Title:

SEMI-LAGRANGIAN SHALLOW WATER MODELING ON THE CM-5

Author(s):

B. T. Nadiga, L. G. Margolin, P. K. Smolarkiewicz

Submitted to:

Parallel CFD '95
Pasadena, CA
July 1995

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Los Alamos
NATIONAL LABORATORY

WATER

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the University of California for the U.S. Department of Energy under contract W-7405-ENG-36. By acceptance of this article, the publisher recognizes that the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. The Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

Semi-Lagrangian Shallow Water Modeling on the CM-5

B. T. Nadiga^a, L. G. Margolin^a, and P. K. Smolarkiewicz^b

^aLos Alamos National Lab., MS-B258, Los Alamos, NM 87545

^bNational Center for Atmospheric Research, Boulder, CO 80307

We discuss the parallel implementation of a semi-Lagrangian shallow-water model on the massively parallel Connection Machine CM-5. The four important issues we address in this article are (i) two alternative formulations of the elliptic problem and their relative efficiencies, (ii) the performance of two successive orders of a generalized conjugate residual elliptic solver, (iii) the time spent in unstructured communication—an unavoidable feature of semi-Lagrangian schemes, and (iv) the scalability of the algorithm.

1. The Physical Problem: A SubTropical-SubPolar Gyre System

We model the wind-driven circulation in a closed rectangular midlatitude ocean basin. The model consists of a dynamically active shallow surface layer overlying an infinitely deep lower layer and the chosen wind forcing, acting on the surface layer, results in the formation of a double gyre. The vertically-averaged circulation of the surface layer can then be modelled using the hydrostatic shallow water equations [1]:

$$\frac{\partial \eta}{\partial t} + \nabla \cdot (\eta \mathbf{u}) = 0, \quad (1a)$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -g \frac{\Delta \rho}{\rho} \nabla \eta + f \hat{\mathbf{z}} \times \mathbf{u} + \sigma_x(y) \hat{\mathbf{x}}, \quad (1b)$$

where η is the fluid thickness and $\mathbf{u}$ is the horizontal velocity vector. The domain size is 1000 kms in the east-west direction and 2000 kms in the north-south direction; the initial undisturbed depth of the upper layer is 500 meters. The reduced gravity parameter, $g \Delta \rho / \rho (\equiv g')$, where $\Delta \rho$ is the difference in densities between the upper and lower layers, is 0.031, the wind stress profile $\sigma_x(y) = -A \cos(2\pi y/D)$, where A corresponds to a wind stress amplitude of 0.1 N/m². The simulations are carried out on a beta plane, i.e., $f = f_0 + \beta y$, with $f_0 = 5 \cdot 10^{-5} \text{ s}^{-1}$ and $\beta = 2 \cdot 10^{-11} \text{ m}^{-1} \text{ s}^{-1}$. The model ocean is started from rest and the performance comparisons of the various implementations are done during the spin up period consisting of the first 30 days (at the end of which all implementations give identical results).

2. The Semi-Lagrangian Scheme

To better illustrate the issues encountered in our parallel implementation, we briefly describe the semi-Lagrangian technique. Equations 1 have two different types of terms: advective or streaming terms, and the remaining terms which represent forces acting on Lagrangian fluid particles. To better treat the different natures of the two types of terms, the semi-Lagrangian technique employs separate discretizations for them—integrating the forces along a parcel trajectory (the Lagrangian aspect) and evaluating the advection terms along a residual contour [2]. This differential treatment is in contrast to the single

form of discretization for any term, regardless of its physical nature, employed in an Eulerian scheme.

