Electronic Transactions on Numerical Analysis.
Volume 6, pp. 120-132, December 1997.
Copyright © 1997, Kent State University.
ISSN 1068-9613.

ETNA
Kent State University
etna@mcs.kent.edu

A COMPARISON OF MULTILEVEL ADAPTIVE METHODS FOR HURRICANE TRACK PREDICTION[*]

SCOTT R. FULTON[†]

Abstract. Adaptive multilevel methods are described and tested for the problem of predicting the path of a moving hurricane. The physical model consists of conservation of vorticity in a two-dimensional incompressible fluid; the discrete model uses conservative second-order finite differences. The methods described are the Berger-Oliger (BO) algorithm, with the Poisson problem for the streamfunction solved by standard multigrid techniques, and a full approximation scheme multigrid (MG) algorithm which incorporates more complete interaction between the computational grids. Numerical results are presented demonstrating the conservation properties, convergence, accuracy, and efficiency of the methods. Adaptive mesh refinement produces speedup factors of 10–20 compared to using uniform resolution. Differences between the performance of the BO and MG methods are slight.

Key words. multigrid, multilevel, adaptive, incompressible, hurricane.

AMS subject classifications. 65M50, 65M55, 76C15, 86A10.

1. Introduction.

Many interesting problems in computational fluid dynamics and other fields have solutions with spatial scales which vary substantially from one part of the domain to another. Solving such problems efficiently and accurately requires variable resolution. One way to accomplish this is by superimposing several uniform grids of different mesh sizes and extents, thus successively refining the mesh in the regions of interest. This approach can be naturally combined with multigrid processing [4], providing both fast solvers and truncation error estimates which can be used for self-adaptive mesh refinement. Implementation of such methods for complex problems is nontrivial; much can be learned by applying them to reasonably simple problems.

The model problem treated in this paper is that of predicting the track (path) of a moving hurricane. One can think of a hurricane as a small-scale vortex embedded in a large-scale environmental flow, with an order of magnitude difference in the typical scales of (horizontal) variation. In addition to being a natural candidate for adaptive mesh refinement, this problem can be modeled with simple dynamics (two-dimensional incompressible flow) and has a simple measure of accuracy (location of the vortex center). It is also a problem of substantial real-world importance.

In this paper we detail the performance of two methods for this problem, both of which are based on superimposing uniform grids to achieve nonuniform resolution. Section 2 describes the continuous problem. Details of the numerical methods are given in §3, and §4 presents numerical results.

2. Continuous problem.

While the true physical problem of hurricane motion is quite complex (involving three-dimensional compressible flow, air-sea interaction, radiative effects, phase changes and precipitation), surprisingly reasonable results are obtained using simple two-dimensional incompressible dynamical models. Indeed, some of the best prediction models currently in use are of this type [5, 6, 9]. Here we will use a model based on conservation of vorticity (§2.1). Additional conservation properties are given in §2.2.

2.1. Governing equations.

We use Cartesian coordinates with x and y being east and north, respectively, and consider two-dimensional incompressible flow with velocity compo-

[*] Received May 14, 1997. Accepted for publication July 31, 1997. Communicated by J. Ruge. This research was supported by NSF grant No. ATM-9118966

[†]Department of Mathematics and Computer Science, Clarkson University, Potsdam, NY 13699-5815 (fulton@sun.mcs.clarkson.edu).

nents u and v. The corresponding vorticity $\zeta = \partial v/\partial x - \partial u/\partial y$ satisfies

$$\frac{\partial \zeta}{\partial t} + \frac{\partial(\zeta,\psi)}{\partial(x,y)} + \beta\frac{\partial\psi}{\partial x} = 0. \tag{2.1}$$

Here ψ is the streamfunction satisfying $u = -\partial\psi/\partial y$, $v = \partial\psi/\partial x$; it is related to ζ by the Poisson problem

$$\Delta\psi = \zeta. \tag{2.2}$$

The parameter β (the derivative of Coriolis parameter) measures the effect of the variation of the Earth's rotation with latitude; for simplicity, we treat β as constant (the β-plane approximation). We seek to solve (2.1–2.2) on a rectangular domain $\Omega = [-L_x, L_x] \times [-L_y, L_y]$, with ψ specified on the boundary $\partial\Omega$ and ζ specified where there is inflow; with these conditions the problem is well-posed [8].

2.2. Conservation properties.

If we set $\beta = 0$ and assume there is no flow through the boundary, then from (2.1) the total vorticity

$$Z = \int_\Omega \zeta \tag{2.3}$$

is conserved (independent of time t). Under the same conditions, the total enstrophy

$$E = \int_\Omega \frac{1}{2}\zeta^2 \tag{2.4}$$

and kinetic energy

$$K = \int_\Omega \frac{1}{2}(u^2+v^2) = \int_\Omega \frac{1}{2}\nabla\psi\cdot\nabla\psi \tag{2.5}$$

are also conserved. The conservation of E and K implies that the mean wavenumber of the flow is constant [1], so there is no net cascade of energy between large and small scales.

