JOURNAL OF COMPUTATIONAL PHYSICS 80, 27–60 (1989)

Adaptive Grid Refinement for Numerical Weather Prediction*

WILLIAM SKAMAROCK

Environmental Fluid Mechanics Laboratory, Stanford University, Stanford, California 94305

JOSEPH OLIGER

Department of Computer Science, Stanford University, Stanford, California 94305

AND

ROBERT L. STREET

Environmental Fluid Mechanics Laboratory, Stanford University, Stanford, California 94305

Received April 8, 1987; revised January 26, 1988

An adaptive atmospheric flow model is described and results of integrations with this model are presented. The adaptive technique employed is that of Berger and Oliger. The technique uses a finite difference method to integrate the dynamical equations first on a coarse grid and then on finer grids which have been placed based on a Richardson-type estimate of the truncation error in the coarse grid solution. By correctly coupling the integrations on the various grids, periodically re-estimating the error, and recreating the finer grids, uniformily accurate solutions are economically produced. The "primitive" hydrostatic equations of meteorology are solved for the advection of a barotropic cyclone and for the development of a baroclinic disturbance which results from the perturbation of an unstable jet. These integrations demonstrate the feasibility of using multiple, rotated, overlapping fine grids. Direct computations of the truncation error are used to confirm the accuracy of the Richardson-type truncation error estimates.

1. INTRODUCTION

Accurately computing atmospheric flows is a difficult task. Integrations must be performed accurately and in a timely manner if the simulations are to be helpful to

* This work has been supported by the Office of Naval Research Contracts N00014-84-K-0267 and N00014-82-K-0335.

0021-9991/89 $3.00

coincide. The scheme is thus two-way interactive because the fine grid solution does influence the coarse grid solution.

The two-way interactive method has been implemented in two forms, one where the fine grid location is fixed and another where the fine grid is allowed to move during the time integration. An example of the former is the nested grid model (NGM) which has been developed at the National Meteorological Center. The National Weather Service distributes NGM results as guidance to forecasters. The model consists of a hemispherical grid over which two finer grids have been placed. The fine grids are centered over North America because this is where forecast information is needed.

Several tropical cyclone models have fine grids which move during the integration to keep the fine grid over the cyclone. This is accomplished by moving the fine grid when a solution feature, such as the surface low associated with the cyclone, moves. In all cases the fine grids are aligned with the coarse grids, but they may move incrementally up, down, or sideways. Examples of these are tropical cyclone models described by Harrison [10] and Jones [14].

The tropical cyclone models attempt to provide resolution around features which are local and poorly resolved by the coarse grid. The NGM provides increased resolution over a continent because increased resolution may be necessary there, but it is difficult to know a priori precisely where it will be necessary. From a global perspective tropical cyclones, fronts, jet streams, and other atmospheric phenomena are spatially and temporally localized and are often poorly resolved in present atmospheric models. Local phenomena should be handled adaptively but no adaptive atmospheric flow solvers exist. The nested tropical cyclone models are not truly adaptive. The initial location of the cyclone must be known to locate the fine grid. The fine grid will remain even if the cyclone disappears, and if a new cyclone were to appear elsewhere no new fine grid would appear over it.

Phenomena in many other fluid flows which are difficult to resolve are often localized. Adaptive solvers do exist for many of these flows. In general two adaptive strategies are used. In the first all existing gridpoints are redistributed from regions of small solution variation to regions of large solution variation. These global methods vary in the criteria and methods used to move the points, but in all cases the total number of points remains the same. They are usually used in conjunction with grid transformation methods which involve mapping an irregular physical domain into a rectangular computational domain. The second strategy involves adding or deleting grid points so as to obtain a desired solution accuracy. The additions and deletions are local, thus the techniques are *local* grid refinement techniques.

Atmospheric flows appear ideally suited to local grid refinement techniques because the important phenomena are localized. The local grid refinement techniques can be broken into two categories: One in which the new points are inserted or imbedded into the existing grid, and hence only one grid exists, and a second where refinements are placed over the existing grid, the refinements constituting separate grids.

A fine example of embedding new points on an existing grid is the work of Dannenhoffer and Baron [8]. Their code solves transonic flow over a 2-D airfoil. Refinements are based on refinement parameters, for example, first or second order differences in the density, pressure, or entropy. An expert system handles the refinement parameters and rules governing how and where to refine.

In this technique grids are no longer rectangular in nature and neither is the data structure which holds the solution fields. A significant amount of information must be stored to describe the grid structure. The solver which uses this grid structure is complex. Even with this complexity and loss of rectangularity much vectorization of the code is possible and efficient integrations are being performed.

An example where refinements are placed over the existing grid is the scheme of Berger and Oliger [6]. The same scheme has been used by Berger and Jameson [5] who also solve transonic flow over a 2-D airfoil. In this technique the refinements are separate rectangular grids rather than being points embedded in the coarse grid. Any solver which works on a rectangle can be used, because the solver is just a module called by the adaptive routines to advance the solution. Berger and Jameson calculate results similar to those of Dannenhoffer and Baron. There are many other differences between the two schemes and interested readers should consult the referenced papers.

Solver complexity is a strong barrier in NWP to techniques which imbed points into existing grids. Weather prediction codes solve much more than a simple set of dynamical equations. There are equations for water vapor (and possibly water in its other states), routines which calculate radiational heating and heating due to phase changes of water, routines which model cloud effects, complex calculations for fluxes of heat and moisture into the atmosphere, and usually parameterizations of other processes. Most codes are the result of many peoples' effort over several years. Proven and tested routines are often borrowed from one code to put into another. Adaptive methods using refinements which are separate grids appear the logical choice for use as the basis of an adaptive weather model. Existing, well-tested software can often be used with only minor changes and procedures can be written with little knowledge of the adaptive routines.

In this paper we present results from an adaptive atmospheric flow solver which uses the method and software developed by Berger and Oliger. The adaptive method operates on multiple, component grids. Fine grids, which overlie the coarse grid(s), are created and removed based on a Richardson-type estimate of the truncation error in the finite difference solution. The goal is to maintain a given accuracy for a minimum amount of work. This is the first attempt with this method to adaptively solve a system describing a three-dimensional time-dependent flow.

Our purpose is to show that an adaptive atmospheric flow solver is feasible and that the adaptive technique will produce self-consistent results. In essence we are proving a concept, the concept being (1) that refinement should occur only where necessary, as dictated by the error in the numerical solution, and in this way improve the accuracy and overall resolution of the entire solution, and (2) that this can be accomplished by using the method of Berger and Oliger. Hence, we wish

only to demonstrate that our adaptive model is successful when compared with results from the same solver on a single grid, this being sufficient to demonstrate the feasibility of the adaptive atmospheric flow solver. For prediction purposes one would use the best available solver for the scales one is attempting to predict.

In Section 2 we review the adaptive grid refinement technique of Berger and Oliger. The solution procedure is outlined along with a description of the data structure and program design. Section 3 describes the equations and solver we have used in the adaptive package. The initial test cases describe flow in an idealized atmosphere: adiabatic flow in a periodic channel with no moisture present in the atmosphere and no diabatic heating. We are using the Euler equations with the hydrostatic assumption, the so called "primitive equations" of meteorology. At the end of the section we discuss the issues of stability and accuracy for these equations as they are used in this adaptive context. In Section 4 we examine the results of two simulations, one for a barotropic cyclone and another for a baroclinically unstable jet, and show that our adaptive model is self consistent and successful in simulating these flows. We examine the error estimate in Section 5 and conclusions follow in Section 6.

2. Review of the Adaptive Grid Refinement Technique

We describe the adaptive procedure as used for 2-D hyperbolic problems. For large-scale atmospheric flows the horizontal scales are orders of magnitude larger than the vertical scales. In our adaptive model we refine in the horizontal and keep the same number of layers in the vertical. Thus we treat atmospheric flow as a 2-D grid refinement problem even though it is a 3-D flow. We are currently working on a model which will also refine in the vertical.

The adaptive method is based on the idea of using multiple, component grids on which the partial differential equations are solved. Refined grids are created or removed based on a Richardson-type estimate of the truncation error in the finite difference solution. The goal is to maintain a given accuracy in the numerical solution for a minimum amount of work. A complete description of the method can be found in [2] and [6].

The solution procedure for the adaptive grid method is as follows. We begin with a solution on a coarse grid that has been integrated to some time t. The error introduced through the use of the numerical procedure is estimated at grid points and where these errors are judged to be too large the points are flagged. Then 2-D rectangular grids with finer meshes are fitted around these flagged points. These fine grids, which are completely separate from the coarse grid, may have orientations differing from the coarse grid. Initial and boundary conditions for these new fine grids are interpolated from the coarse grid solution and the fine grids are integrated along with the coarse grid to a new time $t + \Delta t_c$, where Δt_c is the coarse grid time step. Smaller time steps are taken on the fine grids to keep $\Delta x/\Delta t$ constant. The coarse grid solution is then updated with the more accurate fine grid solutions. The

updating consists of replacing the coarse grid solution at coarse grid points which lie inside of the fine grids with the appropriately averaged fine grid values. Truncation error may also be estimated on the fine grids and still finer grids introduced. Thus, there can be several levels of fine grids. The errors on the grids can be re-estimated every few time steps and then new fine grids can be created and old fine grids removed.

The large errors in the numerical solution are usually associated with sharp variations in the solutions, e.g., at fronts or other disturbances. By re-estimating the error after several time steps and regridding we create a scheme whereby the fine grids track the disturbances. The fine grids can be arbitrarily oriented and may overlap. This allows them to line up with the disturbances which minimizes the size of the fine grids and provides for better resolution in the numerical scheme.

The program can be viewed as consisting of three components: (1) a data structure, (2) a solver, and (3) management routines. Due to the constructs of FORTRAN the data structure is fixed. It stores the solution vectors for all grids and information about these grids. The information and where it is stored are altered by the management routines. These routines also pass to the solver the locations of the solution vectors in the data structure. The overall program can be viewed as the interaction between the solver and data structure with the management routines controlling this interaction and performing the necessary intergrid communication (setting boundary conditions and updating).