Equations 1 may be rewritten symbolically in the form

$$\frac{d\psi}{dt} = R, \quad (2)$$

where $\psi = \psi(x, t)$ is a fluid variable (η or u), $R = R(x, t)$ is the corresponding force expressible in terms of ψ and its derivatives, and $d/dt \equiv \partial/\partial t + u \cdot \nabla$ is the material time derivative. Integration of (2) in time leads to

$$\psi(x, t) = \psi(x_o, t_o) + \int_T R dt, \quad (3)$$

where T is the parcel trajectory connecting the arrival point (x, t) , assumed to coincide with a grid point, to (x_o, t_o) , the corresponding departure point, not necessarily a grid point. We compute the departure point x_o corresponding to the arrival point x to second-order accuracy by two iterations of

$$x_o = x + \int_{t_1}^{t_o} u(x, t) dt. \quad (4)$$

Since the departure point will not in general coincide with a grid point, an interpolation procedure has to be invoked in order to determine the discrete field value $\psi(x_o, t_o)$ in (3). This interpolation is equivalent to solving a constant coefficient advection problem locally at the departure point, with the constant coefficient being proportional to the separation of the departure point x_o from its nearest grid point x_* [2]. Thus (3) may be rewritten as

$$\psi(x, t) + O(\mathcal{A}) = \mathcal{A}(\psi(x_*, t_o), \Upsilon_o) + \int_T R dt, \quad (5)$$

where $\mathcal{A}$ is a constant coefficient Eulerian advection operator, $O(\mathcal{A})$ is the truncation error associated with $\mathcal{A}$, and $\Upsilon = (x_* - x_o)/\Delta x$ is the nondimensional separation of the departure point x_o from its nearest grid point x_* , and represents the Courant number for the advection problem. In view of the constancy of the advective velocity, the multi-dimensional advection operator can be replaced *exactly* by a combination of one-dimensional advection operators (*i.e.*, there are no splitting errors) and $\mathcal{A}$ can be chosen to be a one-dimensional constant coefficient Eulerian advection operator. Thus, the entire semi-Lagrangian flow model can be built upon a one-dimensional Eulerian advection scheme, while retaining the formal accuracy of its constant-coefficient limit. We use the second-order flux-corrected Lax-Wendroff scheme for $\mathcal{A}$ (*e.g.*, see [2] and references therein).

Semi-Lagrangian techniques are usually found to be more accurate than Eulerian techniques, but are not formulated in terms of a detailed balance of fluxes and so are not exactly conservative. Another recognized advantage of the semi-Lagrangian technique is the relaxation of the Courant-Friedrichs-Lewy (CFL) stability criterion (based on the velocity) to a less restrictive condition based on the spatial derivatives of velocity.

3. Semi-Lagrangian Discretization of the Shallow Water Equations

After adding provision to absorb waves at the boundaries without reflection (restoring boundaries), (1a) may be rewritten as

$$\frac{d\eta}{dt} = -\eta \nabla \cdot u - \frac{\eta - \eta_a}{\tau(x)}. \quad (6)$$

A discretization of (6) centered on the midpoint of T leads to

$$\frac{\eta - \eta_o}{\Delta t} = -\frac{1}{2} \{ \eta_o \nabla \cdot \mathbf{u} + \eta [\nabla \cdot \mathbf{u}]_o \} - \frac{1}{2} \left\{ \left[\frac{\eta - \eta_a}{\tau} \right]_o + \frac{\eta - \eta_a}{\tau} \right\}. \quad (7)$$

where we have used harmonic averaging in the midpoint evaluation of the first term on the right hand side of (6) to linearize the equation in the as yet unknown values of the fluid variables at the arrival point. ψ_o is a short-hand notation for $\psi(\mathbf{x}_o, t_o)$, and is estimated using the second-order procedure $\mathcal{A}(\psi(\mathbf{x}_*, t_o), \Upsilon_o)$ of (5). The momentum equation is, again with provision for restoring boundaries,

$$\frac{d\mathbf{u}}{dt} = -g' \nabla \eta + f \hat{\mathbf{z}} \times \mathbf{u} + \sigma_x(y) \hat{\mathbf{x}} - \frac{\mathbf{u} - \mathbf{u}_a}{\tau}, \quad (8)$$

and a similar discretization of it leads to

$$\begin{aligned} \frac{\mathbf{u} - \mathbf{u}_o}{\Delta t} = & -\frac{1}{2} g' (\nabla \eta + [\nabla \eta]_o) + \frac{1}{2} (f \hat{\mathbf{z}} \times \mathbf{u} + [f \hat{\mathbf{z}} \times \mathbf{u}]_o) \\ & + \frac{1}{2} (\sigma_x(y) + [\sigma_x(y)]_o) \hat{\mathbf{x}} - \frac{1}{2} \left(\frac{\mathbf{u} - \mathbf{u}_a}{\tau} + \left[\frac{\mathbf{u} - \mathbf{u}_a}{\tau} \right]_o \right). \end{aligned} \quad (9)$$

Equations (7) and (9) (with centered differencing for ∇), form the basis of our spatially and temporally second-order accurate, two-time level, semi-Lagrangian model of (1).