3. Multilevel adaptive methods.

3.1. Basic discretization.

On a uniform grid with grid points $(x_i, y_j) = (x_0 + ih, y_0 + jh)$, $i = 0, \ldots, M$, $j = 0, \ldots, N$, we can discretize (2.1–2.2) in space by second-order centered finite differences as

$$\frac{d\zeta_{i,j}}{dt} - J^h_{i,j}(\zeta,\psi) + \beta\left(\frac{\psi_{i+1,j} - \psi_{i-1,j}}{2h}\right) = 0, \tag{3.1}$$

$$\Delta^h_{i,j}\psi = \zeta_{i,j}. \tag{3.2}$$

Here $J^h_{i,j}$ is the *Arakawa Jacobian* [1] as given in the Appendix and Δ^h is the usual five-point approximation to the Laplacian. Equations (3.1–3.2) are applied at all interior points; at the boundaries a modified version of (3.1) is applied, with a one-sided second-order difference in the β term and appropriate modifications of $J^h_{i,j}$. The use of the Arakawa Jacobian ensures conservation of discrete analogues of (2.3–2.5), as detailed in [1] and the Appendix.

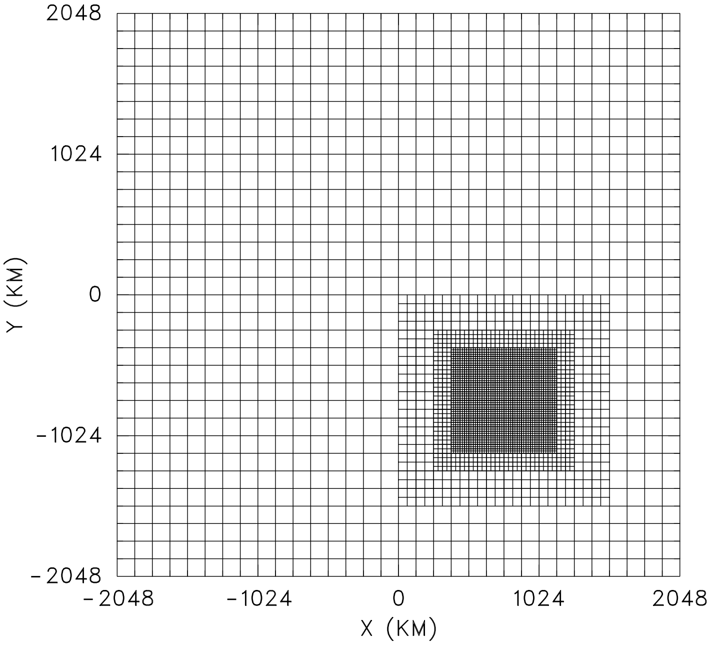

FIG. 3.1. *Example of computational grids, with base grid* G_1 *(*32×32*,* $h = 128\,km$*), patch* G_2 *(*24×24*,* $h = 64\,km$*), patch* G_3 *(*32×32*,* $h = 32\,km$*), and patch* G_4 *(*48×48*,* $h = 16\,km$*)*

3.2. Grid structure. To construct a composite grid with nonuniform resolution we start with a base grid G_1 with mesh size h_1 covering the full domain Ω, and superimpose a sequence of grids G_l, $l = 2,\ldots,n$ with smaller mesh sizes h_l covering successively smaller regions (called *patches*). For simplicity we use the mesh ratio $h_l/h_{l+1} = 2$ and require the patches to be rectangular, aligned (fine-grid boundaries are coarse-grid lines), and strictly nested (fine grid contained in the interior of the next coarser grid). This leads to the structure shown in Fig. 3.1.

Solving (3.2) by a multigrid method on one of the computational grids G_l requires additional coarse grids covering the same domain as G_l. These we denote by $G_l^{(m)}$, $m = 1$, $\ldots$, n_l, and refer to them as the *local coarse grids*. They are indexed such that $G_l = G_l^{(1)}$, and their mesh sizes *increase* with m with mesh ratio 2, so that $G_l^{(m)} \subset G_{l-1}^{(m-1)}$ when $l > 1$ and $m > 1$ (if the boundaries are aligned). A one-dimensional depiction of the resulting grid structure is shown in Fig 3.2.

3.3. Berger-Oliger method. Perhaps the simplest algorithm for solving a time-dependent problem on the grid structure described above is that of Berger and Oliger [3]. With two computational grids (coarse and fine), one step of this algorithm consists of:

1. One step (length Δt) on the coarse grid,
2. Two steps (length $\Delta t/2$) on the fine grid, using boundary values interpolated from the fine grid in space and time,
3. Transfer the fine-grid solution to the coarse grid where they overlap.

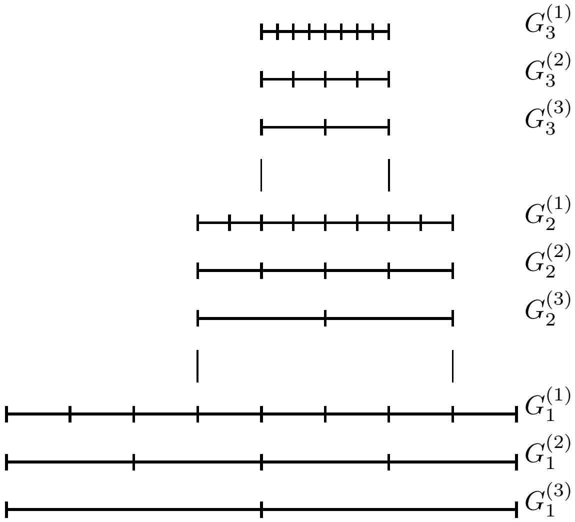

FIG. 3.2. *Example of local coarse grids (one-dimensional depiction)*

Thus, this algorithm is applied separately on each computational grid and uses the fine-grid solution where the grids overlap, with local time stepping (smaller time steps on the fine-grid patch) for efficiency. The recursive generalization to more than two computational grids is immediate.