The key algorithms are those which perform the integration, error estimation, and grid generation. To illustrate how they work, consider the grid arrangement shown in Fig. 1. There are several ways to advance the solution by one Δt_c on the grids. These are dependent on the interface conditions between the grids. For exam-

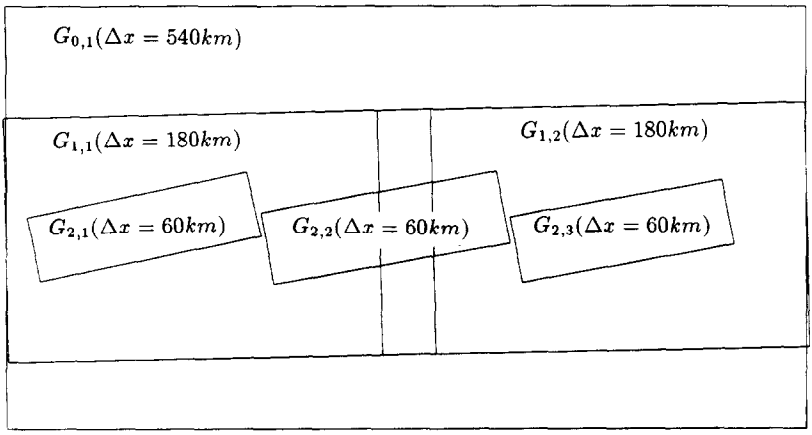
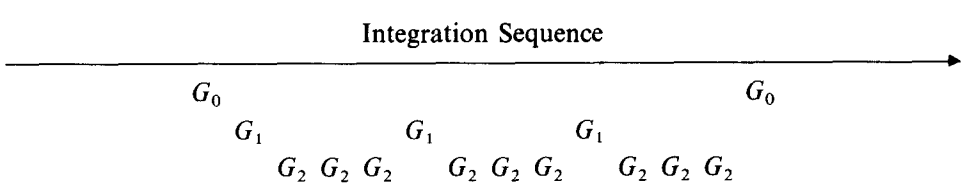


FIG. 1. Adaptive run with two levels of refinement. G_0 is the coarse grid, G_1 fine and G_2 finer resolution grids.

ple, with a refinement ratio $r = 3$ ($r = h_c/h_f$ = the ratio of the coarse grid Δx to the fine grid Δx), the integration order (from coarsest to finest) is



where G_i are grids at level i with i increasing for finer levels. Grids at level i are integrated r times as often as grids at level $i-1$ but with $\Delta t_i = \Delta t_{i-1}/r$, thus all are integrated to the same point in time. The order of integration assures that all fine grids will have sources for boundary values.

Error estimation is also repeated and solutions from the fine grids must be passed to the coarse grids. Errors are estimated and grids replaced on each level after a number of time steps specified by the user. Grids at level i will be replaced, based on an error estimate on level $i-1$ grids, r times more often than grids at level $i-1$. Solutions on level i are updated with those on a higher level when the higher level solutions have reached the same point in time.

Errors on the various grids are estimated using a method based on Richardson extrapolation. If the solution is smooth the local truncation error can be expressed as

$$\begin{aligned} u(x, t+k) - Q_h(u(x, t)) &= k(k^{q_1} a(x, t) + h^{q_2} b(x, t)) \\ &\quad + kO(k^{q_1+1} + h^{q_2+1}) \\ &= \tau + kO(k^{q_1+1} + h^{q_2+1}), \end{aligned} \tag{2.1}$$

where q_1, q_2 are the orders of accuracy in time and space, Q_h is an operator representing the finite difference scheme and defined as $u(x, t+k) = Q_h(u(x, t))$, and h and k correspond to Δx and Δt, respectively. If we let Q_{2h} be the same difference operator with step sizes of $2k$ and $2h$ and if we assume that the order of accuracy of the time and space differencing are equal and also that the solution is sufficiently smooth, then we can derive the following expression for the leading order term τ of the truncation error:

$$\frac{Q_h^2(u(x, t)) - Q_{2h}(u(x, t))}{2^{q+1} - 2} = \tau + O(h^{q+2}) \tag{2.2}$$

where Q_h^2 is the operator Q_h twice applied to $u(x, t)$.

This gives us an estimate of the local truncation error at time t. The procedure is to take a giant step based on mesh widths $2h$ and $2k$ using the solution at time t and then to compare it with the solution found by taking two regular steps. The same solver used to integrate the equations can be used to estimate the error. The estimator is independent of the finite difference method and is also independent of

the PDE. The method does not produce accurate estimations of τ for nonsmooth solutions but no problem is envisaged because the estimates will probably be large in these regions anyway and will lead to the desired regridding.

Estimating the error at grid points is the first step in the regridding procedure. The regridding procedure is

(1) flag points needing refinement,

(2) cluster the flagged points,

(3) fit rectangular grids around the clustered points, and

(4) repeat if necessary.

Gridpoints are flagged if the estimated error exceeds a user specified value. Clustering the flagged points serves two purposes. First, it separates spacially distinct phenomena. In many problems there are often multiple shocks or fronts. These features will then be on different grids. The second purpose is to subdivide points when one large region should be fit with several grids.

Clustering the grid points and fitting rectangles to the clusters are the most difficult parts of the regridding task. Inexpensive clustering algorithms are rarely satisfactory for both clustering purposes. For this reason, a simple algorithm is used

nearest neighbor principle. If a point has any other point within some specified minimum distance then these points are in the same cluster. This method works well for the first purpose, but very poorly for the second. The more complex methods use minimum spanning trees or nearest neighbor graphs. These structures connect the points in an organized way. An iterative method is then used which merges points with core groups of points and attempts to maximize the efficiency of a grid. The efficiency of a grid is a measure of how large the grid is compared to how many flagged points the grid contains.

The last task the regridding algorithm must complete is to actually fit the rectangles to the clusters. There are several methods which will do this and some produce, on the average, more efficient rectangles than others. In the algorithm a simple, less expensive method is used because it works well and also must frequently be used in the clustering routine. The method fits rectangles by computing a least-squares fit line to the cluster of points. This line is the principal axis of the rectangle and an orthogonal line will be parallel to the other axis. It is then an easy matter to compute where the sides and the corners of the rectangles should be, though this is the most expensive part of fitting the rectangle. Finally, the rectangle is enlarged so as to provide a buffer zone between the flagged points (the phenonmena) and the rectangle (fine grid) boundaries.

The data structure stores two kinds of information, descriptions of grids and the grid solution vectors. It is natural to think of these grids in the context of a tree, with the coarse grid being at the root of the tree. At each node of the tree lies a grid. Each grid (node) is characterized by

(1) grid location,
(2) grid point specifications,
(3) level in tree,
(4) offspring pointer,
(5) sibling pointer,
(6) parent pointer,
(7) intersection pointer,
(8) pointer to the next grid at same level,
(9) time to which grid has been integrated, and
(10) index in storage array for solution on grid.

This information is stored for each grid at the nodes of the tree. Figure 2 shows an example of the tree for the grid system of Fig. 1.

All solution vectors are stored in one array. This array is managed as a linked list of used and available blocks of storage. Storage is allocated in contiguous blocks by scanning the list of available blocks, taking the fist block that is large enough, and returning whatever space is unused. Reclaimed space can be re-inserted into the list and reused. The structure of FORTRAN does not allow for dynamic memory allocation outside the program, hence all storage is defined initially.

We have described the algorithms which control the integration sequence, error estimation, clustering, gridfitting, and data management. The program is constructed modularly. The data structure and the methods used to alter it can be accessed by all routines. The user has only to supply an integration routine (a solver) which solves the equations that he is interested in. Purposes necessitating changes usually require altering only one or a few modules, and not the entire code.

Our use of the adaptive grid method and the program just described is greatly facilitated by code modularity. The program contains the necessary algorithms and data structures to carry out the adaptive method outlined earlier. Many of these algorithms have been borrowed from the fields of computer science, mathematics

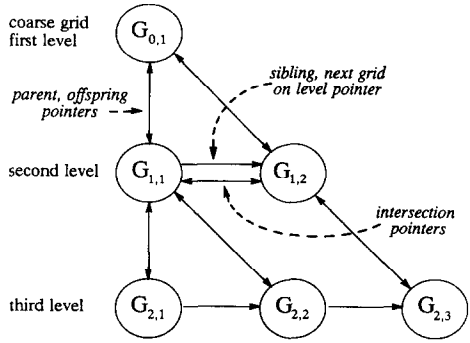


FIG. 2. Tree for grids of Fig. 1. Included are some of the pointers listed in the text.

and other disciplines and it is their application for numerical weather prediction that is new.

3. Primitive Equations

3.1. *Equations*

We solve the Euler equations for dry, adiabatic large-scale atmospheric flows. The set commonly used for studying compressible fluid flows has as dependent variables u, v, and w, the horizontal and vertical velocities, and T, p, and ρ, the temperature, pressure, and density, with geometric coordinates x, y, and z. We make several changes to this set.

Large-scale flows are very nearly hydrostatic, and making this approximation reduces the z momentum equation to the hydrostatic equation with the added benefit of removing sound waves from the solution. We also recast the system using the nondimensional pressure σ in place of the vertical coordinate z. The coordinate σ is defined as

$$\sigma = p/\pi$$

where π is the surface pressure. Thus, at the surface $p = p_s = \pi$, $\sigma = 1$ and at the top of the atmosphere $p = 0$ and $\sigma = 0$. The vertical coordinate σ has a range $0 \leqslant \sigma \leqslant 1$.

Using the definition $\pi = p_s$ and $\sigma = p/\pi$ we can write the equation of state as $\sigma\pi = \rho RT$ and use it to eliminate the density ρ from the equations. The final step is the introduction of the variable $\phi = gz$, the geopotential, which replaces z as a dependent variable. The reduced Euler set, known as the hydrostatic primitive equations, is

$$\frac{\partial}{\partial t}(\pi u) = -\frac{\partial}{\partial x}(\pi u^2) - \frac{\partial}{\partial y}(\pi uv) - \frac{\partial}{\partial \sigma}(\pi u\dot{\sigma}) + \pi fv - \pi\frac{\partial \phi}{\partial x} - RT\frac{\partial \pi}{\partial x} + \pi F_x, \tag{3.1}$$

$$\frac{\partial}{\partial t}(\pi v) = -\frac{\partial}{\partial x}(\pi vu) - \frac{\partial}{\partial y}(\pi v^2) - \frac{\partial}{\partial \sigma}(\pi v\dot{\sigma}) - \pi fu - \pi\frac{\partial \phi}{\partial y} - RT\frac{\partial \pi}{\partial y} + \pi F_y, \tag{3.2}$$

$$\frac{\partial \phi}{\partial(\ln \sigma)} = -RT, \tag{3.3}$$

$$\frac{\partial}{\partial t}(\pi c_p T) = -\nabla_\sigma \cdot (\pi c_p TV) - \frac{\partial}{\partial \sigma}(\pi c_p T\dot{\sigma}) + \pi Q + RT\omega + \pi F_T, \tag{3.4}$$

$$\frac{\partial \pi}{\partial t} = -\nabla_\sigma \cdot \pi V - \pi\frac{\partial \dot{\sigma}}{\partial \sigma} + F_\pi \tag{3.5}$$

where

$$V = u\mathbf{i} + v\mathbf{j},$$

$$R = \text{gas constant},$$

$$\text{geopotential } \phi = gz,$$

$$\dot{\sigma} = \partial\sigma/\partial t,$$

$$\omega = dp/dt = \pi\dot{\sigma} + \sigma(\partial\pi/\partial t + V \cdot \nabla_\sigma \pi).$$

The independent variables are x, y, σ, and t and the dependent variables are u, v, $\dot{\sigma}$, ϕ, π, and T. Equations (3.1) and (3.2) are the u and v momentum equations, (3.3) is the first law of thermodynamics, and (3.4) is the hydrostatic equation. Equation (3.5) is the transformed continuity equation. A complete derivation of these equations can be found in Holton [12] and Haltiner and Williams [9].

