AN ADAPTIVE MULTIGRID MODEL FOR HURRICANE TRACK PREDICTION

Scott R. Fulton
Department of Mathematics and Computer Science
Clarkson University
Potsdam, NY

SUMMARY

This paper describes a simple numerical model for hurricane track prediction which uses a multigrid method to adapt the model resolution as the vortex moves. The model is based on the modified barotropic vorticity equation, discretized in space by conservative finite differences and in time by a Runge-Kutta scheme. A multigrid method is used to solve an elliptic problem for the streamfunction at each time step. Nonuniform resolution is obtained by superimposing uniform grids of different spatial extent; these grids move with the vortex as it moves. Preliminary numerical results indicate that the local mesh refinement allows accurate prediction of the hurricane track with substantially less computer time than required on a single uniform grid.

INTRODUCTION

Accurately predicting the track of a moving hurricane is a problem of great practical importance. One approach is to treat the problem as one in computational fluid dynamics, taking observed meteorological data as initial values for a numerical model. Many factors influence the accuracy of this approach, including the initial data (or lack thereof), the dynamical and physical processes included in the model, and the numerical scheme employed. While the relative importance of these three factors is a subject of considerable debate, in this paper we focus on the third.

Our premise is that predicting the track of a moving hurricane accurately requires resolving the flow field adequately on both the large scale surrounding the vortex and the small scale within the vortex itself. Since the spatial scales involved may differ by more than an order of magnitude, models using uniform resolution are inherently less efficient than what should be possible. Here, we use a simple dynamical model which has been used successfully by many authors (ref. 1, 2, 3), namely, the modified barotropic vorticity equation. However, rather than use a single uniform grid as in those studies, we investigate the use of adaptive multigrid techniques, with the goal of obtaining high accuracy at low computational cost. In the following sections we detail the formulation of the model, describe the mesh refinement scheme, and present some preliminary numerical results.

*Work supported by the National Science Foundation under Grant No. ATM-9118966

MODEL FORMULATION

Governing Equations

We formulate the model on a section of the sphere using a Mercator projection (true at latitude $\phi = \phi_c$). The model consists of the modified barotropic vorticity equation

$$\frac{\partial \zeta}{\partial t} + m^2 J(\psi, \zeta) + \beta m \frac{\partial \psi}{\partial x} = \nu m^2 \nabla^2 \zeta, \tag{1}$$

where the relative vorticity ζ and streamfunction ψ are related by

$$\left(m^2 \nabla^2 - \gamma^2\right) \psi = \zeta. \tag{2}$$

Here $\nabla^2 = \partial^2/\partial x^2 + \partial^2/\partial y^2$, $J(\psi, \zeta)$ is the Jacobian of (ψ, ζ) with respect to (x, y), $\beta = 2\Omega \cos\phi / a$ (with a and Ω the radius and rotation rate of the earth), and $m = \cos\phi_c / \cos\phi$ is the map factor. There are two quasi-physical parameters: the diffusion coefficient ν, and the parameter γ (inverse of the effective Rossby radius) which helps prevent retrogression of ultralong Rossby waves (ref. 4). We also consider versions on the f-plane ($m = 1$ and $\beta = 0$) and β-plane ($m = 1$ and $\beta = 2\Omega \cos\phi_c / a$). The model domain is a rectangle in x and y centered at $(x, y) = (0, 0)$, where $(\lambda, \phi) = (\lambda_c, \phi_c)$. At the boundaries we specify the streamfunction ψ (and thus the normal component of the velocity); where there is inflow, we also specify the vorticity ζ.