In implementing this algorithm for the problem at hand, we use the space discretization described above on each computational grid G_l. We use the classical fourth-order Runge–Kutta method in time, so that each time step on each grid involves four stages. Denoting the discrete problem (3.1–3.2) on a single grid G_l with mesh size $h = h_l$ as

$$\frac{d\zeta^h}{dt} = F^h, \qquad \Delta^h \psi^h = \zeta^h, \tag{3.3}$$

each stage of the Runge-Kutta method takes the form

$$\zeta^h(t+\alpha\Delta t) = \zeta^h(t) + \alpha\Delta t \bar{F}^h(t), \qquad \Delta^h\psi^h(t+\alpha\Delta t) = \zeta^h(t+\alpha\Delta t) \tag{3.4}$$

for some α ($0 < \alpha \leq 1$). The term $\bar{F}^h(t)$ represents F^h computed from the solution ψ, ζ from the previous stage, or—in the last stage—a linear combination of the previous terms $\bar{F}^h$. The new ζ^h is computed at all points (interior and boundary), with the boundary values replaced by specified values where there is inflow. The interpolation for fine-grid boundary values is linear in space and time; the fine-to-coarse transfer at the end of the full time step consists of injection for ψ and full weighting for ζ.

Solving the Poisson problem for ψ^h in each stage is accomplished by a multigrid method. The components of this method are standard: point Gauss-Seidel relaxation with red-black ordering, full weighting of residuals, bilinear interpolation of corrections, and a full multigrid (FMG) control algorithm with one $V(1,1)$-cycle per level and bicubic FMG interpolation. Note that this method uses only the local coarse grids $G_l^{(m)}$; there is no interaction between the computational grids G_l during the multigrid processing. Thus, the overall algorithm uses multigrid processing only as a fast solver for the Poisson problem on each computational grid; the interaction between grids in one time step is one way only (the coarse grid influences the fine grid through boundary values). We refer to this algorithm as the Berger-Oliger (BO) method.

3.4. Multigrid method. Since the computational grids overlap each other, it is natural to try to use them directly in the multigrid processing. For a time-dependent problem, this is possible only when the fine- and coarse-grid solutions are at the same point in time (in the Runge–Kutta method, at the end of the last stage only); then the problem is equivalent to the steady-state problem treated by Bai and Brandt [2]. To solve the Poisson problem for ψ by their method, we use the multigrid method described above but using the computational grids G_l instead of the local coarse grids $G_l^{(m)}$. In the case of more than two computational grids, the coarse grids used in the multigrid solution for ψ on grid G_l are G_{l-1}, G_{l-2}, $\ldots$, $G_k = G_k^{(1)}$, $G_k^{(2)}$, $\ldots$, $G_k^{(n_k)}$, where k is the index of the coarsest computational grid at the same point in time as G_l. Note that the local coarse grids are still required (for local time stepping), and that the BO method is recovered simply by always setting $k = l$. Since the computational grids are not coextensive, the Full Approximation Scheme (FAS) is needed here. To maintain the proper net vorticity on the coarse grid in the fine-grid region, a correction based on the relative truncation error is added to the coarse-grid values of ζ located on the fine-grid boundary [2]. The interaction between grids in one time step is two-way: the coarse grid sets the original boundary values on the fine grid, which then affects the solution on the full coarse grid through FAS transfer and relaxation. For simplicity, we refer to this method as the multigrid (MG) method.

3.5. Grid adaption. For the results presented here, the number and size of the computational grids are fixed for a given run; however, the grids are allowed to move to track a moving hurricane. This is accomplished by locating the center of the vortex (i.e., the point of maximum vorticity) and moving the grids so they are approximately centered over the vortex while remaining strictly nested. The grid sizes could also be adjusted as the solution evolves (in the MG method) by using the relative truncation error to decide where to refine or coarsen the grid [4]. Experiments with this self-adaptive version of the model will be reported in a later paper.

4. Numerical results. Since no appropriate analytical solutions of the governing equations are available, we test the performance of the model by comparing results computed at various resolutions with those from high-resolution reference runs. The following sections detail the test problems and present numerical evidence for the convergence, conservation properties, and accuracy and efficiency of the model, comparing the BO and MG methods.

4.1. Test problems. The initial condition in each model run consists of an axisymmetric hurricane-like vortex embedded in a large-scale environmental flow. The initial vortex is as given in [5], with tangential wind

$$V(r) = 2V_m \left(\frac{r}{r_m}\right) \frac{\exp[-a(r/r_m)^b]}{1+(r/r_m)^2}, \tag{4.1}$$

corresponding to the initial vorticity

$$\zeta(r) = \frac{\partial(rV)}{r\partial r} = \frac{V}{r}\left[\frac{2}{1+(r/r_m)^2} - ab\left(\frac{r}{r_m}\right)^b\right]. \tag{4.2}$$

Here, $r = [(x-x_0)^2 + (y-y_0)^2]^{1/2}$ is the radial distance from the vortex center (x_0, y_0). Note that V has the approximate maximum value V_m near $r = r_m$ (exact when $a = 0$); the exponential factor is included to make V vanish quickly for large r. We use the values $V_m = 30\,\mathrm{ms}^{-1}$, $r_m = 80\,\mathrm{km}$, $a = 10^{-6}$, and $b = 6$. The computational domain is a square of side length $2L_x = 2L_y = 4096\,\mathrm{km}$, with the vortex initially centered at $x_0 = 768\,\mathrm{km}$ and $y_0 = -768\,\mathrm{km}$. The model is run from $t = 0$ to $t = 72\,\mathrm{hr}$.

