the preconditioner, we use 192 processor cores to run a fixed size problem on a $512 \times 512 \times 6$ grid for 10 time steps with $\Delta t = 0.1$ days

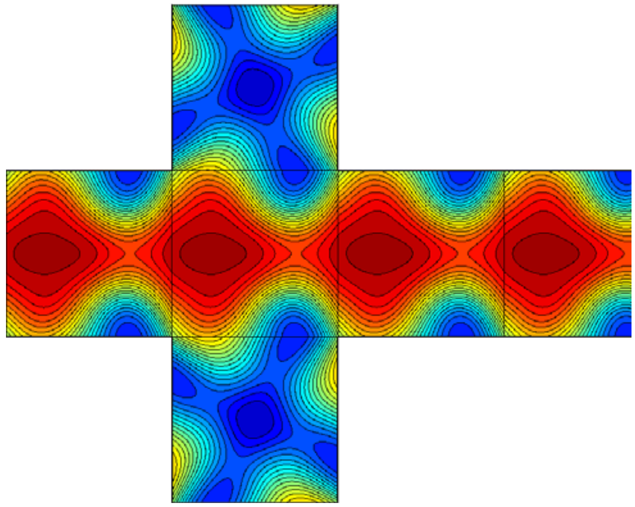

Fig. 1. Height field of Rossby–Haurwitz problem on day 14, grid size $128 \times 128 \times 6$, time step size $\Delta t = 0.1$ days. The contour levels are from $8{,}300$ to $10{,}500$ m with an interval of 100 m. The four innermost lines near to the equators are at $10{,}500$ m.

repeatedly with various levels of overlaps and fill-in ratios. First we try to use RAS preconditioner obtained directly from the Jacobian matrix J_n. In this case the ILU factorizations of the subdomain problems results in many GMRES iterations. If we use LU factorization instead, however, the factorization may fail due to insufficient memories and the performance is very poor even when the factorization succeeds.

Thus we use Jacobian matrix with first-order spatial discretization to construct the RAS preconditioner even when a high order scheme is used in the nonlinear function evaluation. This is based on the fact that Jacobian matrices are all related to the original SWEs no matter what spatial discretization is used. As it can be seen in Tables 1 and 2, the RAS precoditioner works in the NKS algorithm. Larger overlaps or subdomain ILU fill-ins help in reducing the number of GMRES iteration. However, the per-iteration work increases at the same time. The optimal choice in terms of computing time for this test is ILU(3) subdomain solvers with overlapping factor 2.

Using the optimal parameters, we run a set of large-scale tests with the same problem on a $1{,}024 \times 1{,}024 \times 6$ grid with gradually increased number of processor cores. As seen from Fig. 2, our solver scales up to 6,144 processor cores almost linearly with parallel efficiency 73.8%.

Table 1. The number of GMRES iterations per Newton iteration, averaged over the first 10 time steps.

Overlap	0	1	2	3	4
ILU(0)	309.5	299.2	294.4	292.7	291.4
ILU(1)	199.6	178.0	171.7	168.4	166.0
ILU(2)	194.1	150.2	141.0	139.3	137.8
ILU(3)	188.1	134.5	125.3	122.3	120.4
ILU(4)	184.9	127.0	116.1	111.8	110.9
LU	139.9	87.8	78.7	76.2	75.4

Table 2. The averaged compute time (in seconds) over the first 10 time steps.

Overlap	0	1	2	3	4
ILU(0)	10.13	11.13	11.38	11.65	11.90
ILU(1)	7.99	8.36	8.43	8.58	8.67
ILU(2)	8.13	7.84	7.79	7.94	8.09
ILU(3)	8.51	7.80	7.74	7.84	7.99
ILU(4)	8.90	7.96	7.83	7.87	8.04
LU	10.97	9.75	9.88	10.25	10.69

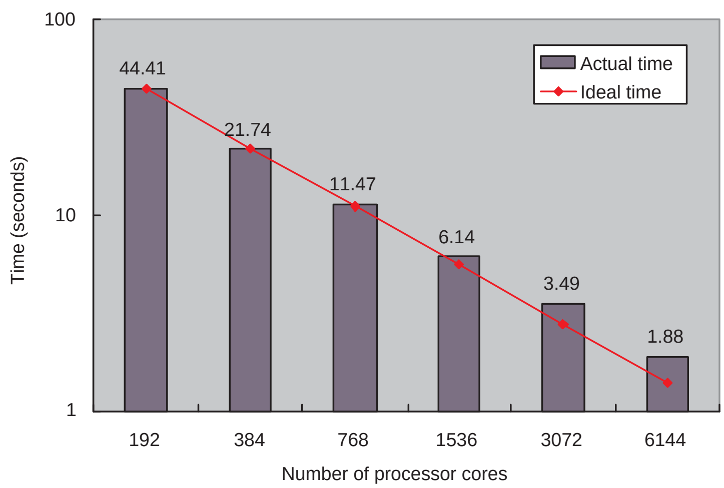

Fig. 2. Compute time curve on the Rossby–Haurwitz problem.

References

1. X.-C. Cai and M. Sarkis. A restricted additive Schwarz preconditioner for general sparse linear systems. *SIAM J. Sci. Comput.*, 21:792–797, 1999.
2. S.C. Eisenstat and H.F. Walker. Choosing the forcing terms in an inexact Newton method. *SIAM J. Sci. Comput.*, 17:1064–8275, 1996.

3. C. Jablonowski. *Adaptive Grids in Weather and Climate Modeling*. PhD thesis, University of Michigan, Ann Arbor, MI, 2004.
4. R. Jakob-Chien, J.J. Hack, and D.L. Williamson. Spectral transform solutions to the shallow water test set. *J. Comput. Phys.*, 119:164–187, 1995.
5. S. Osher and S. Chakravarthy. Upwind schemes and boundary conditions with applications to Euler equations in general geometries. *J. Comput. Phys.*, 50:447–481, 1983.
6. S. Osher and F. Solomon. Upwind difference schemes for hyperbolic systems of conservation laws. *Math. Comput.*, 38:339–374, 1982.
7. C. Ronchi, R. Iacono, and P. Paolucci. The cubed sphere: A new method for the solution of partial differential equations in spherical geometry. *J. Comput. Phys.*, 124:93–114, 1996.
8. R. Sadourny. Conservative finite-difference approximations of the primitive equations on quasi-uniform spherical grids. *Mon. Wea. Rev.*, 100:211–224, 1972.
9. A. Toselli and O. Widlund. *Domain Decomposition Methods – Algorithms and Theory*. Springer, Berlin, 2005.
10. D.L. Williamson, J.B. Drake, J.J. Hack, R. Jakob, and P.N. Swarztrauber. A standard test set for numerical approximations to the shallow water equations in spherical geometry. *J. Comput. Phys.*, 102:211–224, 1992.
11. C. Yang and X.-C. Cai. A parallel well-balanced finite volume method for shallow water equations with topography on the cubed-sphere. *J. Comput. Appl. Math.*, 2010. to appear.
12. C. Yang, J. Cao, and X.-C. Cai. A fully implicit domain decomposition algorithm for shallow water equations on the cubed-sphere. *SIAM J. Sci. Comput.*, 32:418–438, 2010.