

Higher Order Adaptive Multigrid Solution of a Fluid Flow Problem

Miao Hu
Scott R. Fulton

May 2000
(revised 9/08/2000)

Abstract

This paper describes the extension of a tropical cyclone track prediction model (MUDBAR) to fourth-order accuracy, focusing on the details of the higher-order discretization for the streamfunction and vorticity equations. For the Helmholtz problem for the streamfunction, we use a fourth-order FMG algorithm, including: (1) fourth-order compact discretization on the finest level of each V-cycle, (2) Gauss-Seidel relaxation with red-black ordering, (3) two V(1,1) cycles on each level, and (4) sixth-order initial interpolation. For the vorticity equation we prove that no fourth-order generalization of the Arakawa Jacobian is both compact and conservative; consequently, we use the noncompact fourth-order conservative Jacobian originally proposed by Arakawa. Numerical tests show that every component of the solution is fourth-order. We also compare the fourth-order model to the original second-order version, and quantify the gains in accuracy and efficiency achieved by fourth-order discretization.

Technical Report No. 2000-03
Department of Mathematics and Computer Science
Clarkson University, Potsdam, New York

*This work was supported by the
Office of Naval Research
Marine Meteorology and Atmospheric Effects Program
Grants N00014-98-1-0103 and N00014-98-1-0368*

Contents

1	Introduction	1
2	Model Description	2
2.1	Governing equations	2
2.2	Space discretization	2
2.3	Time discretization	3
3	Fourth-Order Discretization	4
3.1	Streamfunction equation	4
3.2	Multigrid algorithm	5
3.2.1	Relaxation sweeps	5
3.2.2	Control algorithm	6
3.2.3	Summary	7
3.3	Vorticity equation	7
4	Numerical Results	10
4.1	Numerical tests for the Helmholtz problem	10
4.2	Numerical tests for the Jacobian	11
4.3	Overall model convergence	13
4.4	Accuracy vs. efficiency	15
5	Conclusions	17
A	FMG interpolation	19
B	Compact conservative Jacobians	20

1 Introduction

The multigrid method is a fast and flexible numerical technique for solving partial differential equations. This method can be applied to a given problem in conjunction with any of the known discretization techniques, such as finite differences, finite elements, and spectral methods. Furthermore, the Full Multigrid (FMG) algorithm can solve problems to the accuracy of the truncation error in a number of operations that is proportional to the number of unknowns. Multigrid processing can be combined with local mesh refinement to provide higher resolution where needed.

This paper examines the application of adaptive multigrid methods to the problem of tropical cyclone track prediction. This problem involves a small-scale vortex (the tropical cyclone) embedded in a large-scale environmental flow, with an order of magnitude difference in the typical scales of (horizontal) variation, and thus benefits greatly from local mesh refinement. The underlying dynamical model represents two-dimensional incompressible flow. It consists of two partial differential equations: a vorticity equation, expressing the conservation of absolute vorticity, and a Helmholtz problem relating the streamfunction and vorticity. A second-order accurate adaptive multigrid model (MUDBAR) based on these equations has been developed[4, 5]; the goal of the present paper is to extend this model to higher-order accuracy.

In this paper, section 2 introduces the background of the adaptive multigrid tropical cyclone track model, and details the governing equations and the time discretization. Section 3 describes the fourth-order space discretization used to solve the Helmholtz problem and vorticity equations. Numerical results are summarized in section 4 and the conclusions are given in section 5. Appendix A gives the details of the the sixth-order FMG interpolation, and Appendix B shows that no fourth-order generalization of the Arakawa Jacobian is both compact and conservative.

2 Model Description

The MUDBAR model is an adaptive multigrid tropical cyclone track model based on the nondivergent barotropic vorticity equation. In this section set the stage for our extension of this model to fourth-order accuracy by reviewing the governing equations, outlining the space discretization, and detailing the time discretization.

2.1 Governing equations

We formulate the model on a section of the sphere, transforming longitude λ and latitude ϕ to Cartesian coordinates x and y via the Mercator projection

$$x = (\lambda - \lambda_0)a \cos \phi_0, \quad y = \left[\tanh^{-1}(\sin \phi) - \tanh^{-1}(\sin \phi_0) \right] a \cos \phi_0, \quad (1)$$

where a is the radius of the earth, so the projection is true at a specified point (λ_0, ϕ_0) where $(x, y) = (0, 0)$. The governing equation is the modified barotropic vorticity equation

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

with the relative vorticity ζ and the streamfunction ψ related by

$$L\psi := \left(\nabla^2 - \frac{\gamma^2}{m^2} \right) \psi = \frac{\zeta}{m^2}. \quad (3)$$

Here $\nabla^2 = \partial^2/\partial x^2 + \partial^2/\partial y^2$, $J(\zeta, \psi)$ is the Jacobian of (ζ, ψ) with respect to (x, y) , $\beta = 2\omega \cos(\phi)/a$ (with a and ω the radius and rotation rate of the earth), $m = \cos(\phi_0)/\cos(\phi)$ is the map factor, ν is the diffusion coefficient, and γ is the inverse of the effective Rossby radius. The model domain is a rectangle Ω in x and y centered at $(x, y) = (0, 0)$. At the boundaries of Ω we specify the streamfunction ψ (and thus the normal component of the velocity); where there is inflow, we also specify the vorticity ζ .