3a. Implicit Free Surface: Velocity Formulation

Equation (7) can be solved explicitly for the depth field at the arrival point η in terms of the divergence of the velocity field as

$$\eta = \alpha - \beta \nabla \cdot \mathbf{u}, \quad (10)$$

$$\alpha = \left\{ \left[\eta - \frac{\Delta t}{2\tau} (\eta - \eta_a) \right]_o + \frac{\Delta t}{2\tau} \eta_a \right\} \{ 1 + \Delta t/2 ([\nabla \cdot \mathbf{u}]_o + 1/\tau) \}^{-1},$$

$$\beta = \frac{\Delta t}{2} \eta_o \{ 1 + \Delta t/2 ([\nabla \cdot \mathbf{u}]_o + 1/\tau) \}^{-1}.$$

Further, (9) can be simplified to express $\mathbf{u}$ in terms of $\nabla \eta$ as follows:

$$\mathbf{A} \mathbf{u} = \tilde{\mathbf{u}} - \frac{\Delta t}{2} g' \nabla \eta, \quad (11)$$

$$\mathbf{A} = \begin{pmatrix} 1 + \frac{\Delta t}{2\tau} & -f \frac{\Delta t}{2} \\ f \frac{\Delta t}{2} & 1 + \frac{\Delta t}{2\tau} \end{pmatrix}, \quad \tilde{\mathbf{u}} = \left[\mathbf{u} + \frac{\Delta t}{2} \left(-g' \nabla \eta + f \hat{\mathbf{z}} \times \mathbf{u} + \sigma_x(y) \hat{\mathbf{x}} - \frac{\mathbf{u} - \mathbf{u}_a}{\tau} \right) \right]_o + \frac{\Delta t}{2} \left(\sigma_x(y) \hat{\mathbf{x}} + \frac{\mathbf{u}_a}{\tau} \right).$$

We insert (10) into (11) to obtain a variable-coefficient linear elliptic equation for $\mathbf{u}$:

$$\mathcal{L}(\mathbf{u}) = \frac{\Delta t}{2} g' \nabla (\beta \nabla \cdot \mathbf{u}) - \mathbf{A} \mathbf{u} = \frac{\Delta t}{2} g' \nabla \alpha - \tilde{\mathbf{u}}, \quad (12)$$

where the right-hand-side and the coefficients $\mathbf{A}$ and β are all known.

Equations (12) and (10) constitute our implicit free surface semi-Lagrangian shallow water model formulated in terms of an elliptic equation for the velocity field. This formulation is particularly useful when dispersion is introduced at the next higher level of approximation (*e.g.*, the Green-Naghdi equations [3]) of the shallow fluid equations. Here the introduction of singular dispersive terms in the velocity equations necessitates an implicit treatment of those terms in order to allow an affordably large time step. We will discuss these higher-order formulations elsewhere.

3b. Implicit Free Surface: Depth Formulation

Alternately, (11) may be solved analytically for the velocity $\mathbf{u}$ in terms of $\nabla\eta$:

$$\mathbf{u} = \mathbf{A}^{-1} \left(\tilde{\mathbf{u}} - \frac{\Delta t}{2} g' \nabla \eta \right). \quad (13)$$

Then, inserting (13) in (10), we again obtain a variable-coefficient linear elliptic equation, but now for η :

$$\mathcal{L}(\eta) = \frac{\Delta t}{2} g' \beta \nabla \cdot [\mathbf{A}^{-1} \nabla \eta] - \eta = \beta \nabla \cdot [\mathbf{A}^{-1} \tilde{\mathbf{u}}] - \alpha. \quad (14)$$

Equations (14) and (11) constitute our semi-Lagrangian shallow water model, formulated now in terms of an elliptic equation for the depth field. We note that this formulation does not extend to the next higher level of approximation of the shallow fluid equations since the higher-order dispersive terms that would then appear in the momentum equation do not allow an explicit solution of $\mathbf{u}$ in terms of $\nabla\eta$, as we have done here in (13).