Space Discretization

On a single uniform rectangular grid Ω^h consisting of gridpoints (x_i, y_j) with mesh spacing h in x and y, we discretize (1) and (2) in space by finite differences as

$$\frac{d\zeta_{i,j}}{dt} + m_j^2 \mathcal{J}_{i,j}(\psi, \zeta) + \beta_j m_j \partial_x^{2h} \psi_{i,j} = \nu m_j^2 \nabla_{i,j}^2 \zeta_{i,j} \tag{3}$$

and

$$\left(m_j^2 \nabla_{i,j}^2 - \gamma^2\right) \psi_{i,j} = \zeta_{i,j}, \tag{4}$$

respectively. Here $\mathcal{J}_{i,j}(\psi, \zeta)$ is the discrete Jacobian of Arakawa (ref. 5), and $\partial_x^{2h} \psi_{i,j}$ and $\nabla_{i,j}^2 \psi_{i,j}$ are the $O(h^2)$ centered difference approximations to $\partial\psi/\partial x$ and the Laplacian operator, respectively. We apply (3) and (4) at the interior points. At boundary points where there is inflow, ζ is specified; otherwise, we predict ζ on the boundaries by applying an equation of the form (3) but using appropriate one-sided differences. It should be noted that using the Arakawa Jacobian is crucial here: the fact that it conserves discrete analogues of vorticity, enstrophy (mean square vorticity), and kinetic energy implies that the model is not subject to nonlinear computational instability.

To write the space-discretized equations in a more compact form, we collect the values $\psi_{i,j}$ and $\zeta_{i,j}$ into grid functions ψ^h and ζ^h, respectively, defined on the grid Ω^h. We can then write (3) and (4) as

$$\frac{d\zeta^h}{dt} = F^h(\psi^h, \zeta^h) \tag{5}$$

and

$$G^h(\psi^h) = \zeta^h, \tag{6}$$

where the operators F^h and G^h express the space discretization described above.

Time Discretization

To discretize (5) and (6) in time we use the classical fourth-order Runge-Kutta (RK4) scheme. To describe it, we specify a time step $\Delta t > 0$ and introduce time levels $t_k = k\Delta t$ for $k = 0, \ldots$. Suppressing the superscript h for simplicity, we now use the superscript k to denote values at time level k, e.g., $\psi^k \approx \psi^h(t_k)$. With this notation, the RK4 scheme can be written as

$$\begin{aligned}
\frac{\tilde{\zeta}^{k+\frac{1}{2}} - \zeta^k}{\frac{1}{2}\Delta t} &= F^k \quad := F(\psi^k, \zeta^k), & G(\tilde{\psi}^{k+\frac{1}{2}}) &= \tilde{\zeta}^{k+\frac{1}{2}}, \\
\frac{\zeta^{k+\frac{1}{2}} - \zeta^k}{\frac{1}{2}\Delta t} &= \tilde{F}^{k+\frac{1}{2}} := F(\tilde{\psi}^{k+\frac{1}{2}}, \tilde{\zeta}^{k+\frac{1}{2}}), & G(\psi^{k+\frac{1}{2}}) &= \zeta^{k+\frac{1}{2}}, \\
\frac{\tilde{\zeta}^{k+1} - \zeta^k}{\Delta t} &= F^{k+\frac{1}{2}} := F(\psi^{k+\frac{1}{2}}, \zeta^{k+\frac{1}{2}}), & G(\tilde{\psi}^{k+1}) &= \tilde{\zeta}^{k+1}, \\
\frac{\zeta^{k+1} - \zeta^k}{\Delta t} &= \bar{F}^{k+1}, & G(\psi^{k+1}) &= \zeta^{k+1},
\end{aligned} \tag{7}$$

where

$$\bar{F}^{k+1} = \frac{1}{6}\left(F^k + 2\tilde{F}^{k+\frac{1}{2}} + 2F^{k+\frac{1}{2}} + \tilde{F}^{k+1}\right). \tag{8}$$

Thus, to execute a single time step $t_k \rightarrow t_{k+1}$, we perform the four stages indicated in (7); each of these stages consists of computing F based on known values of ψ and ζ, predicting a new vorticity ζ, and solving the diagnostic equation for the corresponding streamfunction ψ.

