A Fully Implicit Compressible Euler Solver for Atmospheric Flows *

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

[1] Institute of Software, Chinese Academy of Sciences, Beijing 100190, P. R. China, yangchao@iscas.ac.cn
[2] Department of Computer Science, University of Colorado at Boulder, Boulder, CO 80309, USA, chao.yang@colorado.edu, cai@cs.colorado.edu

1 Introduction

Numerical methods for global atmospheric modeling have been widely studied in many literatures [5, 7, 9]. It is well-recognized that the global atmospheric flows can be modeled by fully compressible Euler equations with almost no approximations necessary [7]. However, due to the multi-scale nature of the global atmosphere and the high cost of computation, other simplified models have been favorably used in most community codes.

There are two main difficulties in using fully compressible Euler equations in atmospheric flow simulations. One is that the fast waves in the equations lead to very restrictive stability conditions for explicit time-stepping methods; see, e.g., [11]. Another difficulty is that the flow is nearly compressible and the low Mach number results in large numerical dissipation errors in many classical numerical schemes.

To deal with the fast acoustic and inertio-gravity waves in the fully compressible model, we develop a fully implicit method so that the time step size is no longer constrained by the stability condition. And to treat the low-Mach number flow, an improved version of the Advection Upstream Splitting Method (AUSM$^+$-up, [8]) is adapted. This technique has been successfully employed for a shallow water model in [12]. In the fully implicit solver, we use an inexact Newton method to solve the nonlinear system arising at each time step; and the linear Jacobian system for each Newton step is then solved by a Krylov subspace method with an additive Schwarz preconditioner. We show by numerical experiments on a machine with thousands of processors that the parallel Newton-Krylov-Schwarz approach works well for fully compressible atmospheric flows.

* CY was supported in part by NSFC under 61170075 and 91130023, in part by 973 and 863 Programs of China under 2011CB309701 and 2010AA012301. XCC was supported in part by NSF under DMS-0913089 and EAR-0934647.

R. Bank et al. (eds.), *Domain Decomposition Methods in Science and Engineering XX*, Lecture Notes in Computational Science and Engineering 91,
DOI 10.1007/978-3-642-35275-1_81, © Springer-Verlag Berlin Heidelberg 2013

2 Governing Equations

Various formulations of the governing equations for mesoscale atmospheric models can be found in, e.g., [6]. In this paper, we focus on the compressible Euler equations by restricting the study on two dimensions (the $x-z$ plane) and omitting the Coriolis terms. The compressible Euler equations for the atmosphere take the following form

$$\frac{\partial Q}{\partial t}+\frac{\partial F}{\partial x}+\frac{\partial G}{\partial z}+S=0,$$

where

$$Q=\begin{pmatrix}\rho\\ \rho u\\ \rho w\\ \rho\theta\end{pmatrix},F=\begin{pmatrix}\rho u\\ \rho u^2+p\\ \rho uw\\ \rho u\theta\end{pmatrix},G=\begin{pmatrix}\rho w\\ \rho wu\\ \rho w^2+p\\ \rho w\theta\end{pmatrix},S=\begin{pmatrix}0\\ 0\\ \rho g\\ 0\end{pmatrix}, \tag{1}$$

where $g=9.80665\,\mathrm{m/s^2}$ is the effective gravity on the surface of the Earth. In the equation, the prognostic variables are the density ρ, the velocity (u,w) and the potential temperature θ of the atmosphere. The system is closed with the equation of state

$$p=p_{00}\left(\frac{\rho R\theta}{p_{00}}\right)^{\gamma},$$

where $p_{00}=1013.25\,\mathrm{hPa}$ is the reference pressure on the surface, $R=287.04\,\mathrm{J/(kg\cdot K)}$ is the gas constant for dry air and $\gamma=1.4$. For the sake of brevity, we assume the computational domain Ω is a rectangle and the boundary conditions are given in Sect. 5. In some cases, a physical dissipation is added to the left-hand-side of the momentum and velocity equations. The dissipation term is $-\nabla\cdot(\nu\rho\nabla\phi)$ for $\phi=u$, w, and θ.