Two different cases of environmental flow are considered: a quasi-circular flow on an f-plane (i.e., $\beta = 0$), given by

$$\bar{\psi}(x, y) = \left(\frac{\bar{u}_0 L}{\pi}\right) \cos\left(\frac{\pi x}{L}\right) \cos\left(\frac{\pi y}{L}\right), \tag{4.3}$$

which has no flow through the boundaries, and a more realistic zonal (east-west) flow on a β-plane (for latitude 20° N), given by

$$\bar{\psi}(y) = \left(\frac{\bar{u}_0 L}{2\pi}\right) \cos\left(\frac{2\pi y}{L}\right). \tag{4.4}$$

In each case, $\bar{u}_0$ measures the maximum wind speed associated with the flow and L measures the scale of spatial variation; we use the values $\bar{u}_0 = 10\,\mathrm{ms}^{-1}$ and $L = 4096\,\mathrm{km}$. The solutions for the streamfunction ψ and vorticity ζ are shown in Fig. 4.1 for the environmental flow (4.3) and in Fig. 4.2 for the environmental flow (4.4).

4.2. Conservation. To test the conservation properties of the model we use the environmental flow (4.3). Without adaptive mesh refinement, the discrete analogues of total vorticity, enstrophy, and kinetic energy are conserved exactly (in the limit as $\Delta t \to 0$) as explained in the Appendix. With adaptive mesh refinement, deviations from conservation are small, as shown in Fig. 4.3. Small jumps are evident in kinetic energy with the MG method; these occur at times when the grids are moved. With the BO method, the initialization does not include interaction between grids, so the initial kinetic energy is not particularly accurate. Including a single V-cycle of the FAS multigrid method (as in MG) when initializing the BO method removes this problem.

4.3. Convergence. The model has second-order accuracy in space; to verify this numerically and quantify the effects of the local refinement, we compare the hurricane track computed with various combinations of mesh sizes and patch sizes to that of a high-resolution reference run (512×512 grid, $h = 8\,\mathrm{km}$). We measure the accuracy by the mean forecast "error", defined as the time mean of the l_2 distance between the location of the vortex center (x_c, y_c) (approximated using quadratic interpolation using five points surrounding the vorticity maximum on the finest grid) and that obtained in the reference run. Trapezoidal quadrature is used in time with step size 1 hour.

Figure 4.4 shows this error as a function of the finest mesh size for a variety of model runs. Each point on the graph corresponds to a single model run; the label gives the corresponding base mesh size (in km) and the sizes of the patches used (if any). Patches A, B, C, D, and E have side length 1/2, 3/8, 1/4, 3/16, and 1/8 of the domain length, respectively. For the uniform-resolution runs, the error decreases at the rate $O(h^2)$ as it should. With adaptive refinement, the error is larger; large patches produce little degradation, while smaller patches produce larger errors, as should be expected.

4.4. Accuracy and efficiency. The price paid for adaptive mesh refinement is some degradation in accuracy, as shown above. However, the benefit is increased speed. To evaluate this tradeoff, Fig. 4.5 shows the errors for the same model runs as in Fig. 4.4 (plus some others) as a function of the execution time for the model run (to $t = 72$ hours). In all cases, the runs with adaptive mesh refinement show improvement over the uniform-resolution runs. For approximately the same level of error, adaptive refinement saves factors of up to about 20 in execution time.