The need to make several other assumptions arises when these equations are used. The terms F_x, F_y, and Q are forcing or source terms that account for processes not explicitly accounted for in the dynamics equations. The terms in the momentum equations theoretically include the effects of diffusion, both turbulent and molecular. In the thermodynamic equation Q represents latent and radiational heating and cooling, fluxes of heat from the boundaries and in essence all diabatic effects. Major assumptions underlying models often are found in the parameterizations of these terms.

In large-scale flows the effects of viscosity and turbulence have negligible contributions to the forcing terms F_x and F_y. Fourth order horizontal diffusion is included only to stabilize the numerical solution. This stabilizing diffusion term is also calculated for F_π and F_T. There is no vertical diffusion in any of the equations. We are solving for adiabatic flow; the model does not consider radiation effects and there are no sources of heat.

Statically unstable temperature profiles are addressed using convective adjustment parameterizations. The use of the hydrostatic assumption in the equations removes the mechanism which allows the atmosphere to respond to instabilities in a vertical air column. Convection (vertical air motion) often takes place in the atmosphere as a response to an instability in an air column. The instability can be thought of as the presence of more dense air above less dense air. This instability cannot give rise to vertical motions because there is no feedback to a vertical momentum equation where $\partial w/\partial t$ is driven by the forcing. A common approach is to parameterize convection that arises from these instabilities through the use of convective adjustment schemes. We describe a simple dry convective adjustment scheme used in this solver. For a more detailed explanation see Haltiner and Williams [9].

Air parcels can be characterized by their lapse rates γ

$$\gamma = -\frac{\delta T}{\delta z}.$$

If in a dry atmosphere the lapse rate at a point is greater than the dry adiabatic lapse rate, then a vertical adiabatic displacement of a parcel at this point will be unstable. This will produce vertical convection in the region. A simple way to parameterize this process is as follows:

(1) Calculate the large-scale fields without considering instabilities.

(2) Calculate the actual lapse rates and dry adiabatic lapse rates in a column.

(3) Compare the lapse rates:

$\gamma \geqslant \gamma_d$ *stable* $\delta T = 0$,

$\gamma > \gamma_d$ *unstable* $\delta T \neq 0$,

γ_d is the dry adiabatic lapse rate,

δT is temperature change in the column due to convection.

In the first case nothing is done, for the column is stable. In the second case the vertical temperature profile is adjusted to a neutral or slightly stable lapse rate γ_d subject to the condition that total potential energy is conserved, i.e.,

$$\int_{Z_b}^{Z_t} c_p \, \delta T \rho \, dz = \frac{c_p}{g} \int_{p_t}^{p_b} \delta T \, dp = 0$$

where b and t represent the bottom and top of the unstable layer and c_p is the specific heat of air at constant pressure. The scheme assumes that convection causes potential energy to be converted to kinetic energy which is eventually converted into internal energy.

3.2. *Discretization*

On a horizontal plane the grid used in this model is the C grid described in Arakawa and Lamb [1]. The C grid is shown in Fig. 3. It resolves shorter waves well, helps accurately represent wave group velocities and amplitudes, and possesses

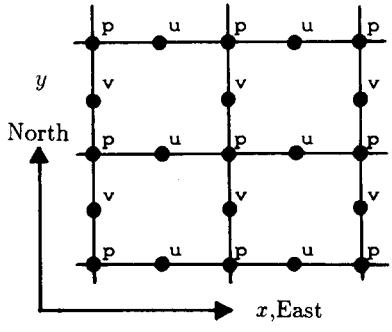


FIG. 3. Grid C. The variables T, ϕ, and π are found at p points. u and v are the velocities in the x and y directions (east and north). respectively.

very good geostrophic adjustment properties. Vertical differencing takes place on the grid shown in Fig. 4. The variables shown on the horizontal grid are carried between σ levels; σ and its time derivative are carried at σ levels. The surface pressure π and the surface geopotential ϕ_s are defined at the surface. The σ axis is taken as those values at the surface directly (vertically) below the point at the surface.

We first consider how the equations are differenced on the horizontal σ surfaces on the C grid. Centering a control volume over a u velocity point at (x, y), where $x = i \cdot \Delta x$ and $y = j \cdot \Delta y$, we can denote fluxes of u-momentum through the control volume faces in the x and y direction as F^u and G^u. The discretization for the horizontal advection terms in the u-momentum equation is

$$\frac{\partial}{\partial x}(\pi uu) + \frac{\partial}{\partial y}(\pi uv) = \frac{1}{\Delta x}(F^u_{i+1/2,j} - F^u_{i-1/2,j}) + \frac{1}{\Delta y}(G^u_{i,j+1/2} - G^u_{i,j-1/2})$$

where

$$\begin{aligned} F^u_{i+1/2,j} &= \tfrac{1}{8}(u_{i+1,j} + u_{i,j}) \\ &\quad \cdot (u_{i,j}(\pi_{i-1/2,j} + \pi_{i+1/2,j}) + u_{i+1,j}(\pi_{i+1/2,j} + \pi_{i+3/2,j})), \\ G^u_{i,j+1/2} &= \tfrac{1}{8}(u_{i,j+1} + u_{i,j}) \\ &\quad \cdot (v_{i+1/2,j+1/2}(\pi_{i+1/2,j} + \pi_{i+1/2,j+1}) \\ &\quad + v_{i-1/2,j+1/2}(\pi_{i-1/2,j} + \pi_{i-1/2,j+1})). \end{aligned}$$

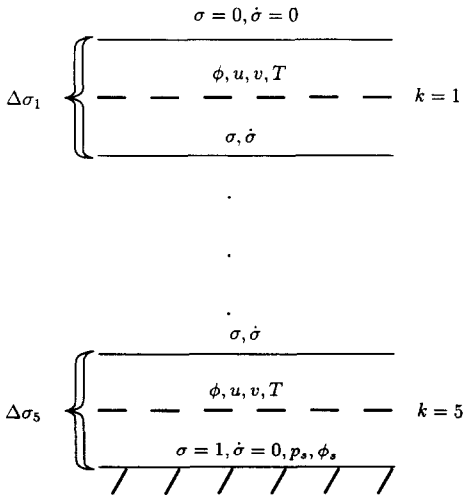


FIG. 4. Vertical structure of the finite difference grid in the sigma (σ) coordinate system for a 5-level model.

Vertical (σ) derivatives, for example $(\partial(\pi u\dot{\sigma})/\partial\sigma)$, are computed as

$$\frac{\partial}{\partial\sigma}(\pi u\dot{\sigma}) = \frac{\overline{\pi_{i,j}}}{2\Delta\sigma_k}\left(\overline{\dot{\sigma}_{i,j,k+1/2}}(u_{i,j,k+1}+u_{i,j,k})\right.$$
$$\left.-\overline{\dot{\sigma}_{i,jk-1/2}}(u_{i,j,k}+u_{i,j,k-1})\right)$$

where the overbar denotes an averaged value for π and $\dot{\sigma}$ at the points (i, j). The velocity u is averaged to arrive at a value of u at $k \pm \frac{1}{2}$.

Similarly, we can center a control volume about a π point and denote "mass" fluxes through the surfaces as F, G, and $\dot{S}$. The continuity equation (3.5) is finite differenced as

$$\frac{\partial}{\partial t}\pi_{i,j} + \frac{1}{\Delta x}(F_{i+1/2,j,k} - F_{i-1/2,j,k}) + \frac{1}{\Delta y}(G_{i,j+1/2,k} - G_{i+j-1/2,k})$$
$$+\frac{1}{\Delta\sigma_k}(\dot{S}_{i,j,k+1/2} - \dot{S}_{i,j,k-1/2}) = 0$$

where

$$\dot{S}_{i,j,k+1/2} = \pi_{i,j}\dot{\sigma}_{i,j,k+1/2},$$
$$F_{i+1/2,j,k} = \tfrac{1}{2}u_{i+1/2,j,k}(\pi_{i,j} + \pi_{i+1,j}),$$
$$G_{i,j+1/2,k} = \tfrac{1}{2}v_{i,j+1/2,k}(\pi_{i,j+1} + \pi_{i,j}).$$

The remaining terms in Eqs. (3.1)–(3.4) are differenced in a similar manner. The overall scheme will conserve mass (disregarding the diffusion term in the pressure tendency equation) but will not exactly conserve total (kinetic + potential) energy.

The leapfrog method is used to integrate the spacially differenced equations in time. The method is explicit, second order in time and possesses good phase and amplitude wave propagation characteristics. Equations (3.1)–(3.3) are marched forward in time. The surface pressure π is found by integrating the continuity equation (3.5) vertically at each π gridpoint. It is only when integrating this equation that the vertical boundary conditions ($\dot{\sigma} = 0$ at $\sigma = 0, 1$) are needed. The vertical integration of (3.5) results in

$$\frac{\partial\pi}{\partial t} = \sum_{k=1}^{K} \nabla_\sigma \cdot (\pi V)\,\Delta\sigma_k.$$

The $\dot{\sigma}$'s can be found by integrating (3.5) down to the required level. The geopotential ϕ is found by integrating the hydrostatic equation (3.4).

The stability of the leapfrog scheme is limited by the CFL condition

$$\frac{c\,\Delta t}{\Delta x} \leqslant \frac{1}{\sqrt{2}}.$$

The fastest waves in these equations are the gravity waves, where the external gravity wave travels approximately an order of magnitude faster than the meteorological waves of interest. These waves are important in the geostrophic adjustment process, thus they cannot be filtered out of the equations without some adverse affects. The gravity waves severely limit the maximum time step which can be used.

The leapfrog scheme cannot be used to begin an integration because the variables at time $t - \Delta t$ are not known. We use the forward Euler scheme to start the integrations. We also use the forward Euler scheme to begin integrations on newly created fine grids.

3.3. *Boundary Conditions*

This problem is posed as an initial boundary value problem (IVBP). The initial values for the velocities u and v, the surface pressure p_s or π, and the temperature T or the geopotential ϕ are necessary as initial conditions for the model.