To recover the hydrostatic solution from the equation, instead of using (1) directly, the following shifted system is often preferred [6, 11]:

$$Q=\begin{pmatrix}\rho'\\ \rho u\\ \rho w\\ (\rho\theta)'\end{pmatrix},F=\begin{pmatrix}\rho u\\ \rho u^2+p'\\ \rho uw\\ \rho u\theta\end{pmatrix},G=\begin{pmatrix}\rho w\\ \rho wu\\ \rho w^2+p'\\ \rho w\theta\end{pmatrix},S=\begin{pmatrix}0\\ 0\\ \rho' g\\ 0\end{pmatrix} \tag{2}$$

where

$$\rho'=\rho-\bar{\rho},\quad p'=p-\bar{p},\quad (\rho\theta)'=\rho\theta-\bar{\rho}\bar{\theta}$$

and the variables with 'bar' satisfy the hydrostatic condition $\frac{\partial\bar{p}}{\partial z}=-\bar{\rho}g$ and $\bar{\theta}$ is obtained from the equation of state. It is clear that the flux Jacobian of the shifted system (2) in each spatial direction is, respectively,

$$\frac{\partial F}{\partial Q}=\begin{pmatrix}0&1&0&0\\ -u^2&2u&0&c^2/\theta\\ -uw&w&u&0\\ -u\theta&\theta&0&u\end{pmatrix},\quad \frac{\partial G}{\partial Q}=\begin{pmatrix}0&0&1&0\\ -wu&w&u&0\\ -w^2&0&2w&c^2/\theta\\ -w\theta&0&\theta&w\end{pmatrix},$$

where $c=\sqrt{\gamma p/\rho}$ is the sound speed.

3 Discretizations

Suppose the computational domain is covered by a uniform rectangular $N_x \times N_z$ mesh. Mesh cell $\mathscr{C}_{ij}$ is centered at (x_i, z_j), for $i = 1, \ldots, N_x$ and $j = 1, \ldots, N_z$, with mesh size $\Delta x \times \Delta z$. The solution in cell $\mathscr{C}_{ij}$ at time t is approximated as

$$Q_{ij} \approx \frac{1}{\Delta x \Delta z} \int_{z_j - \Delta z/2}^{z_j + \Delta z/2} \int_{x_i - \Delta x/2}^{x_i + \Delta x/2} Q(x, z, t)\, dx dz.$$

We employ a cell-centered finite volume method for the spatial discretization of the compressible Euler equations (2). Integrating (2) over $\mathscr{C}_{ij}$ leads to the following semi-discrete system

$$\frac{\partial Q_{i,j}}{\partial t} + \frac{F_{i+1/2,j} - F_{i-1/2,j}}{\Delta x} + \frac{G_{i,j+1/2} - G_{i,j-1/2}}{\Delta z} + S(Q_{i,j}) = 0,$$

where the numerical fluxes of F and G are averaged on the edges of each mesh cell.

To calculate the numerical fluxes on cell edges, we first employ a piecewise linear formulation to reconstruct constant states in both left and right direction, i.e.,

$$\begin{aligned}
Q^-_{i+\frac{1}{2},j} &= Q_{ij} + \frac{1}{4}(Q_{i+1,j} - Q_{i-1,j}), \quad & Q^+_{i-\frac{1}{2},j} &= Q_{ij} - \frac{1}{4}(Q_{i+1,j} - Q_{i-1,j}), \\
Q^-_{i,j+\frac{1}{2}} &= Q_{ij} + \frac{1}{4}(Q_{i,j+1} - Q_{i,j-1}), \quad & Q^+_{i,j-\frac{1}{2}} &= Q_{ij} - \frac{1}{4}(Q_{i,j+1} - Q_{i,j-1}).
\end{aligned}$$

Then we use an improved version of the Advection Upstream Splitting Method (AUSM$^+$-up, [8]) to approximate the numerical fluxes based on the reconstructed states. The basic idea of AUSM$^+$-up scheme is to split the flux into two parts, e.g.,

$$F = F^{(c)} + F^{(p)},$$

where the convective flux $F^{(c)} = \rho u(1, u, w, \theta)^T$ and the pressure flux $F^{(p)} = (0, p', 0, 0)^T$ are estimated separately, both in an upwinded manner. For instance, denote the left and right reconstructed states for the prognostic variables on an edge of a mesh cell as $(\rho_-, u_-, w_-, \theta_-)$ and $(\rho_+, u_+, w_+, \theta_+)$, the pressure flux is approximated by $F^{(p)} \approx (0, \widetilde{p}', 0, 0)^T$, where

$$\widetilde{p}' = \mathscr{P}_5^+(M_-)p'_- + \mathscr{P}_5^-(M_+)p'_+ - (3/2)\mathscr{P}_5^+(M_-)\mathscr{P}_5^-(M_+)\widetilde{\rho}\,\widetilde{c}(u_+ - u_-),$$