Although it requires four times as much work (per time step) as the second-order Adams-Bashforth scheme commonly used in such models, this RK4 scheme has several advantages. First, it allows time steps at least four times as large, so in fact it is more efficient. Second, it is more accurate, so time discretization errors are less likely to distort the conservation properties of the Arakawa Jacobian. Finally, since it is a one-step scheme, it has no computational modes and needs no other method for the initial time step.

Multigrid Solution

To solve the diagnostic equation at each stage for the streamfunction ψ, we use a multigrid method. For the relaxation scheme we use a point Gauss-Seidel method formulated as follows. The discrete (interior) equation (4) can be written as

$$(L\psi)_{i,j} = \frac{1}{h^2}\left(\sigma_j \psi_{i,j} - S_{i,j}\right) = -\frac{\zeta_{i,j}}{m_j^2} = F_{i,j}, \tag{9}$$

where

$$S_{i,j} := \psi_{i-1,j} + \psi_{i+1,j} + \psi_{i,j-1} + \psi_{i,j+1} \tag{10}$$

is the sum of the neighboring values of ψ and

$$\sigma_j := 4 + \frac{\gamma^2 h^2}{m_j^2} \tag{11}$$

is the diagonal term of the discrete Helmholtz operator. Given an approximate solution $\tilde{\psi}$ of (9), we relax at a point (i,j) by changing the value there to satisfy the corresponding equation (9); this results in the new value

$$\bar{\psi}_{i,j} = \frac{h^2 F_{i,j} + \tilde{S}_{i,j}}{\sigma_j}, \tag{12}$$

where $\tilde{S}_{i,j}$ is defined using the current surrounding values in (10). The corresponding residual (if needed) is given by

$$r_{i,j} := F_{i,j} - \frac{1}{h^2}\left(\sigma_j \tilde{\psi}_{i,j} - \tilde{S}_{i,j}\right) = \frac{\sigma_j}{h^2}\left(\bar{\psi}_{i,j} - \tilde{\psi}_{i,j}\right). \tag{13}$$

We use this relaxation (with red-black ordering) as a smoother in a multigrid method, using half-injection for the fine-to-coarse transfer of residuals and bilinear interpolation for the coarse-to-fine transfer of corrections. For the control algorithm we use repeated V(1,1)-cycles.

LOCAL MESH REFINEMENT

Given the premise that the flow near the center of the vortex requires much higher resolution than the flow surrounding the vortex, we now consider how to provide such variable resolution. Our basic method is essentially that of (ref. 6), constructing nonuniform resolution by superimposing uniform grids of varying spatial extent. Since all calculations are carried out on the uniform grids, programming remains relatively easy.

To illustrate the method, let us consider first the case of two grids: a coarse grid Ω^{2h} covering the whole domain Ω, and a fine grid Ω^h which covers only a portion of the domain (i.e., enclosing the vortex). We assume that the boundaries of the fine grid coincide with coarse grid lines. The model variables ζ and ψ are carried on both the coarse and fine grids (denoted by ζ^{2h}, ψ^{2h} and ζ^h, ψ^h, respectively). Noting that the coarse grid allows time steps twice as large as those on the fine grid, we use the following basic procedure for stepping the model from time t_k to t_{k+1}:

1. Execute one time step of length Δt on the coarse grid to produce $\zeta^{2h,k+1}$, $\psi^{2h,k+1}$;

2. Execute two time steps of length $\Delta t/2$ on the fine grid to produce $\zeta^{h,k+1}$, $\psi^{h,k+1}$, using boundary values for ψ interpolated from the coarse grid (in space and time);

3. Copy the fine-grid solution to the coarse grid at points common to both.

Several points deserve mention here. First, in solving the implicit problem for ψ on either grid, we use the multigrid method outlined above. This introduces additional coarse grids, e.g., a grid with mesh spacing $2h$ covering only the region of the local fine grid Ω^h. In fact, the "underlying" part of the coarse grid Ω^{2h} could be used for this; however, the resulting complications of preserving interface values (for fine-grid boundary values) and restricting relaxation to only part of Ω^{2h} seem too high a price to pay for the relatively small savings in storage which would be achieved. Second, after completing the above three steps, the resulting solution on the composite grid $\Omega^{\underline{h}} = \Omega^h \cup \Omega^{2h}$ could be further refined by applying a composite-grid discretization of the governing equations; this FAC (Fast Adaptive Composite grid) method and several variants are described in (ref. 7), and will be explored in future work. Finally, the above approach generalizes immediately to more than two grids.