2.2 Space discretization

To discretize (2) and (3) in space we use finite differences. To summarize the approach, first consider the discretization on a single uniform grid Ω_h of mesh size h covering the computational domain Ω . We write the space-discretized versions of (2) and (3) in the form

$$\frac{d\zeta^h}{dt} = G^h(\psi^h, \zeta^h) \quad (4)$$

and

$$L^h \psi^h = \frac{\zeta^h}{m^2}, \quad (5)$$

where the grid functions ζ^h and ψ^h represent the values of the approximate solution on a grid Ω_h of mesh size h . The details of the finite-difference approximations L^h to the Helmholtz operator L and G^h to the terms in the vorticity equation (2) will be given in section 3.

However, since a tropical cyclone is a small-scale vortex embedded in a larger-scale surrounding flow, obtaining the solution with optimum efficiency and accuracy requires using local mesh refinement. To do so we use a single “base grid” covering the entire domain and—in the region of the vortex—superimpose nested overlapping “grid patches” with successively smaller mesh sizes. On each grid the equations take the same form (4)–(5); the problem is solved on each in turn using the Berger-Oliger algorithm[2]. Since we use a multigrid method (as detailed in section 3) to solve the streamfunction equation (5) on each grid with optimum efficiency, it is possible to combine this multigrid processing with the local mesh refinement to obtain *adaptive grids*. In this approach, estimates of the truncation error computed during multigrid processing are used to decide where to refine—or coarsen—the grids. Here we take a simpler approach and use only *movable grids*, i.e., we specify the grid sizes in advance and move the grids to keep them approximately centered on the vortex as it moves. Details of both approaches—for the original second-order model—are given in [5].

2.3 Time discretization

Both the second-order and fourth-order versions of the model use the same time discretization, namely, the classical fourth-order Runge-Kutta (RK4) scheme. A single step of the RK4 scheme (from $t = t_n = n\Delta t$ to $t = t_{n+1}$) then consists of the following four stages:

Stage 1:

$$\tilde{\zeta}^{h,n+\frac{1}{2}} = \zeta^{h,n} + \frac{\Delta t}{2} G^{h,n} \quad (6)$$

Stage 2:

$$\zeta^{h,n+\frac{1}{2}} = \zeta^{h,n} + \frac{\Delta t}{2} \tilde{G}^{h,n+\frac{1}{2}} \quad (7)$$

Stage 3:

$$\tilde{\zeta}^{h,n+1} = \zeta^{h,n} + \Delta t G^{h,n+\frac{1}{2}} \quad (8)$$

Stage 4:

$$\zeta^{h,n+1} = \zeta^{h,n} + \Delta t \bar{G}^{h,n+1} \quad (9)$$

Here $\zeta^{h,n}$ and $\psi^{h,n}$ are the time-discrete approximations to ζ^h and ψ^h at $t_n = n\Delta t$, respectively, $G^{h,n} = G^h(\psi^{h,n}, \zeta^{h,n})$, $\tilde{G}^{h,n+\frac{1}{2}} = G^h(\tilde{\psi}^{h,n+\frac{1}{2}}, \tilde{\zeta}^{h,n+\frac{1}{2}})$, $G^{h,n+\frac{1}{2}} = G^h(\psi^{h,n+\frac{1}{2}}, \zeta^{h,n+\frac{1}{2}})$, and $\bar{G}^{h,n+1} = \frac{1}{6}(G^{h,n} + 2\tilde{G}^{h,n+\frac{1}{2}} + 2G^{h,n+\frac{1}{2}} + \tilde{G}^{h,n+1})$. For each stage, after getting the new value of ζ^h from the RK4 scheme, we solve (5) for the corresponding ψ^h using this ζ^h .

3 Fourth-Order Discretization

The original MUDBAR model[5] used second-order discretizations in space; to obtain fourth-order accuracy overall we must replace these with fourth-order discretizations. This section details these discretizations, using using finite differences on a uniform grid

$$\Omega_h = \{(x_i, y_j) = (x_0 + ih, y_0 + jh) : 0 \leq i \leq M, 0 \leq j \leq N\} \quad (10)$$

of mesh size h . We describe in turn the discretization of the streamfunction equation (3), a multigrid algorithm for its solution, and the discretization of the vorticity equation (2).