4. Implementation

We have implemented both formulations (Eqs. (12) & (10) and Eqs. (14) & (11)) on the massively parallel Connection Machine CM-5 at Los Alamos National Laboratory. Each node of the CM-5 has four vector units and we use the data parallel mode of computation, programming in CM Fortran. The problem size is fixed at 256 grid points in both the latitude and longitude directions unless otherwise specified and we use partitions of three different sizes—either 32 or 64 or 128 nodes. In all implementations, we use the default array layout with both the latitude and longitude axes declared parallel. (Performance was severely degraded by declaring one of the axes serial.) We base our computations at the arrival point to avoid difficulties of balancing the load. All computations are performed in single precision 32 bit mode. We have verified the sufficiency of single precision calculations by comparisons with double precision computations in steady flow cases. Care was taken in the coding to make the parallel code blocks as long as possible by separating out the statements that required communications from those that did not. We did not use the optimization flag on the compiler.

Table 1 shows where the program spends most of the time. The timings are given for the implementation UC4 discussed later. All times reported in this article were obtained using the `'-cmprofile'` option of the CM Fortran compiler and analysed using `'prism'`. In addition to the total time spent by the procedures on the nodes, we have considered the usage of four different resources on the CM-5 to characterise performance.

1. Node CPU—the time the program spent in processing on any of the nodes.
2. NEWS—the time spent in structured grid communication (*e.g.*, `cshift`).
3. Send/Get—the time spent in router communication (*e.g.*, indirect addressing).
4. Reduction—the time spent in data reductions (*e.g.*, global sum/maximum/minimum).

Procedure	Node Total	Node CPU	Communication		
			NEWS	Send/Get	Reduction
Total	275.3	169.8	63.7	31.7	10.1
$\mathcal{A}(\psi(x_*, t_0), \Upsilon_0)$	149.7	100.8	17.2	31.7	0.0
$\mathcal{L}(\psi) = \mathcal{R}$	109.1	56.8	42.2	0.0	10.1

Table 1: The time spent (in seconds) in the advection of various quantities and the solution of the elliptic equation for a 30 day run on 32 nodes.

There are seven calls to the advection routine and one call to the elliptic solver. The program spends about 50% of the time in advection or interpolation of various quantities and about 40% of the time in the solution of the elliptic equation. Below, we consider in detail the performance aspects of the different implementations. Unless otherwise specified, all performance data presented in this article are time spent in seconds on a 32 node partition for a 30 day run using a 256×256 mesh.

4a. Unstructured Communication

The time step in our semi-Lagrangian simulations is determined by restricting $\|\frac{\Delta u}{\Delta x}\| \Delta t$ (where the norm used is the L_∞ norm) to be less than 0.5. This condition is much less restrictive than the CFL condition characteristic of Eulerian schemes and allows much larger time steps. The price to be paid for this is unstructured communication or in other words indirect addressing. Owing to architecture of the CM-5, unstructured communication is much slower than the regular shift type of communication. The cost of unstructured communication is reflected in the Send/Get measure. We consider four different ways of implementing the unstructured communication:

	Node Total	Node CPU	Communication		
			NEWS	Send/Get	Reduction
UC1	338.4	168.9	63.1	96.6	9.7
UC2	321.6	162.1	64.1	85.6	9.8
UC3	396.6	162.3	65.1	159.5	9.8
UC4	275.5	169.9	63.6	31.8	10.1

Table 2: Usage of different resources by the four implementations of unstructured communication.

1. UC1 uses the *forall* construct of CM Fortran and is the closest to the usual Fortran77.
2. UC2 computes a trace for the unstructured communication pattern once the departure point calculations are done and then repeatedly uses the trace until the departure point changes.
3. UC3 is similar to UC2, but differs by computing the trace in a manner such that all get operations are performed using send operations.
4. UC4 computes a trace as in UC2 and in addition, whenever unstructured communication is to be done, packs four single precision real fields into one double precision

complex field. This packing (and consequent unpacking) requires no copying or movement of data, and is achieved by maintaining two different front end array descriptors describing the same parallel memory area.