For the initial work reported here, we have made the following simplifying assumptions. First, we require the grids to be rectangular and strictly nested (i.e., any fine grid is contained wholly within the interior of the next coarser grid), with one grid per level (i.e., the refinement occurs in one region only, surrounding the vortex). Second, we use a constant mesh ratio of two (i.e., the mesh spacing h on any grid is twice that of the next finer grid, if any). Finally, we will specify the number of grids and their sizes in advance but allow them to move following the vortex as the solution is computed.

Since the problem to be solved has an easily identifiable region of interest surrounding the vortex, we take the following simple approach to moving the grids. First, we locate the vortex center on the finest grid. Then for each grid in turn, from the next-to-coarsest to the finest, we decide whether or not to move the grid. This decision is based on the distance of the vortex center from the center of the grid: if it is more than a specified fraction α of the distance L to the boundary, we move the grid. The move is calculated so as to "overshoot" a bit, i.e., aiming to put the vortex center beyond the (new) grid center by a specified fraction δ of the distance to the grid boundary. Note that care must be taken at this stage to ensure the strict nesting of grids assumed above. Finally, the grid is moved by shifting the values which remain on the grid and filling in the rest by interpolation from the next coarser grid. For the results presented here, we check for possible grid moves after each time step on the coarsest grid, and use the parameters $\alpha = 0.4$ and $\delta = 0.2$.

To locate the vortex center (needed both for moving the grids as described above and for determining the vortex track), we first locate the point of maximum vorticity on the finest grid. We then interpolate the vorticity at that point and its nearest neighbors in x and y (five points total) by a quadratic function, and define the vortex center to be the location of the maximum of that quadratic.

RESULTS

The initial conditions for the test problem consist of an axisymmetric vortex superimposed on an environmental flow, as considered in (ref. 1). The environmental flow is given by

$$\bar{\psi}(y) = \left(\frac{\bar{u}_0 L}{2\pi}\right) \cos\left(\frac{2\pi y}{L}\right), \tag{14}$$

which corresponds to the zonal current

$$\bar{u}(y) = -\frac{d\bar{\psi}}{dy} = \bar{u}_0 \sin\left(\frac{2\pi y}{L}\right). \tag{15}$$

The tangential wind in the initial vortex is given by

$$V(r) = 2V_m \left(\frac{r}{r_m}\right) \frac{\exp[-a(r/r_m)^b]}{1 + (r/r_m)^2}, \tag{16}$$

where $r = [(x - x_0)^2 + (y - y_0)^2]^{1/2}$ 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. The vorticity corresponding to (16) is

$$\zeta(r) = \frac{\partial(rV)}{r\partial r} = \frac{V}{r}\left[\frac{2}{1 + (r/r_m)^2} - ab\left(\frac{r}{r_m}\right)^b\right]. \tag{17}$$

We will use the following parameter values: $\bar{u}_0 = 10\,\mathrm{ms}^{-1}$ and $L = 4000\,\mathrm{km}$ for the environmental flow, and $V_m = 30\,\mathrm{ms}^{-1}$, $r_m = 80\,\mathrm{km}$, $a = 10^{-6}$, and $b = 6$ for the initial vortex. The computational domain is a square of side length 4000 km on a β-plane, using β for the latitude 20° N; the vortex is initially centered at $x_0 = 750\,\mathrm{km}$ and $y_0 = -750\,\mathrm{km}$. The model was run from $t = 0$ to $t = 72\,\mathrm{hr}$; for simplicity we have set $\nu = 0$ and $\gamma = 0$ here.