The boundary conditions which must be specified are those in the vertical (at the top and bottom sigma layers) and in the horizontal at the lateral boundaries. The choice of the sigma coordinates leaves us with very simple boundary conditions in the vertical. The conditions are

$$\dot{\sigma} = \frac{d\sigma}{dt} = 0$$

at both the top ($p = 0$, $\sigma = 0$) and the bottom ($p = p_s = \pi$, $\sigma = 1$). At the surface ϕ_s is specified and this serves as the lower boundary condition for the integration of the hydrostatic equation (3.4). At the upper and lower boundaries free slip conditions are applied for the u and v velocities and no flux conditions are applied for the temperature.

The lateral boundary conditions may vary with the application of the model. Our test cases are for flow in a free slip, east–west periodic channel. The north–south boundaries are no flux ($v = 0$). We use these boundary conditions on the base (coarse) grid.

For fine grid boundary conditions we specify u, v, T, and π at the boundaries using bilinearly interpolated values from another grid. The interpolation is in both space and time. In this regard, we are choosing to apply continuity conditions at the fine grid boundaries as opposed to treating each fine grid as a separate initial boundary value problem where we would apply open boundary conditions.

3.4. *Primitive Equations Considerations*

Oliger and Sundström [16] have shown that the primitive equations are ill-posed for the initial boundary value problem with open boundaries. More specifically, they state that "local, pointwise boundary conditions cannot yield a well-posed problem for the open boundary problem for the hydrostatic equations." This statement holds for the continuous equations and is based on examinations of

the behavior of appropriate norms of the solution. It logically follows that a discrete approximation of these equations cannot have a norm which behaves reasonably if it accurately approximates an ill-posed problem (Thomée [17]).

Limited area modellers have traditionally circumvented the problem of ill-posedness and resulting exponential error growth by including horizontal dissipation in their models and, more importantly, by using sponge-type boundary conditions and increased horizontal dissipation close to the boundaries. This leaves open the question of exactly what equations are being solved and the accuracy of the limited-area model solutions. Our example calculations are for flow in a periodic channel. The primitive equations are weakly well posed for these boundary conditions.

The question also arises as to whether the primitive equations are ill-posed for solution on the local refinements (fine grids). This would be the case if we chose to treat a fine grid as a seperate IVBP and use boundary conditions appropriate for the equations. Instead we have chosen to use continuity conditions and interpolate all data from one grid on to the boundary of another. Now, we must consider the accuracy and stability of these conditions.

There are no analyses for the primitive equations or for nonlinear hyperbolic equations concerning the accuracy or stability of our boundary scheme. For a 2-D model hyperbolic equation Berger [3] has shown that using leapfrog with overlapping grids and grid refinement in both time and space is stable. Our adaptive integrations of the primitive equations have also proven to be stable.

Accuracy and conservation are related issues and there are few results concerning overlapping boundaries which are rotated with respect to each other. Berger [4] derives a conservative boundary scheme for use in solving hyperbolic systems of conservation laws on 2-D rotated rectangles. We do not implement the scheme here and know of no implementation of it. Henshaw [11] has found that when solving elliptic equations on overlapping grids one must use boundary value interpolation schemes that are at least as accurate as the interior numerical scheme and in some cases at least of one order higher accuracy. Browning *et al.* [7] show that solutions of hyperbolic equations on embedded grids of different resolutions may produce phenomena similar to that of the propagation of waves through materials of different densities. There can be interference of waves which have passed through refined regions with those that have not. This interference is a weak instability and obviously results in a grossly inaccurate solution.

We address the issues of accuracy and conservation in how and where we decide to place fine grids. In the adaptive scheme, fine grid boundary values are interpolated bilinearly from other grids. We use an estimate of the error in the solution to place the fine grids and periodically re-estimate and replace the fine grids so that regions of high error always remain contained in the fine grids. The regions of high error (the disturbances) must never be allowed to pass through fine grid boundaries onto the coarse grid. Thus fine grid boundaries always lie in areas where the solution error is low, i.e., where bilinear interpolation will introduce only small errors. Our integrations indicate that in this context the use of bilinear interpolation is sufficient to ensure accurate and stable solutions.

4. Test Case Results

In this section we present results for two flow simulations using the adaptive primitive equations model. We present no detailed data concerning run times for different models. Our calculations show a breakeven point for using adaptive methods over a single fine grid of 50–60% fine grid coverage in a 2-level adaptive model run. This model is a research model and with some optimization we feel that the breakeven point could be improved to 70–80%.

4.1. *Barotropic Cyclone*

The first test case for the adaptive atmospheric solver is the simulation of a barotropic cyclone (no vertical variation) which is being advected by an easterly flow in an east–west periodic channel. The equations are solved on an f-plane (f = constant = $5.0 \times 10^{-4}\,s^{-1}$). There is no surface friction or energy sink other than the diffusion used to stabilize the computations, hence the solution should show the cyclone being advected toward the west with little change in its appearance.

The initial conditions are shown in Fig. 5. The surface pressure field depicted in Fig. 5 is found by solving a Poisson equation which is derived by taking the divergence of the momentum equation and assuming that the divergence of the horizontal velocity field is zero. The Poisson equation is

$$\nabla^2\pi = -\frac{1}{T_{\text{mean}}}\left(\frac{\partial}{\partial x}\left(\frac{\partial(uu)}{\partial x}+\frac{\partial(uv)}{\partial y}-fv\right)+\frac{\partial}{\partial y}\left(\frac{\partial(uv)}{\partial x}+\frac{\partial(uv)}{\partial y}+fu\right)\right).$$

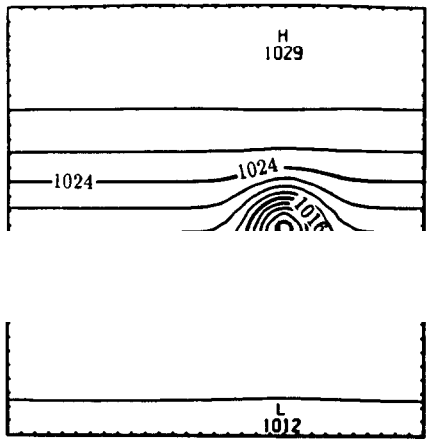


FIG. 5. Initial conditions for the barotropic cyclone. Surface pressure is in millibars. Winds travel counterclockwise around the low pressure. The cyclone is advected toward the left by a zonal wind.

The initial wind fields consist of a zonal wind over which a cyclone is superposed. The zonal flow has the form

$$u(y) = U_0 \cdot \sin^2\left(\frac{\pi y}{L_y}\right)$$

where L_y is the width of the channel and $0 \leqslant y \leqslant L_y$. The symmetric cyclonic wind field superposed over the zonal flow has the form

$$U_T(x, y) = U_c \sqrt{\frac{(x-x_0)^2 + (y-y_0)^2}{L_c^2}} \cdot \exp\left[\frac{1}{2}\left(1 - \frac{(x-x_0)^2 + (y-y_0)^2}{L_c^2}\right)\right],$$

where U_T is the tangential wind velocity of a cyclone centered at (x_0, y_0). The results which follow are for the case $U_c = 20$ m/s, $U_0 = -10$ m/s, and $L_c = 350$ km.

Figure 6 is the solution after 3 days for a coarse grid run with $\Delta x = \Delta y = 180$ km and $\Delta t = 900$ s, and Fig. 7 an adaptive solution with a refinement ratio of three and one level of refinement. These results clearly show that the coarse grid does not sufficiently resolve the cyclone resulting in its premature decay while the adaptive run does resolve the cyclone and shows very little decay. The fine grid is needed and the adaptive calculation is successful.

Figure 7 also shows the placement of the fine grids in the channel. The error estimation and regridding was performed every 9 h. There is only one fine grid in the channel at any one time and the plots show all the grids that were placed. The fine grids are usually just slightly larger than the cyclone and the regridding occurs often enough so that the fine grid tracks the cyclone. Also note that the fine grids in this simulation are not aligned with the coarse grid. Another simulation was performed where the fine grids were aligned and had points coincident with the coarse

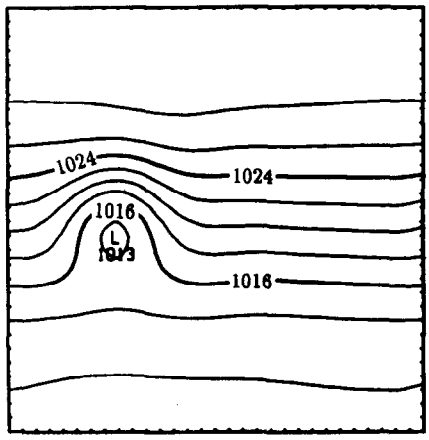


FIG. 6. Cyclone after 3 days with the integration carried out on the coarse grid with $\Delta x = 180$ km.

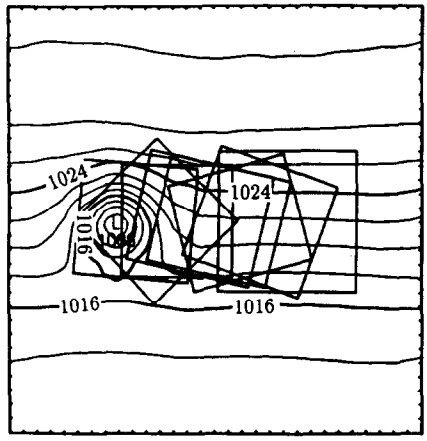


FIG. 7. Cyclone after 3 days using adaptive model. There is only a single fine grid in the region at any time. $\Delta x_{\text{coarse}} = 180$ km, $\Delta x_{\text{fine}} = 60$ km. Fine grids used in the calculation are displayed.

grid. The simulations differ only slightly, this demonstrates that the orientation of the fine grids has little effect on the solution.

Fine grid placement depends on an error estimate of the coarse grid solution as described in Section 2. Only the velocity error estimates were used to place fine grids. We will also present error estimates for the surface pressure π and temperature for the baroclinically unstable jet but they are not yet used for fine grid placement. Figure 8 shows a typical error estimate for the u velocity field along with the fine grid placed over a set of flagged points derived from the error estimate. The error in the u velocity is normalized by $|U|_{\max} = 27$ m/s and the dimensionless error tolerance for flagging point is $\tau = 0.025$ (see Eq. (2.2)). For this flow the error

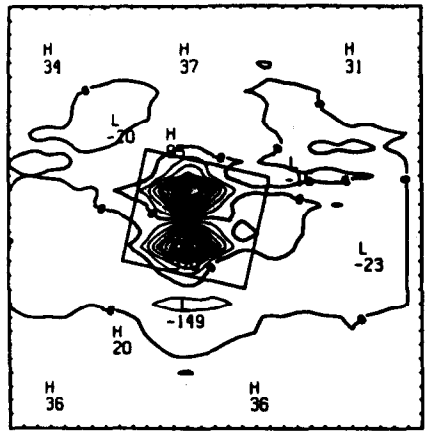


FIG. 8. Typical error estimate of the u velocity (dimensionless $\times 10^4$). The estimate has been normalized by $|U|_{\max} = 27$ m/s.

estimate of these fields places the fine grid over the cyclone—which is where we expect it would be necessary.