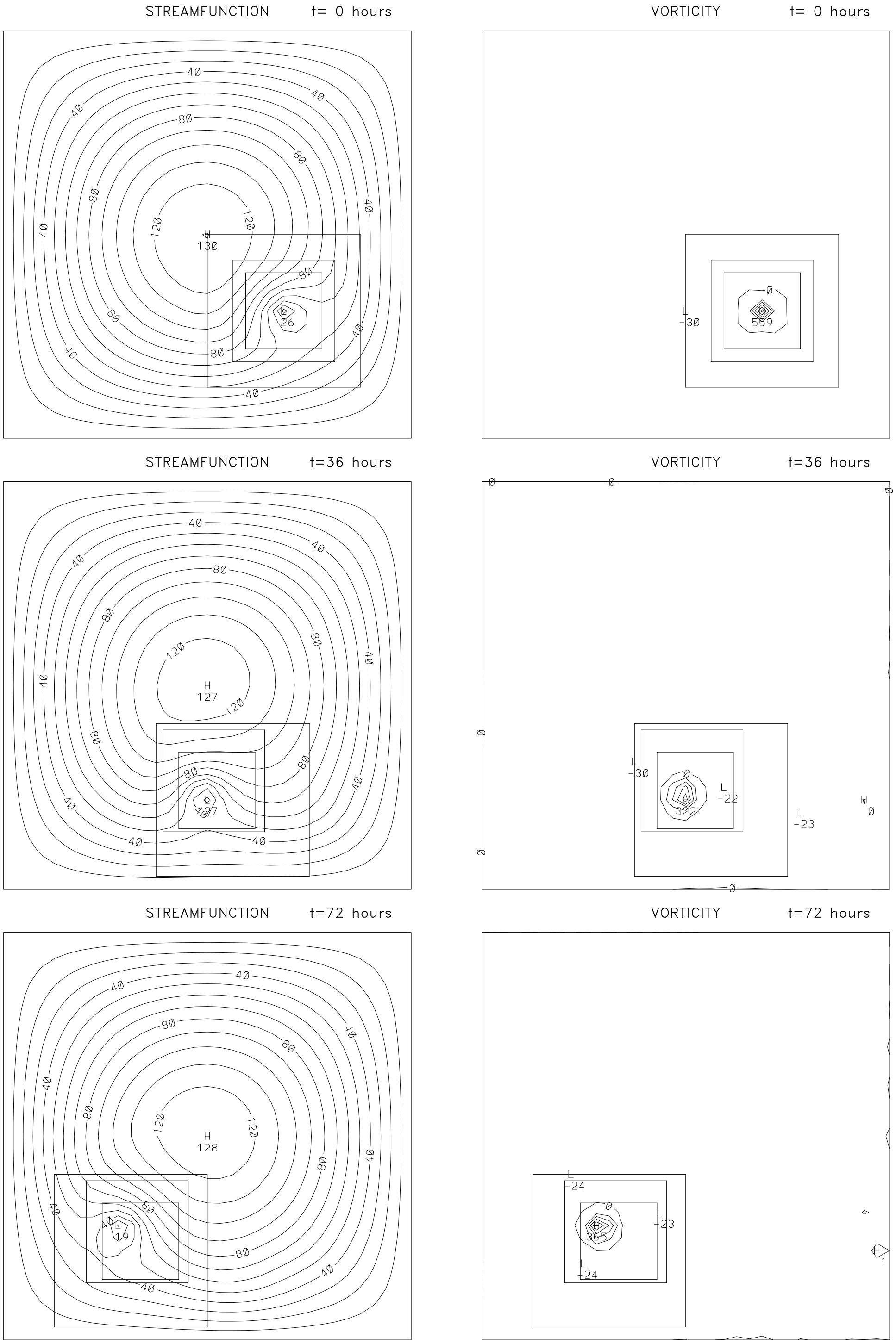

FIG. 4.1. *Streamfunction ψ (left) and vorticity ζ (right) for the vortex (4.1) in the environmental flow (4.3). The squares surrounding the vortex show the extent of the patches used in the numerical solution (MG method).*

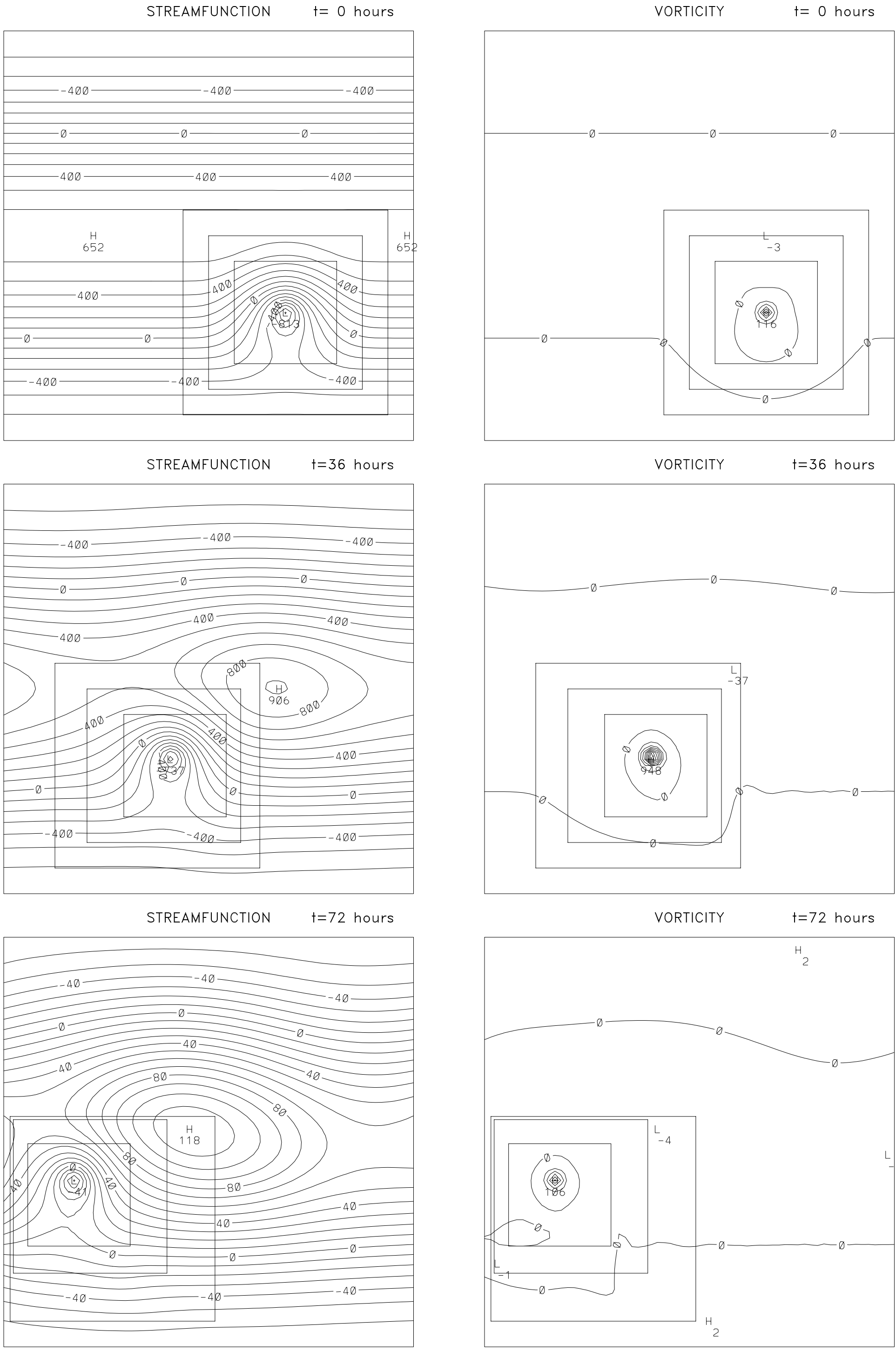

FIG. 4.2. *Streamfunction ψ (left) and vorticity ζ (right) for the vortex (4.1) in the environmental flow (4.4). The squares surrounding the vortex show the extent of the patches used in the numerical solution (MG method).*