and

$$\begin{aligned}
&\widetilde{\rho} = (\rho_- + \rho_+)/2, \quad \widetilde{c} = (\sqrt{\gamma p_+/\rho_+} + \sqrt{\gamma p_-/\rho_-})/2, \quad p'_\pm = p_\pm - \bar{p}, \\
&\mathscr{P}_5^\pm(M) = \begin{cases} (1 \pm \mathrm{sign}(M))/2, & \text{if } |M| \geq 1, \\ \mathscr{M}_2^\pm(M)\left[(\pm 2 - M) \mp 3M\mathscr{M}_2^\mp(M)\right], & \text{otherwise}, \end{cases} \\
&\mathscr{M}_2^\pm(M) = (M \pm 1)^2/4, \quad M_\pm = u_\pm/\widetilde{c}.
\end{aligned}$$

More details can be found in [8].

For the temporal integration, instead of using explicit methods that suffer from severe stability restriction on the time step size, we employ a fully implicit method. Given a semi-discrete system

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

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

$$\frac{1}{2\Delta t}\left(3Q^{(k)} - 4Q^{(k-1)} + Q^{(k-2)}\right) + \mathscr{L}(Q^{(k)}) = 0.$$

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

4 Newton-Krylov-Schwarz Solver

The fully implicit method leads to a large sparse nonlinear algebraic system at each time step. In this study, we use the Newton-Krylov-Schwarz (NKS) algorithm as the nonlinear solver. Given a nonlinear system $\mathscr{F}(X) = 0$, an inexact Newton method is used to solve the system in the outer loop of the NKS approach. Let X_n be the approximate solution for the n-th Newton iterate, we find the next solution X_{n+1} as

$$X_{n+1} = X_n + \lambda_n s_n, \quad n = 0, 1, \ldots$$

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

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

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

$$\|r_n\| \leq \eta \|\mathscr{F}(X_n)\|,$$

where $\eta > 0$ is the nonlinear forcing term that has been set to be a fixed value $\eta = 1.0 \times 10^{-6}$ in our test. A multi-coloring finite difference method [4] is used to form the Jacobian J_n in the calculation. To achieve uniform residual error at each time step, we use the same adaptive stopping conditions as in [13].

Given the computational domain Ω, we first decompose it into non-overlapping subdomains Ω_k, $k = 1, \ldots, np$, where np is the number of subdomains and also the number of processor cores. Then each subdomain Ω_k is extended to Ω_k^δ within Ω and the number of overlapping mesh layers between subdomains is δ. For the overlapping domain decomposition, a preconditioner M^{-1} is then constructed using the one-level restricted additive Schwarz (RAS, [2]) method defined as follows

$$M^{-1} = \sum_{k=1}^{np} (R_k^0)^T (J_n)_k^{-1} R_k^\delta.$$

Here $(J_n)_k$ is the Jacobian matrix defined on subdomain Ω_k^δ and R_k^δ and $(R_k^0)^T$ are restriction and prolongation operators respectively. Given a solution vector defined on Ω, R_k^δ restricts the vector to the overlapping subdomain Ω_k^δ while $(R_k^0)^T$ prolongates the restricted vector back to the whole domain Ω by putting zeros not only outside Ω_k^δ but also within $\Omega_k^\delta \setminus \Omega_k$. In the implementation of the NKS solver, we use a point-block ordering for both the unknowns and the nonlinear equations, resulting in Jacobian matrices with 4×4-block entries. A point-block version of sparse LU factorization is then used to solve the subdomain problems.

5 Numerical Results

An IBM BlueGene/L supercomputer with 4,096 nodes is used to conduct our numerical tests. Each node of the computer has a dual-core IBM PowerPC 440 processor running at 700 MHz and 512 MB local memory. We implement the NKS algorithm based on the Portable, Extensible Toolkits for Scientific computations (PETSc, [1]) library. In the numerical tests, the overlapping factor in the NKS solver is fixed at $\delta = 2$.