Initial conditions for fine grids are interpolated off the coarse grid or fine grids which existed before the regridding. A bicubic spline interpolation is used to obtain these initial values. Bilinear interpolation has been tried, but was found to excite spurious gravity waves (noise) which take several hours to decay.

A critical component of the adaptive solution procedure is the ability to set accurately the boundary conditions for the fine grid. During this set of simulations kinks arose in the surface pressure field close to the fine grid boundaries. Surface pressure errors are the result of errors in the mass divergence fields which themselves are the result of errors in the specification of the inflow velocity. The numerical scheme does not use the pressure gradient close to the boundary and the kinks do not have any effect on the solution. We include these results only to illustrate problems which can arise.

These errors in specifying the inflow velocities may have a greater effect when moisture is included in the model and investigation reveals that the errors arise from using a numerical scheme which is inconsistent close to the boundary. One row in from the boundary the diffusion is second order as opposed to the interior fourth order diffusion. This early version of the code also used a split explicit scheme for advancing the gravity waves, thereby allowing the use of larger time steps. Use of the scheme along with the change in diffusion next to the boundaries was found to promote growth of the kink.

The adaptive scheme attempts to minimize boundary errors in a manner not connected to the structure of the numerical scheme. First, the fine grids are made large enough so that their boundaries are in regions of small solution error, and hence the values, which are interpolated in both space and time, will have small error. Second, if the region of high error cannot be covered by a single grid then multiple, overlapping grids are used. An important addition to this is that the fine grid boundary values must be interpolated from the best source, which is often another overlapping fine grid. These tenets, along with a consistent numerical scheme, appear to be sufficient to produce reasonable solutions.

Presented next are results from the simulation of an unstable baroclinic jet. Here we use a fully explicit solver. By setting as fine grid boundary values the first two outer rows, we remove the second order diffusion from the solution, and hence, have a fully consistent numerical scheme. Errors in the specification of the inflow velocity, and hence the divergence and pressure fields, are greatly reduced.

4.2. *Baroclinically Unstable Jet*

We have chosen for our second test case to simulate the evolution of an unstable baroclinic jet which has been subjected to an initial perturbation. The disturbance which develops is commonly observed in the atmosphere and easily simulated in a channel. The flow's close analog in the atmosphere and its three-dimensional nature allow testing of several adaptive code components untested in the two-dimensional flow discussed in the previous section.

Previous simulations of this flow were performed in order to gain an understanding of the basic physical processes driving it. Several investigators in the late sixties and early seventies (e.g., Williams [18], Mudrick [15]) used models based on the primitive equations and the quasigeostrophic equations to simulate the developing baroclinic disturbance and the frontal zones associated with it. The cold and warm fronts have received extensive analytic study, most notably in Hoskins and Bretherton [13]. Those interested in the dynamics of this flow can consult these papers or for a more recent and general overview consult Holton [12].

For this simulation, we solve the equations on a β-plane (Coriolis parameter = $f = f_0 + \beta y$, $\beta = \partial f/\partial y = \text{constant}$). The grid has five layers at $\sigma = 0.1$, 0.3, 0.5, 0.7, and 0.9. The channel has a length (west–east) of 5220 km and a width (south–north) of 8640 km. The north–south velocity v is initially zero and the jet has no variation in the east–west direction. The initially geostrophically balanced jet is perturbed by altering the north–south velocity (v). After several days simulation time a single dominent wave appears in the channel. The length of the channel is then tripled, the wave repeated twice, and the channel is now of length $L =$ 15,660 km. By lengthening the channel we force the adaptive code to use multiple overlapping fine grids—as it might in actual atmospheric prediction work.

The initial conditions for this case are shown in Figs. 9–12. The jet core, which contains the maximum jet velocities, is found on the $\sigma = 0.3$ layer. This wave is clearly present in in the plot of the absolute vorticity on the $\sigma = 0.5$ layer (Fig. 9). This primary circulation is the result of the baroclinic instability which arises from the vertical shear present in the jet.

Secondary circulations (vertical and divergent motions) are driven by absolute vorticity advection and temperature advection in the primary circulation and are a result of the hydrostatic and geostrophic nature of the flow. In the lower layers, regions of cold and warm temperature advection are found, respectively, at the trough and crest of the developing wave. The temperature advection can be seen

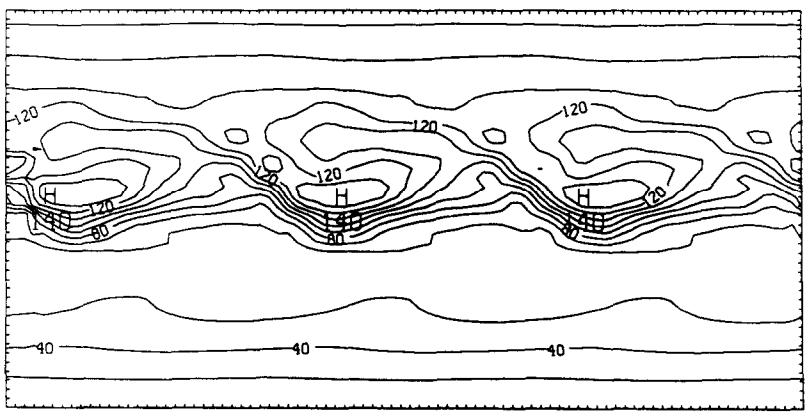
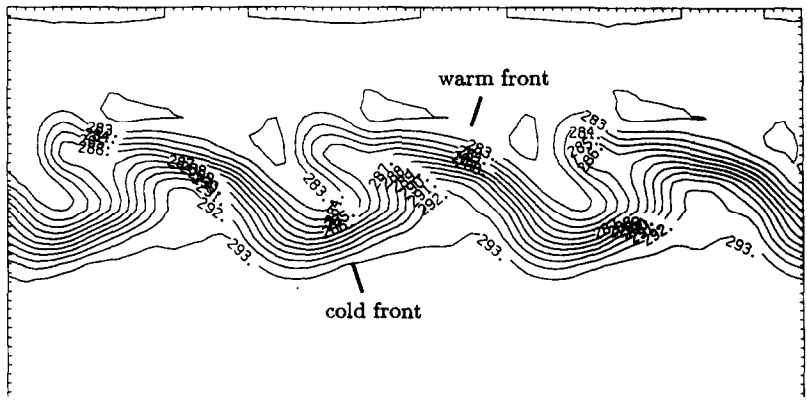


FIG. 9. Absolute vorticity ($10^{-6}\,s^{-1}$) on the $\sigma = 0.5$ surface for the baroclinically unstable wave at $t = 0$ of the adaptive run.



clearly by considerng Fig. 10, the temperature at the $\sigma = 0.9$ level and Fig. 11, the winds at the same level. Horizontal shear and horizontal deformation promote the growth of the cold and warm fronts. The shear and deformation fields intensify in the flow which develops with the development of the surface pressure lows and highs. These surface pressure features are shown in Fig. 12. Cyclonic and anticyclonic circulations form at the lower levels around the surface pressure lows and highes. These circulations are not present in the flow in the upper layers.

Poor representation of the fronts and/or of the jet stream lead to slower development or even decay of the disturbance. The strength and development of these features determines the adequacy of the grid resolution. Figures 13–15 show the results after 3 days of simulation time starting from the fields of Figs. 9–12 and

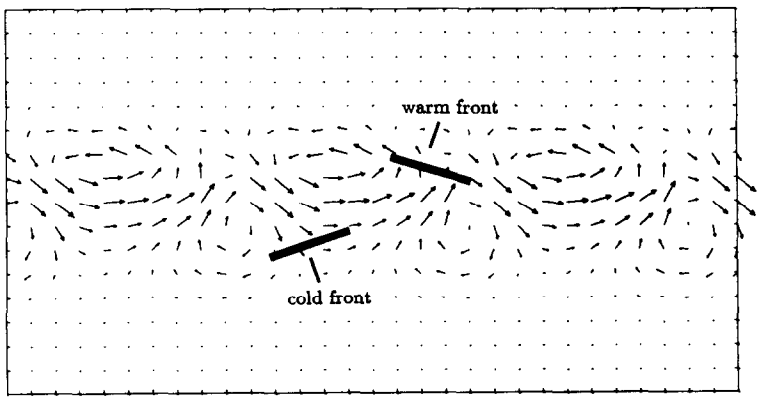


FIG. 11. Velocity vectors on the $\sigma = 0.9$ surface at $t = 0$. Note the positions of the warm and cold fronts (shown in Fig. 10) and the positions of the surface pressure highes and lows (shown in Fig. 12) with respect to the wind fields.

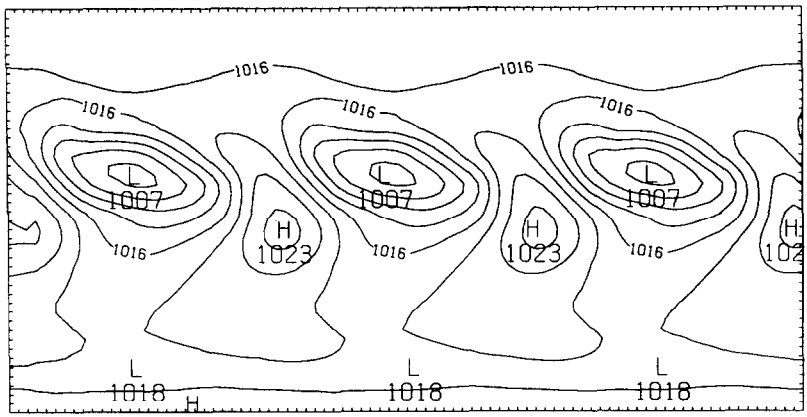


FIG. 12. Surface pressure (millibars) for the baroclinically unstable wave at $t = 0$.

encompass three different runs; a coarse grid run ($\Delta x = \Delta y = 540$ km, $\Delta t = 337.5$ s), a fine grid run ($\Delta x = \Delta y = 180$ km, $\Delta t = 112.5$ s), and an adaptive run with a coarse grid $\Delta x = \Delta y = 540$ km, one level of refinement and a refinement ratio of three (hence $\Delta x_{\text{fine}} \approx \Delta y_{\text{fine}} \approx 180$ km). In these simulations we use a value of $\tau = 0.035$ and the velocity error estimates are normalized by 10 m/s. Figure 13 shows the surface pressure for the coarse, fine, and adaptive grid runs. On the coarse grid the surface pressure highs and lows have lost strength, whereas they have not for both the fine and adaptive runs. The warm and cold fronts exhibit similar behavior. The coarse grid fronts are weakening while in the fine and adaptive grid runs they are strengthening. Again we see that the coarse grid cannot adequately represent the flow while both the fine and adaptive calculations represent well the developing baroclinic disturbance. The same resolution problem is seen in the upper level flow. The maximum absolute vorticity associated with the jet has grown from $1.4 \times 10^{-4}\,\text{s}^{-1}$ to $1.5 \times 10^{-4}\,\text{s}^{-1}$ after 3 days for both the fine and adaptive grid runs but it has diminished to $1.2 \times 10^{-4}\,\text{s}^{-1}$ in the coarse grid run. The primary wave is deepening except for the coarse grid run where it is being washed out.