Methods UC2, UC3, and UC4 make use of CMSSL communication routines. In the model implementations using the four methods above, we formulate the elliptic equation in terms of velocity and solve the elliptic equation using GCR(2) (see later).

The only significant difference in the performance data for these four cases is in the time spent in unstructured communication. Among the first three implementations, UC2 takes the least time and hence it was the method of choice for UC4. (We note that the method of packing of UC4 could have been used in conjunction with any of the first three methods.) The reduction in the unstructured communication time in UC4 is related to the nature of the semi-Lagrangian scheme. As indicated in section 2, the heart of a semi-Lagrangian scheme is a 1D Eulerian advection operator. For our 2D problem, this results in three 1D advections in one direction followed by a combining 1D advection in the other direction. With a nine-point stencil and arrival point based computation, there are thus three sets of data movement, with each set of movement consisting of three field values. In UC4, the movement of three field values has been compressed into one movement by packing three single-precision reals into one double-precision complex and this is reflected in a (close to a) factor of three reduction in the time spent in unstructured communication between UC2 and UC4. (In fact four single precision reals can be compressed into one double precision complex, and so 25% of the data movement is wasted.) Since in UC4, unstructured communication is no more a bottleneck and takes only about 50% of the time spent in grid communications, we consider the UC4 method of unstructured communication satisfactory and only use that in all future comparisons. We note that if the computations were to be done in double precision, there would only be a 33% reduction in the Send/Get time in using UC4 instead of UC2. Thus, by packing shorter data types into longer ones and computing a trace which gets used repeatedly till the pattern of communication changes, we effectively minimize time spent in unstructured communications.

4b. Velocity Formulation vs. Depth Formulation

	ϵ	Iters.	Node Total	Elliptic Solver			
				Total	CPU	NEWS	Reduction
VF	10^{-3}	8	228.9	62.5	32.4	24.5	5.7
DF	10^{-3}	5	220.3	51.3	25.0	23.0	3.3
VF	10^{-5}	14	275.5	109.0	56.8	42.9	9.8
DF	10^{-5}	9	255.8	86.8	42.5	38.5	5.9

Table 3: A comparison of the relative efficiencies of formulating the elliptic problem in terms of depth and in terms of the velocity field.

In Table 3, we show the relative performance of the velocity formulation VF (Eqs. (12) and (10)) and the depth formulation DF (Eqs. (14) and (11)) for two cases—first to reduce the residual in the elliptic equation by a factor of 10^{-3} and next to reduce the residual by a factor of 10^{-5} . It is clear that the depth formulation is consistently more efficient but then, only by about 4-8%. Thus by formulating the elliptic problem in terms of

velocity, while the extra overhead is not large, the formulation readily generalizes to the higher level (dispersive) approximations of the shallow water model. In addition, there is a scope for designing better elliptic solvers with the extra degree of freedom in the velocity formulation.

4c. The Elliptic Solver

We briefly outline the generalized conjugate residual algorithm (*e.g.*, see [4]) we use to solve the elliptic equation $\mathcal{L}(\psi) = \mathcal{R}$. For any initial guess ψ^0 , set $\mathbf{p}^0 = \mathbf{r}^0 = \mathcal{L}(\psi^0) - \mathcal{R}$, and then iterate as follows:

For $n = 1$ to convergence do

for $\nu = 0, \dots, k - 1$ do

$$\beta = -\frac{\langle \mathbf{r}^\nu \mathcal{L}(\mathbf{p}^\nu) \rangle}{\langle \mathcal{L}(\mathbf{p}^\nu) \mathcal{L}(\mathbf{p}^\nu) \rangle}$$

$$\psi^{\nu+1} = \psi^\nu + \beta \mathbf{p}^\nu; \quad \mathbf{r}^{\nu+1} = \mathbf{r}^\nu + \beta \mathcal{L}(\mathbf{p}^\nu)$$

$$\text{exit if } \|\mathbf{r}^{\nu+1}\| \leq \varepsilon \|\mathbf{r}^0\|$$

$$\forall_{l=0, \nu} \quad \alpha_l = -\frac{\langle \mathcal{L}(\mathbf{r}^{\nu+1}) \mathcal{L}(\mathbf{p}^l) \rangle}{\langle \mathcal{L}(\mathbf{p}^l) \mathcal{L}(\mathbf{p}^l) \rangle} \quad \text{GCR(k)}$$