We study a test case describing a rising thermal bubble that is similar to those studied in [3] and [10]. The computational domain is

$$\Omega = \left\{ (x,z) \middle| x \in [-10.0\,\text{km}, 10.0\,\text{km}], z \in [0, 10.0\,\text{km}] \right\},$$

which is assumed to be horizontally periodic with rigid walls (zero normal velocity, i.e., $w = 0$ here) at the bottom and top boundaries. The initial condition for the problem is obtained from a hydrostatic state with $u = w = 0$ and $\bar{\theta} = 300$ K by adding a perturbation

$$\Delta\theta = \begin{cases} 2.0\cos(0.5\pi L)\,\text{K} & \text{if } L \le 1.0, \\ 0.0\,\text{K} & \text{otherwise,} \end{cases}$$

where

$$L = \sqrt{\left(\frac{x - 0.0\,\text{km}}{2.0\,\text{km}}\right)^2 + \left(\frac{z - 2.0\,\text{km}}{2.0\,\text{km}}\right)^2}.$$

A physical dissipation $\nu = 15.0\,\text{m}^2/\text{s}$ is employed in the calculation. The results on a $1{,}000 \times 500$ mesh using the fully implicit method with $\Delta t = 2.0$ s are provided in Fig. 1. We find that the results are in agreement with those provided in several publications; see, e.g., [3, 10] and [6].

To investigate the performance of the preconditioner, we run a fixed size problem on a $1{,}920 \times 960$ mesh for 50 time steps with $\Delta t = 2.0$ s by using gradually doubled numbers of processor cores (np). The results on the averaged number of Newton and GMRES iterations per time step are provided in Fig. 2, from where we observe that

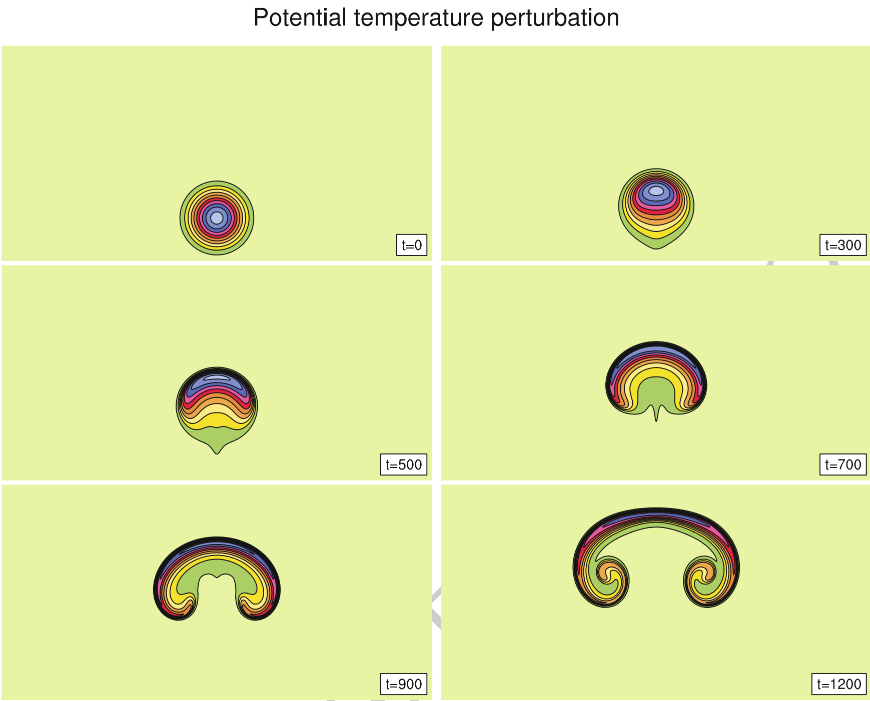

Fig. 1. Contour plots of the potential temperature perturbation (contour interval: 0.2 K)

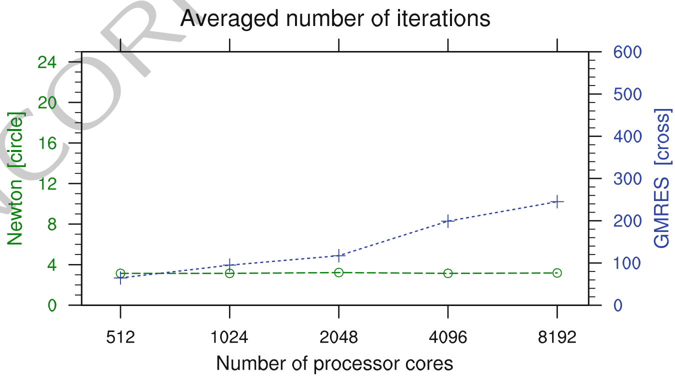

Fig. 2. Averaged numbers of Newton and GMRES iterations per time step

the number of Newton iterations is not sensitive to np but the number of GMRES iterations needed for each time step increases as np increases. The total compute time and the parallel scalability are provided in Fig. 3, which clearly shows that as more processors are used for the fixed size problem, the total compute time is reduced accordingly and the parallel scalability from 512 to $8,192$ processor cores is nearly