3.1 Streamfunction equation

The streamfunction ψ^h is computed from the vorticity ζ^h by solving a discrete version of the Helmholtz equation (3) with Dirichlet boundary conditions (i.e., ψ^h is specified at the boundary points of Ω_h). The second-order model used the usual five-point discretization of Laplacian operator, resulting in the discrete equation

$$L_2^h \psi^h = \frac{1}{h^2} \begin{bmatrix} & & 1 & & \\ & 1 & -4 & -\frac{\gamma^2}{m^2} & 1 \\ & & & & \\ & & & & \\ & & & & \end{bmatrix} \psi^h = \frac{\zeta^h}{m^2} = F^h. \quad (11)$$

Here the stencil notation gives the contributions from the point at which the equation is applied and from its neighbors. Gauss-Seidel (GS) relaxation for this discretization updates the values of an approximate solution $\tilde{\psi}^h$ via

$$\tilde{\psi}_{i,j} \leftarrow \left(4 + \frac{\gamma^2}{m^2}\right)^{-1} \left[(\tilde{\psi}_{i-1,j} + \tilde{\psi}_{i+1,j} + \tilde{\psi}_{i,j-1} + \tilde{\psi}_{i,j+1}) - \frac{h^2}{m^2} \zeta_{i,j} \right], \quad (12)$$

where $\tilde{\psi}_{i,j}$ denotes the current approximation to $\psi_{i,j} = \psi^h(x_i, y_j)$ and $\zeta_{i,j} = \zeta^h(x_i, y_j)$. The corresponding residual is given by

$$r_{i,j} = \frac{\zeta_{i,j}}{m^2} - \frac{[\tilde{\psi}_{i-1,j} + \tilde{\psi}_{i+1,j} + \tilde{\psi}_{i,j-1} + \tilde{\psi}_{i,j+1} - (4 + \frac{\gamma^2}{m^2}) \tilde{\psi}_{i,j}]}{h^2}. \quad (13)$$

To obtain fourth-order accuracy we must replace (11)–(13) with fourth-order analogues. According to the conclusion of Schaffer [6], the Mehrstellen Verfahren (MV) discretization was the most accurate among the fourth-order discretizations he studied. Thus, the fourth-order MV difference stencil was applied here, giving the discrete problem

$$L_M^h \psi^h = \frac{1}{6h^2} \begin{bmatrix} 1 & & 4 & & 1 \\ 4 & -20 & -\frac{\gamma^2}{m^2} & 4 & \\ 1 & & 4 & & 1 \end{bmatrix} \psi^h = \frac{1}{12m^2} \begin{bmatrix} 1 & 1 \\ 1 & 8 & 1 \\ 1 & 1 \end{bmatrix} \zeta^h = F^h. \quad (14)$$

Since this discretization is compact, i.e., it involves only the nearest-neighboring points, it can be applied at all interior points on the grid. Gauss-Seidel relaxation for (14) gives the iteration

$$\begin{aligned} \psi_{i,j} \leftarrow & - \left(20 + \frac{\gamma^2}{m^2} \right)^{-1} \left[\frac{h^2}{2m^2} (8\zeta_{i,j} + \zeta_{i+1,j} + \zeta_{i-1,j} + \zeta_{i,j-1} + \zeta_{i,j+1}) \right. \\ & - \left(\tilde{\psi}_{i-1,j-1} + \tilde{\psi}_{i-1,j+1} + \tilde{\psi}_{i+1,j-1} + \tilde{\psi}_{i+1,j+1} \right) \\ & \left. - 4 \left(\tilde{\psi}_{i-1,j} + \tilde{\psi}_{i+1,j} + \tilde{\psi}_{i,j-1} + \tilde{\psi}_{i,j+1} \right) \right]. \end{aligned} \quad (15)$$

The corresponding residual is given by

$$\begin{aligned} r_{i,j} = & \frac{(\zeta_{i+1,j} + \zeta_{i-1,j} + \zeta_{i,j-1} + \zeta_{i,j+1} + 8\zeta_{i,j})}{12m^2} \\ & - \frac{\tilde{\psi}_{i-1,j-1} + \tilde{\psi}_{i-1,j+1} + \tilde{\psi}_{i+1,j-1} + \tilde{\psi}_{i+1,j+1} - \left(20 + \frac{\gamma^2}{m^2} \right) \tilde{\psi}_{i,j}}{6h^2} \\ & - \frac{4 \left(\tilde{\psi}_{i-1,j} + \tilde{\psi}_{i+1,j} + \tilde{\psi}_{i,j-1} + \tilde{\psi}_{i,j+1} \right)}{6h^2}, \end{aligned} \quad (16)$$

where $\tilde{\psi}^h$ is the current approximate solution.