$$\mathbf{p}^{\nu+1} = \mathbf{r}^{\nu+1} + \sum_{l=0}^{\nu} \alpha_l \mathbf{p}^l; \quad \mathcal{L}(\mathbf{p}^{\nu+1}) = \mathcal{L}(\mathbf{r}^{\nu+1}) + \sum_{l=0}^{\nu} \alpha_l \mathcal{L}(\mathbf{p}^l)$$

end do

reset $[\psi, \mathbf{r}, \mathbf{p}, \mathcal{L}(\mathbf{p})]^k$ to $[\psi, \mathbf{r}, \mathbf{p}, \mathcal{L}(\mathbf{p})]^0$

end do

Table 4. shows the relative performances of GCR(2) and GCR(3). Though GCR(3) performs exceedingly well in terms of the number of iterations to reduce the residual by a factor ϵ compared to GCR(2), there is no speedup either in terms of CPU time or communication time.

	ϵ	Iters.	Node Total	Elliptic Solver			
				Total	CPU	NEWS	Reduction
GCR(2)	10^{-3}	5	220.3	51.3	25.0	23.0	3.3
GCR(3)	10^{-3}	2	225.7	55.5	27.7	23.9	5.7
GCR(2)	10^{-5}	9	255.8	86.8	42.5	38.5	5.9
GCR(3)	10^{-5}	3	265.6	95.2	48.1	39.8	7.3

Table 4: Comparison of the CPU and communication times for GCR(2) and GCR(3).

4d. Scalability

We consider the scalability of the algorithm both when the problem size is held constant and the number of processors varied and when the problem size is scaled with the

number of processors. From entries 1 and 4 in Table 5, it is clear that the algorithm and its implementation scale perfectly in the latter case, so that all individual times—the CPU time and the times involved in the different types of communication—remain almost exactly the same.

For the case of scaling when the problem size is held constant, if the subgrid length (*i.e.*, the number of grid points per vector unit) is over 512, the CPU part of the algorithm scales perfectly with number of processors, and the unstructured communication does almost so. However, regular grid communications and global reduction operations continue to speedup as the subgrid length increases. In view of this, for our algorithm to make good use of the resources on the CM-5, a subgrid length of 512 or higher is required.

# Nodes	Grid Size	Node Total	Node CPU	Communication		
				NEWS	Send/Get	Reduction
32	256x256	275.3	169.8	63.7	31.7	10.1
32	512x512	963.7	709.7	118.0	112.4	23.6
64	512x512	504.7	352.0	77.9	58.8	16.0
128	512x512	289.1	175.2	69.2	32.0	12.8

Table 5: Scaling data for the VF algorithm using the UC4 communication strategy.

In summary, we have designed an efficient semi-Lagrangian algorithm for shallow fluid flows representative of geophysical motions and shown that its implementation on the Connection Machine is almost fully scalable. Familiarity with the available software utilities on the CM-5 allowed us to significantly improve the computational efficiency of the model.

Acknowledgements

We would like to thank the Advanced Computing Laboratory at the Los Alamos National Laboratory for time and system support on the CM-5. This work was supported in part by the U.S. Department of Energy's CHAMMP program.

References

1. Jiang, S., Jin F-F, and Ghil, M., 1995: Multiple equilibria, periodic, and aperiodic solutions in a wind-driven, double-gyre, shallow-water model. *J. Phys. Ocean.*, 25, 764-786.
2. Smolarkiewicz, P. K., and Pudykiewicz, J. A., 1992: A class of semi-Lagrangian approximations for fluids. *J. Atmos. Sci.*, 49, 2082-2096.
3. Green, A. E., and Naghdi, P. M., 1976: A derivation of equations for wave propagation in water of variable depth. *J. Fluid Mech.*, 78, 237-246.
4. Smolarkiewicz, P. K., and Margolin, L. G., 1994: Variational solver for elliptic problems in atmospheric flows. *Appl. Math. and Comp. Sci.*, 4, 101-125.