Examination of the surface pressure fields in Fig. 13 show that the fine grid run results and the base grid fields for the adaptive run results do not match exactly. Indeed there are some very noticable differences, and the differences are even more pronounced in the absolute vorticity fields. This is simply because the coarse grid cannot possibly represent all the features that are representable on the fine grid. The base grid fields for the adaptive run also show the locations of the fine grids which have been placed based on an error estimate of the velocity fields. The adaptive fine grid fields match perfectly those of the fine grid run. Even the vorticity fields (Fig. 14) which are sensitive to small errors in the velocity compare extremely well with the fine grid run solution.

In these simulations the largest errors are associated with the jet. The grid-fitting routine fits just a single grid over the jet but this grid is split into two overlapping grids so that they may be accommodated by the limited workspace in the solver.

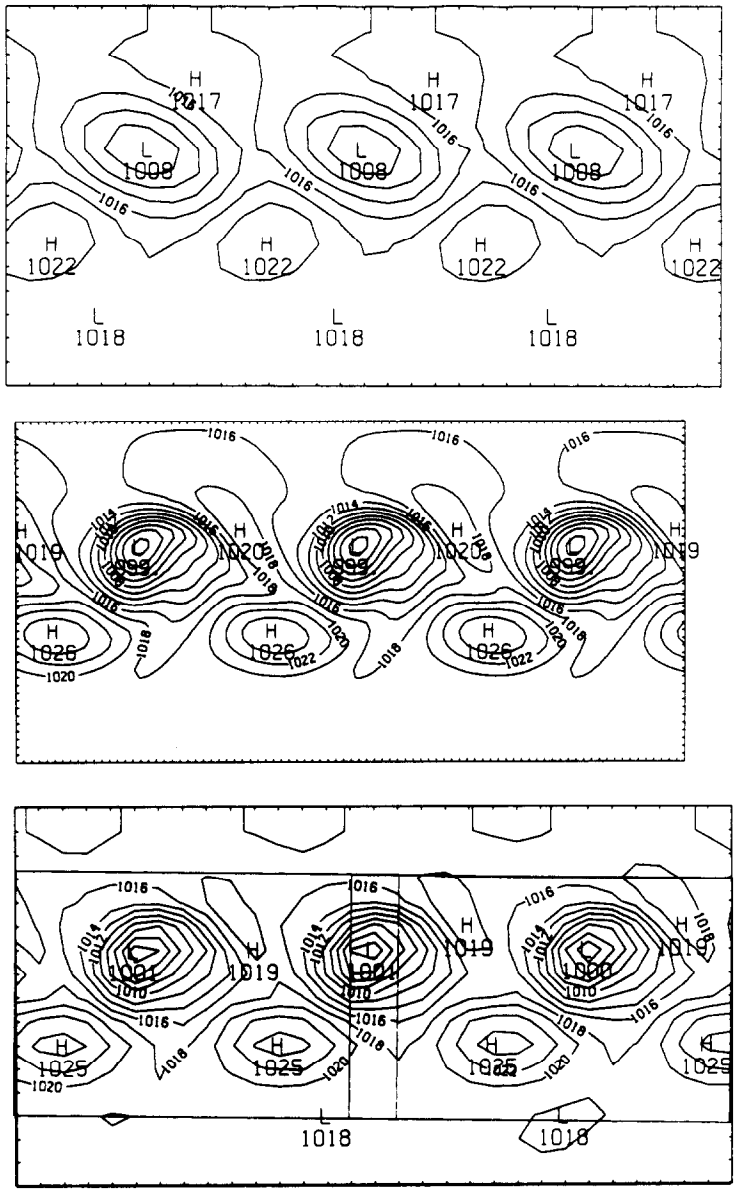


FIG. 13. Surface pressure in millibars at $t = 72$ h. Location of the fine grids in the adaptive calculation at $t = 72$ h are shown in the adaptive coarse grid plot.

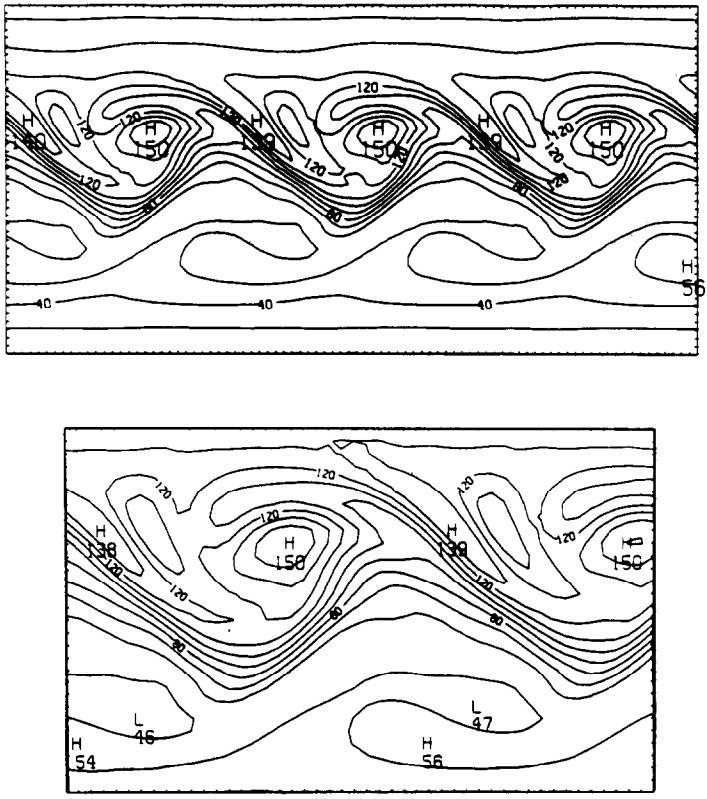


FIG. 14. Absolute vertical vorticity ($10^{-6}\,s^{-1}$) on the $\sigma = 0.5$ surface at $t = 72$ h for the fine grid run (top) and for adaptive fine grid 3 (bottom). Adaptive fine grid 3 is the left fine grid shown in the adaptive run plot of Fig. 13. The adaptive fine grid and the fine grid run absolute vorticity fields are almost identical.

The error is re-estimated every 24 h and new grids created but the positions of the fine grids change little over several days.

Here, as in the previous simulation, the fine grids are not aligned with the base grid. In the overlap region the fine grids are aligned with each other but at the periodic boundary they do not overlap in a manner where their points coincide. In both overlap regions solutions in the overlap region agree. For this to be the case fine grid boundary conditions must come from the other fine grid in regions of overlap. Using only boundary conditions interpolated off the coarse grid results in unstable solutions.

Surface pressure contours run smoothly up to the boundary on the fine grids in the adaptive run. In our previous simulation there were kinks in the surface pressure field and large errors in the divergence fields close to the boundary. By using a fully explicit scheme (which allows the setting of boundary condition values from overlapping grids in regions of overlap) and by setting the variables at the first

two interior rows as boundary values we have eliminated the kinks and large errors. On the fine grids the disturbances are well represented even in the overlap regions. No noise develops in the overlap region or at the coarse–fine grid boundaries even when these boundaries and overlaps have points which are not coincident. Indeed it is difficult to tell where the fine grid boundaries and regions of overlap are. Our disturbances remain as three identical disturbances even though different parts of the disturbances are represented in different overlap regions. Figure 15 is a plot

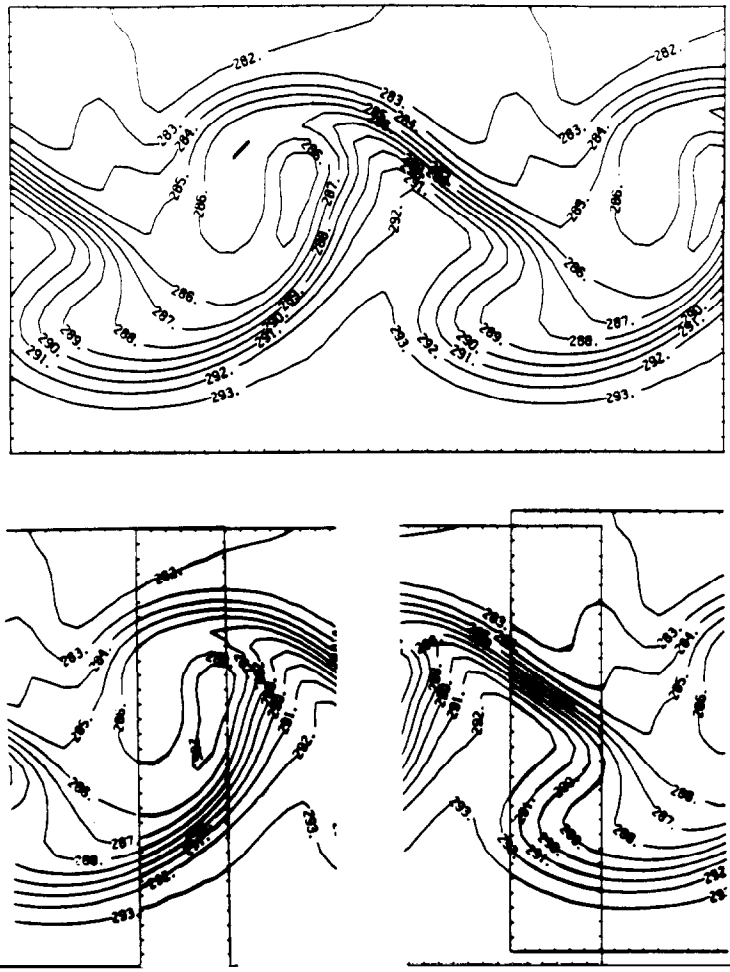


FIG. 15. Temperature (K) on the $\sigma = 0.9$ surface at $t = 72$ h for adaptive fine grids 3 (top), and the overlaps between fine grids 2 and 3 (bottom). The fine grids overlap in the center of the domain and also at the edges of the domain because of the periodic boundary conditions. The solutions agree in the overlap.

of the temperature on the $\sigma = 0.9$ level. Here a cold front and a warm front pass directly through fine grid overlaps and fine grid boundaries. These fronts are identical to those not passing through an overlap region—as they should be.