To establish a standard for comparison, we ran the model with high resolution (384 × 384 grid with spacing $h = 10.42\,\mathrm{km}$ and time step 10 s). We then ran the model with a variety of grid configurations (using up to four grids) and compared the vortex track to that of the reference run. Table I summarizes these results, with the runs listed in order of increasing execution time (on a SUN SPARCstation2). All of the cases in this table use only square grids, with $N_x = N_y = N$. The forecast error is defined as the distance between the predicted vortex location at a given time and that in the reference run. These results show that the local refinement process has the potential to substantially reduce the execution time required to achieve a given accuracy. For example, a single grid with $h = 31.25\,\mathrm{km}$ (run 6) achieves errors on the order of 10–20 km; with local refinement (run 2) comparable accuracy is obtained with only about 36% as much computer time. Similarly, a single grid with $h = 20.83\,\mathrm{km}$ (run 8) achieves errors on the order of 1–5 km; with local refinement (run 7) comparable accuracy is obtained with only about 42% as much computer time. In fact, run 7 with local refinement achieved about the same accuracy as did the single-grid run with $h = 15.625\,\mathrm{km}$ (run 9) but with only about 18% of the computer time. In addition, the solution fields produced with local refinement (run 7) are smooth, as shown in Figures 1–5, with no indication of any problem due to the change of resolution at the grid interfaces.

CONCLUDING REMARKS

The preliminary results reported here show that adaptive multigrid techniques can substantially reduce the computer time required to make accurate hurricane track forecasts. In addition to ongoing testing of the existing model, we plan to investigate the following possible improvements. First, we plan to include the FAC method as discussed above. This should have the advantage of more precise conservation of vorticity, enstrophy, and kinetic energy at the grid interfaces. Second, we plan to construct a fully adaptive version of the model by using the Full Approximation Scheme (FAS) to produce estimates of the local truncation error to be used in an automatic grid refinement scheme (as proposed in ref. 8). Finally, we plan to test the model using real data, and compare its performance to that of models currently in operational use.

REFERENCES

1. DeMaria, M.: Tropical cyclone track prediction with a barotropic model. *Mon. Wea. Rev.*, vol. 113, 1985, pp. 1199–1210.

2. DeMaria, M.: Tropical cyclone track prediction with a barotropic spectral model. *Mon. Wea. Rev.*, vol. 115, 1987, pp. 2346–2357.

3. Smith, R. K., 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.*, vol. 116, 1990, pp. 337-362.

4. Holton, J. R.: *An Introduction to Dynamic Meteorology*, second edition. Academic Press, 1979.

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

6. Berger, M., and J. Oliger: Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comp. Phys.*, vol. 53, 1984, pp. 484–512.

7. McCormick, S. F.: *Multilevel Adaptive Methods for Partial Differential Equations.* SIAM, 1989.

8. Brandt, A.: Multi-level adaptive solutions to boundary-value problems. *Math. Comp.*, vol. 31, 1977, pp. 333–390.

Table I. Results of Model Runs

	Grid size(s)		Δt	Forecast error (km) at:			Execution
Run	N	h (km)	(min)	24 hr	48 hr	72 hr	time (sec)
1	64	62.50	60	110	143	47	170
2	64	62.50	60	11	8	17	504
	64	31.25	30				
3	96	41.67	30	53	12	25	799
4	32	125.0	120	14	24	39	916
	32	62.50	60				
	48	31.25	30				
	64	15.62	15				
5	64	62.50	60	1	6	10	1,174
	64	31.25	30				
	64	15.62	15				
6	128	31.25	30	11	8	19	1,409
7	64	62.50	60	1	5	5	2,047
	64	31.25	30				
	96	15.62	15				
8	192	20.83	20	1	3	5	4,860
9	256	15.62	15	2	3	4	11,405
10	384	10.42	10	–	–	–	41,716