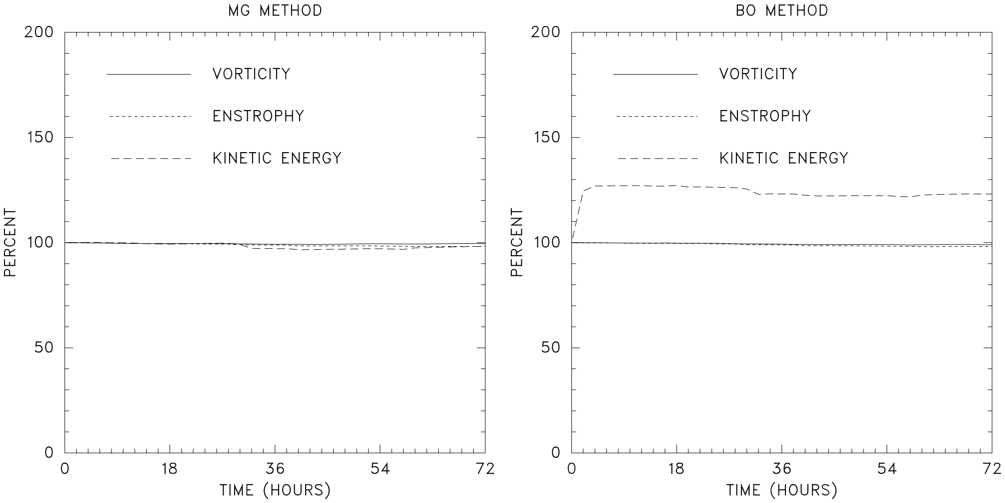

FIG. 4.3. *Normalized vorticity, enstrophy, and kinetic energy in the adaptive model, for the MG method (left panel) and BO method (right panel).*

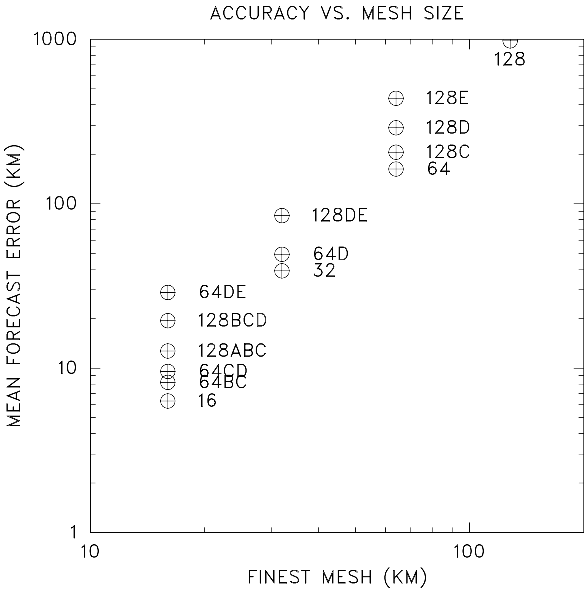

FIG. 4.4. *Accuracy vs. mesh size for various model runs (MG method). Points are labeled with the base mesh size h_1 (in km) and patch sizes (if any).*

4.5. Comparison of BO and MG methods. Since the MG method incorporates more complete interaction between computational grids, it might be expected that it produces better results. Figure 4.6 compares the two methods for some of the runs which were shown in Fig. 4.5. Overall, differences between the two methods are slight, with neither method consistently better. Further experimentation suggests that with many (e.g., three) patches, the

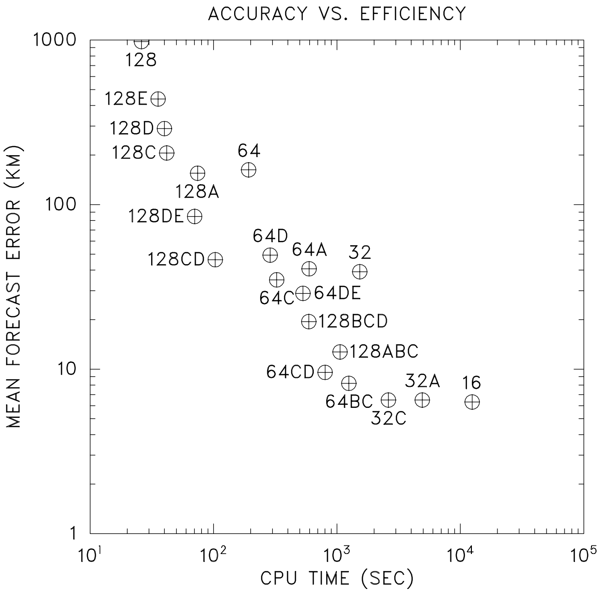

FIG. 4.5. *Accuracy vs. efficiency for various model runs (MG method). Points are labeled as in Fig. 4.4.*

MG method usually produces slightly smaller errors, but the effect is small.