The numerical scheme used in the solver exactly conserves mass (to machine roundoff). The interpolations used in the adaptive method are not constructed so as to preserve this property. Variables on the base (coarse) grid are updated where possible with an averaged value from a fine grid and values for boundary conditions are obtained using bilinear interpolation. For a 6-day fine grid run the maximum mass fluxuation in the channel is approximately 0.007% of the total mass in the channel. The adaptive and coarse grid runs have fluctuations of 0.02%. The error in the mass is on the order of a few tenths of a percent at most, i.e., down at the level of the truncation error of the numerical scheme.

Figure 16 is a plot of the average kinetic energy (KE) of the atmosphere versus time. The integration is carried out on the base grid during the adaptive run because the changing locations of the fine grids make an "adaptive" integration difficult. The kinetic energy should increase as the disturbances grow and for the fine and adaptive grid runs it does. Also note that the oscillations are similar for the fine and adaptive runs. We have calculated the KE which includes the contribution from the fine grids in the adaptive grid run at 3 days and at 6 days. These agree well with the fine grid values.

The total energy in the channel (KE + TPE, TPE = gravitational potential energy and internal energy) is dominated by the TPE. Given that we solve an adiabatic set of equations with no energy sources or sinks and that our flow is in a enclosed

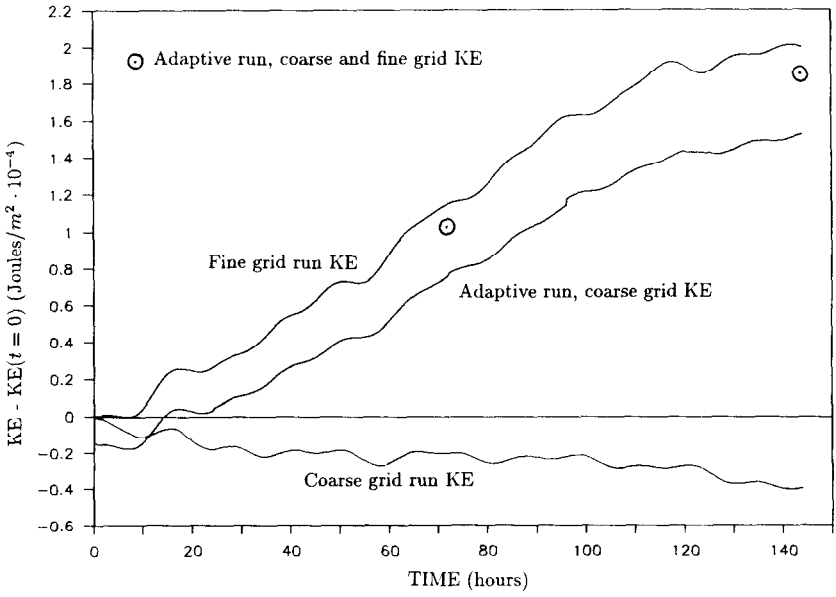


FIG. 16. Horizontally averaged kinetic energy.

channel we should find that the total energy is constant over time. This is not the case, but in all cases the gain in energy in the systems is less than a tenth of a percent of the average TPE, again on the order of the truncation error in the scheme.

One last run was made using two levels of refinement. Figure 1 shows the grid arrangement at 20 h. Regridding at level 2 was performed every 12 h and level 3 every 4 h (refinement ration $r = 3$). The coarse grid has $\Delta x = \Delta y = 540$ km, the first level of refinement has $\Delta x \approx \Delta y \approx 180$ km, and the second level of refinement has $\Delta x \approx \Delta y \approx 60$ km. Again the refinement is found to be needed around the jet and the maximum errors are at the jet core. A single fine grid run with $\Delta x = 60$ km, i.e., of the resolution of the fine grids in the adaptive run, indicates that the increase in resolution from 180 to 60 km was unnecessary. We preformed this integration as a test using multiple levels of refinement. All general conclusions found for the two-level refinement case hold when using three levels.

5. Error Estimation

Fine grids are placed in the solution domain based on an estimate of the truncation error. The procedure used to estimate the truncation error is based on Richardson extrapolation and it has been described in Section 2.

The primary advantage of the Richardson-based error estimate technique is that the form of the truncation error need not be known. The exact truncation error associated with the discretized equations (3.1)–(3.5) is complex and difficult to derive. Also the leading order truncation error terms consists of higher order derivatives which are difficult to compute with more than first or second order accuracy. The truncation error estimate obtained using Eq. (2.2) with a qth order method is accurate to $O(k^{q+1} + h^{q+1})$ which for a second order scheme such as the one used in the present solver produces a third order accurate estimate of the truncation error.

The error estimate for the u velocity field in the barotropic cyclone case (Fig. 8) show that the regions of high error to be around the cyclone. The coarseness of the grid precludes any deeper analysis. To further examine the error estimates we have estimated the truncation error for the fine grid run at time $t = 72$ h. The error estimates on the fine grid contain detail which cannot be represented on coarser grids. By comparing the Richardson estimate with computations of the exact truncation error we can judge the performance of the scheme.

First we should examine the equations and directly estimate what the size of the truncation error should be. Our scheme is second order in both space and time. Small time steps are used in the explicit scheme due to the presence of the fast gravity waves. Thus we expect that the dominent truncation error will arise due to the spatial discretization employed and hence we will focus on the error in the spatial differencing. Later comparison of the spatial truncation error with the total truncation error shows that the spatial error does dominate. We can estimate the size of the truncation error by first scaling and then nondimensionalizing

Eqs. (3.1)–(3.5) along with the truncation error. The scaling and nondimensionalization of Eq. (3.1), the u momentum equation, along with the spatial truncation error, is contained in Appendix 1.

For large-scale atmospheric flows we find that the pressure gradient and Coriolis forces must balance each other and that the advection terms are an order of magnitude smaller than these. This well-known result describes the geostrophic nature of the atmosphere, i.e., the approximate balance between the pressure gradient and Coriolis forces. Large-scale flows can often be considered in terms of adjustments to maintain an approximately geostrophic and hydrostatic state.

The finite difference scheme used to discretize the equations is second order accurate: the leading order truncation error term is of order $Ok(k^2+h^2)$. This truncation error is the sum of the truncation errors for the individual terms, all of which are of second order. In the nondimensional equations the pressure gradient and Coriolis terms have coefficients of $O(10)$ while the advection terms have coefficients of $O(1)$. If we look at the order of accuracy of the scheme we might expect that the truncation errors for the Coriolis and pressure gradient discretizations would be an order of magnitude larger than those of the advection terms. This is not the case.

The leading order terms in the nondimensional truncation errors for the Coriolis, pressure gradient, and advection terms are

$$\tau(f\pi v) \approx \frac{5\,\Delta x^2}{4L^2}\,(v'_{x'x'} + v'_{y'y'}),$$

$$\tau(\partial(\phi\pi)/\partial x) \approx \frac{5\,\Delta x^2}{6L^2}\,(\pi'_{x'x'x'} + \phi'_{x'x'x'}),$$

$$\tau(\partial(\pi uu)/\partial x) \approx \frac{\Delta x^2}{2L^2}\,(\tfrac{2}{3}\,u'_{x'x'x'} + u'_{x'x'}u'_{x'}),$$

$$\tau(\partial(\pi uv)/\partial y) \approx \frac{\Delta x^2}{4L^2}\,(\tfrac{1}{6}\,u'v'_{y'y'y'} + \tfrac{1}{2}\,u'_{y'}v'_{y'y'} + u'_{y'y'}v'_{y'} + \tfrac{2}{3}\,v'u'_{y'y'y'} + u'v'_{x'x'y'} + u'_{y'}v'_{x'x'}).$$

All primed variables are dimensionless and of $O(1)$ and their derivatives are of $O(1)$; thus all the truncation error terms are of the same relative size. Direct computations of these terms confirm this. Thus it appears we cannot ignore any of the terms when computing the truncation error for the equations directly.

Figure 17 is a plot of the truncation error in the u velocity field at $\sigma = 0.3$ (jet core) at $t = 72$ h calculated using the Richardson-based technique. Figure 18 is a plot of the truncation error associated with the spatial discretization of the pressure gradient, Coriolis, and advection terms computed using the results in Appendix 1. It is nondimensionalized by multiplying by $\Delta t/(\pi U)$, i.e., in the same way the Richardson estimate is nondimensionalized. The two estimates compare very well. Both the magnitude and distribution of the error are accurately predicted by

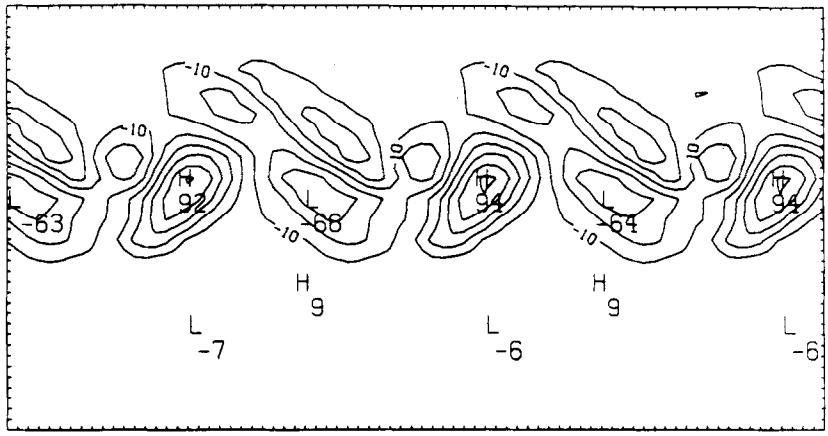


FIG. 17. Truncation error estimate using Eq. (2.2) for the u velocity field at $t = 72$ h on the $\sigma = 0.3$ surface. The estimate has been normalized by dividing by $U = 10$ m/s (dimensionless, $\times 10^5$).

the Richardson technique. We originally assumed that the spatial truncation error dominated the overall truncation error. This comparison indicates that the assumption is correct.

We have also estimated the error in the surface pressure and temperature fields. The errors in temperature are large where the time rate of change of the temperature is large. This occurs in the frontal regions where the temperature gradients and temperature advection are large. The error estimates for the surface pressure are very noisy. The noise may arise due to imbalanced initial conditions on the $2h$ error estimate grid which would lead to gravity waves and hence noise on the $2h$ grid and a noisy error estimate. Also the error in the surface pressure is small relative to the scaled surface pressure and the significance of regions of high error may be small.