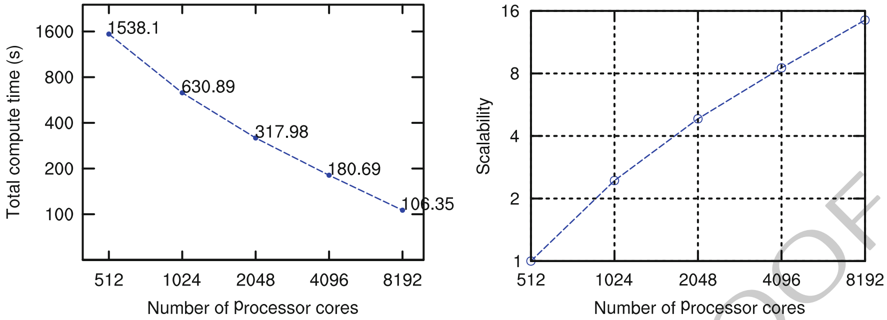

Fig. 3. Total compute time (*left*) and parallel scalability (*right*) results

optimal, with the parallel efficiency reaching 90.38 %. Because of the page limit, we only present a one-level restricted addtive Schwarz method for the compressible Euler problem and only provide some preliminary results in this paper. More advanced algorithms such as multilevel hybrid Schwarz methods will be investigated in a forthcoming paper and more numerical experiments will be carried out in it.

Bibliography

[1] S. Balay, K. Buschelman, V. Eijkhout, W. Gropp, D. Kaushik, M. Knepley, L. C. McInnes, B. Smith, and H. Zhang. *PETSc Users Manual*. Argonne National Laboratory, 2010.

[2] X.-C. Cai and M. Sarkis. A restricted additive Schwarz preconditioner for general sparse linear systems. *SIAM J. Sci. Comput.*, 21:792–797, 1999.

[3] R. L. Carpenter Jr., K. K. Droegemeier, P. R. Woodward, and C. E. Hane. Application of the piecewise parabolic method (PPM) to meteorological modeling. *Mon. Wea. Rev.*, 118:586–612, 1990.

[4] T. F. Coleman and J. J. Moré. Estimation of sparse Jacobian matrices and graph coloring problems. *SIAM J. Numer. Anal.*, 20:187–209, 1983.

[5] K. J. Evans, M. A. Taylor, and J. B. Drake. Accuracy analysis of a spectral element atmospheric model using a fully implicit solution framework. *Mon. Wea. Rev.*, 138:3333–3341, 2010.

[6] F. X. Giraldo and M. Restelli. A study of spectral element and discontinuous Galerkin methods for the Navier-Stokes equations in nonhydrostatic mesoscale atmospheric modeling: Equation sets and test cases. *J. Comput. Phys.*, 227: 3849–3877, 2008.

[7] P. H. Lauritzen, C. Jablonowski, M. A. Taylor, and R. D. Nair, editors. *Numerical Techniques for Global Atmospheric Models*. Springer, 2011.

[8] M.-S. Liou. A sequel to AUSM, part II: AUSM+-up for all speeds. *J. Comput. Phys.*, 214:137–170, 2006.

[9] R. D. Nair, H.-W. Choi, and H. M. Tufo. Computational aspects of a scalable high-order discontinuous Galerkin atmospheric dynamical core. *Comp. Fluids*, 38:309–319, 2009.

[10] A. Robert. Bubble convection experiments with a semi-implicit formulation of the Euler equations. *J. Atmos. Sci.*, 50:1865–1873, 1993.

[11] A. St-Cyr and D. Neckels. A fully implicit Jacobian-free high-order discontinuous Galerkin mesoscale flow solver. In *ICCS 2009, part II, vol. 5545 of Lecture Notes in Computer Science*, pages 243–252. Springer-Verlag, 2009.

[12] P. A. Ullrich, C. Jablonowski, and B. van Leer. High-order finite-volume methods for the shallow-water equations on the sphere. *J. Comput. Phys.*, 229: 6104–6134, 2010.

[13] C. Yang and X.-C. Cai. Parallel multilevel methods for implicit solution of shallow water equations with nonsmooth topography on the cubed-sphere. *J. Comput. Phys.*, 230:2523–2539, 2011.