5. Concluding remarks. The adaptive methods described here work well: they solve the hurricane track problem accurately with speedups of 10–20 compared to using uniform resolution. Differences in performance between the BO and MG methods are small. This can be explained by comparing the interaction between grids in the two methods. The MG method includes mutual interaction between coarse and fine grids at each time step. In the BO method, the coarse grid first influences the fine grid through the boundary values, and then the fine grid affects the coarse grid when the fine-grid solution is transferred to the coarse grid at the end of each time step. The difference between the methods amounts to a delay of one time step; the results shown here suggest that the net effect of that difference is minimal, at least for this problem.

One aspect of the methods apparent from the numerical results is that choosing the optimum grids for a problem—even one as simple as that treated here—is not an easy task. Apparently similar combinations of patch sizes may lead to substantially different increases in efficiency (or degradation of accuracy). Thus, self-adaptive mesh refinement, where the patch sizes (and numbers) are adjusted dynamically based on the evolving solution, could have important practical advantages. Such a self-adaptive version of this model is currently being investigated.

Acknowledgments. The helpful comments and advice of Mark DeMaria are gratefully acknowledged. Mark Loftis coded the original (non-adaptive) version of the model.

Appendix. The Arakawa Jacobian. The discretization of the Jacobian given by Arakawa [1] is

$$\begin{aligned} J^h_{i,j}(\zeta,\psi) = -\frac{1}{12h^2}\Big[&+ \left(\psi_{i,j-1}+\psi_{i+1,j-1}-\psi_{i,j+1}-\psi_{i+1,j+1}\right)\left(\zeta_{i+1,j}+\zeta_{i,j}\right) \\ &- \left(\psi_{i-1,j-1}+\psi_{i,j-1}-\psi_{i-1,j+1}-\psi_{i,j+1}\right)\left(\zeta_{i,j}+\zeta_{i-1,j}\right) \end{aligned}$$

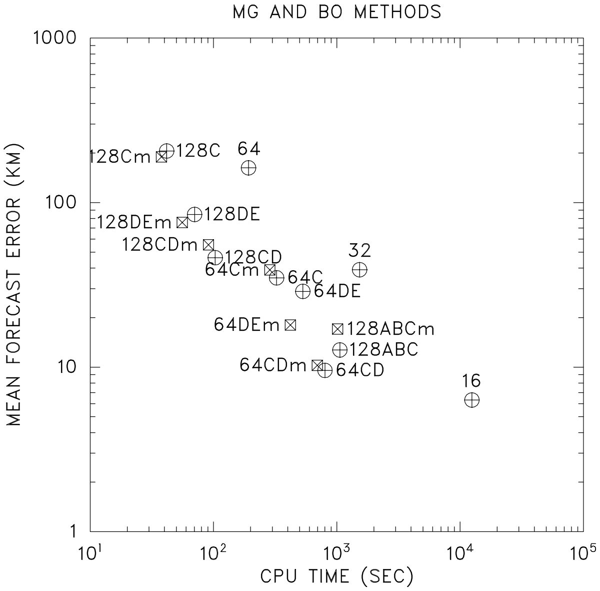

FIG. 4.6. *Accuracy vs. efficiency for various model runs, using both the MG method (circles) and BO method (squares). Points are labeled as in Fig. 4.4.*

$$
\begin{aligned}
&+\left(\psi_{i+1,j}+\psi_{i+1,j+1}-\psi_{i-1,j}-\psi_{i-1,j+1}\right)\left(\zeta_{i,j+1}+\zeta_{i,j}\right)\\
&-\left(\psi_{i+1,j-1}+\psi_{i+1,j}-\psi_{i-1,j-1}-\psi_{i-1,j}\right)\left(\zeta_{i,j}+\zeta_{i,j-1}\right)\\
&+\left(\psi_{i+1,j}-\psi_{i,j+1}\right)\left(\zeta_{i+1,j+1}+\zeta_{i,j}\right)\\
&-\left(\psi_{i,j-1}-\psi_{i-1,j}\right)\left(\zeta_{i,j}+\zeta_{i-1,j-1}\right)\\
&+\left(\psi_{i,j+1}-\psi_{i-1,j}\right)\left(\zeta_{i-1,j+1}+\zeta_{i,j}\right)\\
&-\left(\psi_{i+1,j}-\psi_{i,j-1}\right)\left(\zeta_{i,j}+\zeta_{i+1,j-1}\right)\Big].
\end{aligned}
\tag{A.1}
$$

The form given in [1] for the corresponding discretization at the boundary assumes no flow through the boundary, and thus must be modified for the problem treated here. One way to do so is to simply replace centered differences by one-sided differences at the boundary; this leads to

$$
\begin{aligned}
\frac{1}{2}J^h_{i,N}(\zeta,\psi) = -\frac{1}{12h^2}\Big[&+\left(\psi_{i,N-1}+\psi_{i+1,N-1}-2\psi_{i,N}\right)\left(\zeta_{i+1,N}+\zeta_{i,N}\right)\\
&-\left(\psi_{i-1,N-1}+\psi_{i,N-1}-2\psi_{i,N}\right)\left(\zeta_{i,N}+\zeta_{i-1,N}\right)\\
&-\left(\psi_{i+1,N-1}+\psi_{i+1,N}-\psi_{i-1,N-1}-\psi_{i-1,N}\right)\left(\zeta_{i,N}+\zeta_{i,N-1}\right)\\
&-\left(\psi_{i,N-1}-\psi_{i-1,N}\right)\left(\zeta_{i,N}+\zeta_{i-1,N-1}\right)\\
&-\left(\psi_{i+1,N}-\psi_{i,N-1}\right)\left(\zeta_{i,N}+\zeta_{i+1,N-1}\right)\\
&+4\left(\psi_{i+1,N}-\psi_{i-1,N}\right)\zeta_{i,N}\Big].
\end{aligned}
\tag{A.2}
$$