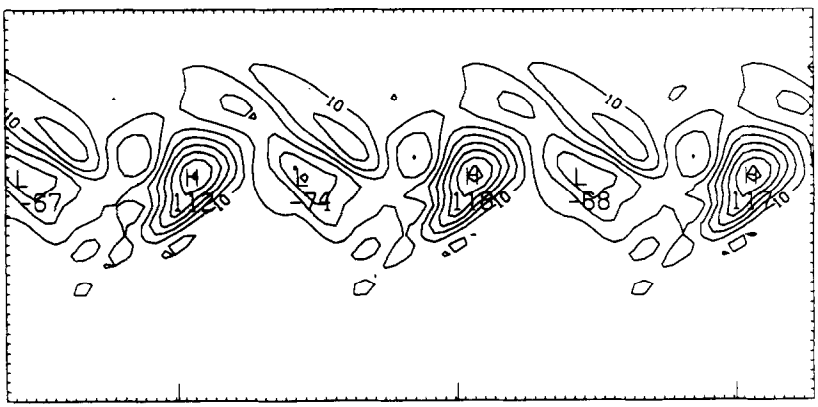


FIG. 18. Truncation error estimate computed from the formulas in Appendix 1 (dimensionless, $\times 10^5$). Normalized and nondimensionalized by multiplying by $\Delta t/(\pi U)$.

6. Conclusions

An adaptive grid refinement technique has been used to compute solutions to [illegible] cally based on a Richardson-type estimate of the local truncation error in the solution. Simulations of a barotropic cyclone and of a baroclinically unstable jet were performed to demonstrate the feasibility of using techniques of this type in NWP and similar large-scale flow computations.

Several critical components ensure an accurate, smooth solution. Numerical schemes must be consistent up to the boundaries. Changing operators close to the boundaries may cause kinks and discontinuities. When fine grids overlap, boundary values for one fine grid must come from the other. This necessitates the use of a fully explicit scheme (explicit even with respect to the boundary conditions) or some new scheme which would take into account the overlapping fine mesh constraint. Higher order interpolation techniques for use in setting the initial conditions are necessary so as not to excite gravity waves when initializing any fine grid. Given these conditions the present solver conserves mass and energy for these flow conditions and produces results which compare well with those on a single fine grid.

Richardson estimates of the truncation error compare well with directly computed truncation errors. The truncation error arising from the spatial discretization dominates the truncation error and is evenly distributed among the terms in the equations. The error is high in regions where we would expect it to be—around the cyclone and the jet stream, and produces the desired refinement. No simpler technique for estimating the truncation error is apparent.

The success of these simulations strongly supports the concept that refinement should occur only where dictated by the error in the numerical solution and that this is sufficient to improve the accuracy and overall resolution of the entire solution. Using this simple but key concept has produced the first adaptive solution of atmospheric flows and the first detailed, quantitative results concerning the error in the numerical solutions.

Finally, we wish to discuss some practical considerations concerning the use of adaptive models for numerical weather prediction. An adaptive model for large-scale atmospheric flow which includes realistic physics has not yet been developed or tested. Only in this way, by testing with actual data and with analyses of its predictive capabilities, will the true worth of the adaptive method become known. We forsee no major difficulties in adding physics to the model or in using real data.

A more immediate research problem connected with developing an operational, adaptive, large-scale atmospheric flow solver is the development of a splitting scheme for use in an adaptive model. In our simulations we used a fully explicit solver because of the need to have fully explicit boundary conditions in the case where fine grids overlap and so that continuity boundary conditions could be successfully applied. Splitting methods can allow the use of time steps 5 to 10 times larger than those used in an explicit technique with little added cost per time step. Even though the overhead intrinsic to the adaptive method is relatively small, the

development of a splitting technique for "adaptive" use may well prove critical in the development of a large-scale operational adaptive atmospheric model.

Appendix 1

We wish to estimate the relative size of the truncation errors associated with the spatial discretization. This can be done by scaling and nondimensionalizing the equations and the truncation error associated with the discretization. In this model we do not refine in the vertical and our Richardson error estimate does not estimate errors in the vertical differencing. Thus to simplify the analysis we do not consider the vertical advection terms.

For our purposes it is sufficient to analyze just one momentum equation. We scale and nondimensionalize the u momentum equation (3.1) with the following changes of variables:

$$
\begin{aligned}
\pi &= \pi_0 + \hat{\pi}\pi', & \pi_0 &= 1000 \text{ mb},\\
& & \hat{\pi} &= 10 \text{ mb},\\
\phi &= \phi_0 + \hat{\phi}\phi', & \phi_0 &= 10^5 \text{ m}^2/\text{s}^2,\\
& & \hat{\phi} &= 10^3 \text{ m}^2/\text{s}^2,\\
T &= T_0 + \hat{T}T', & RT_0 &= 10^5 \text{ m}^2/\text{s}^2,\\
& & R\hat{T} &= 10^4 \text{ m}^2/\text{s}^2,\\
u &= Uu', & U &= 10 \text{ m/s},\\
v &= Uv', & & \\
x &= Lx', & L &= 10^6 \text{ meters},\\
y &= Ly', & & \\
t &= t_0 t', & t_0 &= L/U = 10^5 \text{ s},\\
f &= f_0 f', & f_0 &= 10^{-4}/\text{s}
\end{aligned}
$$

where the primed variables are dimensionless and of $O(1)$. The scaling values are appropriate for large-scale atmospheric flows. If we substitute these into Eq. (3.1), drop the obviously lower order terms, and divide through by $\pi_0 U/t_0$ we arrive at the following nondimensional momentum equation:

$$
\frac{\partial u'}{\partial t'} = \underbrace{-\frac{Ut_0}{L}\left(\frac{\partial(u'u')}{\partial x'} + \frac{\partial(u'v')}{\partial y'}\right)}_{O(1)} \underbrace{-\frac{t_0\hat{\phi}}{UL}\frac{\partial \phi'}{\partial x'}}_{O(10)}
$$

$$
\underbrace{-\frac{\hat{\pi}t_0}{\pi_0 UL} RT_0 \frac{\partial \pi'}{\partial x'}}_{O(10)} + \underbrace{f_0 t_0 f' v'}_{O(10)}.
$$

Using these scalings we find that, for for large-scale atmospheric flows, the Coriolis term must balance the pressure gradient terms and the advection terms are an order of magnitude smaller than either of these. This just describes the geostrophic nature of the flow.

We can now use these results to scale and nondimensionalize the truncation error associated with the spatial discretization of Eq. (3.1). The leading order truncation errors associated with the terms in Eq. (3.1) are

$$\tau(\partial(\pi uu/\partial x)) = \frac{\Delta x^2}{4}\left(\tfrac{4}{3}\pi u u_{xxx} + \tfrac{7}{6}u^2\pi_{xxx} + 3u\pi_x u_{xx} + 4uu_x\pi_{xx} + 2\pi u_x u_{xx} + 2\pi_x u_x^2\right) + O(\Delta x^3),$$

$$\tau(\partial(\pi uv/\partial y)) = \frac{\Delta x^2}{4}\left(uv_{xy}\pi_x + \pi_x v_x u_y + u\pi_{xy}v_x + \tfrac{1}{6}\pi u v_{yyy} + \tfrac{1}{2}\pi u_y v_{yy} + \tfrac{1}{2}u\pi_y v_y\, y\pi u_{yy}v_y + \pi_y u_y v_y + \tfrac{1}{2}u\pi_{xx}v_y + u\pi_{yy}v_y + \tfrac{2}{3}\pi v u_{yyy} + v\pi_y u_{yy}\,\tfrac{1}{2}v\pi_{xx}u_y + v\pi_{xx}u_y + \tfrac{1}{2}uv\pi_{xxy} + \tfrac{2}{3}uv\pi_{yyy} + \pi u v_{xxy} + \pi u_y v_{xx}\right) + O(x^3),$$

$$\tau(\partial(\pi\phi)/\partial x)) = \frac{\Delta x^2}{4}\left(\tfrac{1}{6}\phi\pi_{xxx} + \tfrac{1}{6}\pi\phi_{xxx} + \tfrac{1}{2}\phi_x\pi_{xx} + \tfrac{1}{2}\phi_{xx}\pi_x\right) + O(\Delta x^3),$$

$$\tau((RT-\phi)\,\partial\pi/\partial x)) = \frac{\Delta x^2}{4}\left(\tfrac{1}{6}(RT-\phi)\pi_{xxx} + \pi_x(RT_{xx} - \phi_{xx})\right) + O(\Delta x^3),$$

$$\tau(f\pi v) = \frac{\Delta x^2}{4} f\left(\tfrac{1}{2}v\pi_{xx} + \tfrac{1}{2}\pi(v_{xx} + v_{yy}) + \pi_x v_x\right) + O(\Delta x^3).$$

We nondimensionalize the truncation errors by substituting the previously defined primed variables and dividing by $U\pi_0/t_0$. The leading order terms are given in Section 5. The nondimensionalization of the truncation errors indicate that all are of the same size, even though the respective terms associated with the truncation errors are not.

REFERENCES

1. A. ARAKAWA AND V. R. LAMB, *Methods Comput. Phys.* **17**, 174 (1977).
2. M. BERGER, Thesis, Department of Computer Science, Stanford University, Stanford, California, 1982 (unpublished).
3. M. BERGER, *Math. Comput.* **45**, 301 (1985).
4. M. BERGER, Institute for Computer Applications in Science and Engineering, Report No. 84-83, NASA Langley Research Center, 1984 (unpublished).
5. M. BERGER AND A. JAMESON, *AIAA J.* **23**, 561 (1985).
6. M. BERGER AND J. OLIGER, *J. Comput. Phys.* **53**, 484 (1984).
7. G. BROWNING, H. KREISS, AND J. OLIGER, *Math. Comput.* **27**, 29 (1973).

8. J. F. DANNENHOFFER AND J. R. BARON, "Robust Grid Adaptation for Complex Transonic Flows," AIAA 24th Aerospace Sciences Meeting Proceedings, Paper AIAA-86-0495, Reno, Nevada, January 6–9, 1986.
9. G. J. HALTINER AND R. T. WILLIAMS, *Numerical Prediction and Dynamic Meteorology*, 2nd ed. (Wiley, New York, 1980).
10. E. J. HARRISON JR., *J. Atmos. Sci.* **30**, 1528 (1973).
11. W. D. HENSHAW, Thesis, California Institute of Technology, Pasadena, California, 1985 (unpublished).
12. J. R. HOLTON, *An Introduction to Dynamic Meteorology*, 2nd ed. (Academic Press, New York, 1979).
13. B. J. HOPKINS AND F. P. BRETHERTON, *J. Atmos. Sci.* **29**, 11 (1972).
14. R. W. JONES, *J. Atmos. Sci.* **34**, 1528 (1977).
15. S. E. MUDRICK, *J. Atmos. Sci.* **34**, 869 (1974).
16. J. OLIGER AND A. SUNDSTRÖM, *SIAM J. Appl. Math.* **35**, 419 (1978).
17. V. THOMÉE, *SIAM Rev.* **11**, 152 (1969).
18. R. T. WILLIAMS, *J. Atmos. Sci.* **24**, 627 (1967).