at the north boundary and

$$
\begin{aligned}
\frac{1}{4}J^h_{M,N}(\zeta,\psi) = -\frac{1}{12h^2}\Big[&-\left(\psi_{M-1,N-1}+\psi_{M,N-1}-2\psi_{M,N}\right)\left(\zeta_{M,N}+\zeta_{M-1,N}\right)\\
&-\left(2\psi_{M,N}-\psi_{M-1,N-1}-\psi_{M-1,N}\right)\left(\zeta_{M,N}+\zeta_{M,N-1}\right)
\end{aligned}
$$

$$
\begin{aligned}
&- \left(\psi_{M,N-1} - \psi_{M-1,N}\right)\left(\zeta_{M,N} + \zeta_{M-1,N-1}\right)\\
&+ 4\left(\psi_{M,N-1} - \psi_{M,N}\right)\zeta_{M,N} - 4\left(\psi_{M-1,N} - \psi_{M,N}\right)\zeta_{M,N}\Big].
\end{aligned}
\tag{A.3}
$$

at the northeast corner, with analogous discretizations at the other sides and corners.

Substituting (A.1–A.3) into (3.1), multiplying by h^2 and summing over the grid leads to

$$
\begin{aligned}
\frac{d}{dt}\sum_{i=0}^{M}{}''\sum_{j=0}^{N}{}''\zeta_{i,j}h^2 = &-\sum_{j=0}^{N}{}''\left[(u\zeta)_{M,j} - (u\zeta)_{0,j}\right]h\\
&-\sum_{i=0}^{M}{}''\left[(v\zeta)_{i,N} - (v\zeta)_{i,0}\right]h,
\end{aligned}
\tag{A.4}
$$

where the double primes indicate the first and last terms of the sum are scaled by the factor $\frac{1}{2}$. Here we have assumed that $\beta = 0$, and the discretization of the boundary fluxes matches (A.2), e.g.,

$$
\begin{aligned}
(v\zeta)_{i,N} = \frac{1}{12h}\Big[&\left(\psi_{i,N} - \psi_{i-1,N}\right)\left(\zeta_{i,N} + \zeta_{i-1,N}\right)\\
&+ 4\left(\psi_{i+1,N} - \psi_{i-1,N}\right)\zeta_{i,N}\\
&+ \left(\psi_{i+1,N} - \psi_{i,N}\right)\left(\zeta_{i,N} + \zeta_{i+1,N}\right).
\end{aligned}
\tag{A.5}
$$

Thus, if there is no flow through the boundary, the discrete model conserves the discrete total vorticity

$$
\sum_{i=0}^{M}{}''\sum_{j=0}^{N}{}''\zeta_{i,j}h^2.
\tag{A.6}
$$

Likewise, under the same conditions the discrete total enstrophy

$$
\sum_{i=0}^{M}{}''\sum_{j=0}^{N}{}''\frac{1}{2}\zeta_{i,j}^2 h^2
\tag{A.7}
$$

and discrete kinetic energy

$$
\begin{aligned}
\frac{1}{4}\sum_{i=0}^{M}\sum_{j=0}^{N}\Big[\quad&\left(\frac{\psi_{i-1,j} - \psi_{i-1,j-1}}{h}\right)^2 + \left(\frac{\psi_{i,j} - \psi_{i,j-1}}{h}\right)^2\\
+&\left(\frac{\psi_{i,j-1} - \psi_{i-1,j-1}}{h}\right)^2 + \left(\frac{\psi_{i,j} - \psi_{i-1,j}}{h}\right)^2\Big]h^2
\end{aligned}
\tag{A.8}
$$

are also conserved [1].

REFERENCES

[1] A. ARAKAWA, *Computational design for long-term numerical integration of the equations of fluid motion: Two-dimensional incompressible flow, Part I*, J. Comp. Phys., 1 (1966), pp. 119–143.

[2] D. BAI AND A. BRANDT, *Local mesh refinement multilevel techniques*, SIAM J. Stat. Comput., 8 (1987), pp. 109–304.

[3] M. BERGER AND J. OLIGER, *Adaptive mesh refinement for hyperbolic partial differential equations*, J. Comp. Phys., 53 (1984), pp. 484–512.

[4] A. Brandt, *Multi-level adaptive solutions to boundary-value problems*, Math. Comput., 31 (1977), pp. 333–390.

[5] M. DeMaria, *Tropical cyclone track prediction with a barotropic model*, Mon. Wea. Rev., 113 (1985), pp. 1199–1210.

[6] M. DeMaria, *Tropical cyclone track prediction with a barotropic spectral model*, Mon. Wea. Rev., 115 (1987), pp. 2346–2357.

[7] S. F. McCormick and J. Thomas, *The fast adaptive composite grid method (FAC) for elliptic boundary value problems*, Math. Comput., 46 (1986), pp. 439–456.

[8] J. Oliger, J. and A. Sundström, *Theoretical and practical aspects of some initial boundary value problems in fluid dynamics*, SIAM J. Appl. Math., 35 (1978), pp 419–446.

[9] R. K. Smith, W. Ulrich, and G. Dietachmayer, *A numerical study of tropical cyclone motion using a barotropic model, Part I: The role of vortex asymmetries*, Quart. J. Roy. Meteor. Soc., 116 (1990), pp. 337-362.