In the second-order model, red-black ordering was used in the finite difference discretization. This ordering is often used in multigrid methods since it gives better smoothing factors. We also use the red-black ordering in the higher-order model.

3.2 Multigrid algorithm

Multigrid methods solve discretized elliptic problems such as (11) or (14) by combining relaxation (to reduce errors on the scale of the grid) with corrections computed from coarser grids (to eliminate the smoother error components). An introduction to multigrid methods can be found in [3] and [7]; here we simply give the details of the components of the multigrid algorithm used here. Our focus here is on choosing these components so the resulting solution is obtained *to the level of truncation error*. That is, we want $\|r^h\| \leq \|\tau^h\|$ in an appropriate norm, where $r^h = F^h - L^h \tilde{\psi}^h$ is the residual and $\tau^h = L^h \psi - F^h$ is the truncation error. When this happens, $\|\psi^h - \tilde{\psi}^h\|$ is sufficiently small so that $\|\psi - \tilde{\psi}^h\| \approx \|\psi - \psi^h\|$ and further computational work would be wasted.

3.2.1 Relaxation sweeps

The effectiveness of relaxation as a smoother is measured by the multigrid smoothing factor $\bar{\mu}$. This number is defined as the maximum convergence factor for high-wavenumber errors for

one relaxation sweep; it is computed by local mode analysis[3]. Schaffer[6] gave the smoothing factor $\bar{\mu} = 0.464$ for the fourth-order MV discretization with lexicographic ordering; our numerical results are slightly better, giving an effective convergence factor per sweep of about 0.38. The optimum number of relaxation sweeps per level in a multigrid V-cycle depends on $\bar{\mu}$; for example, Taft[8] used three relaxation sweeps per level in a multigrid algorithm using the MV discretization with lexicographic ordering.

For the second-order discretization (11) it can be shown that red-black ordering (as in a checkerboard) gives significant improvement over lexicographic ordering ($\bar{\mu} = 0.25$ instead of $\bar{\mu} = 0.5$). This result rests on a complicated interaction between Fourier modes made possible by the five-point nature of the operator in (11). Since this five-point structure is lost in going to the MV discretization (14), it might be supposed that there would be no advantage here to using red-black ordering. However, our numerical results show an effective convergence factor per sweep of about 0.32 in this case. This is small enough that two relaxation sweeps per level suffice. Thus, our multigrid algorithm uses V(1,1) cycles, i.e., V-cycles with one sweep each before and after the coarse-grid correction.

3.2.2 Control algorithm

To control the multigrid processing we use the Full Multigrid (FMG) algorithm. This proceeds from the coarsest to the finest grid, solving on each grid with one or more multigrid V-cycles and then interpolating that solution to the next finer grid to serve as the initial approximation there. Properly designed, this algorithm solves to the level of truncation error, independent of the mesh size h . The key here is that the work done by the V-cycle(s) on each level must reduce the residual by at least the same factor that the truncation error is reduced by going to that level; this guarantees that the problem is solved to the level of truncation error on all levels. For our fourth-order algorithm with a mesh refinement ratio of two, that factor is $2^4 = 16$. It would appear that one V(2,1) cycle would suffice, since the corresponding residual reduction is $(0.32)^3 \approx 0.03 < 0.0625 = 1/16$; however, this neglects the additional error introduced by grid transfers. In practice, two V(1,1) cycles per level suffice; the corresponding residual reduction is about $(0.32)^4 \approx 0.01$, which is less than $1/16 = 0.0625$. Thus, we use the 2-FMV(1,1) algorithm, i.e., using two V(1,1) cycles per level in the FMG algorithm.

Within the algorithm, the fourth-order MV discretization need only be used on the “currently finest level”, i.e., the finest level yet reached[6, 11]. Thus, the problem to be solved on that level is (14), it is smoothed by the corresponding relaxation (15), and the corresponding residual (16) is transferred to the coarser grid. However, on that coarser grid (and any still coarser), the second-order discretization (11)–(13) suffices (it is actually better and faster). No special grid transfers are needed within the V-cycle: the residuals can be transferred using normal full weighting (see also section 4.4) and the corrections can be transferred using bilinear interpolation. However, to achieve best performance overall, using higher accuracy in the initial (FMG) interpolation to the next finer level is advantageous[9]. Here we use sixth-order interpolation as detailed in Appendix A.

3.2.3 Summary

To summarize, the fourth-order multigrid algorithm for the Helmholtz problem consists of the following components:

- Full multigrid (FMG) algorithm,
- Sixth-order initial FMG interpolation,
- Two V(1,1) cycles per level,
- Fourth-order (MV) discretization on the currently finest level and second-order discretization on coarser levels,
- Gauss-Seidel relaxation with red-black ordering,
- Full weighting for the transfer of residuals,
- Bilinear interpolation for the transfer of corrections.

Numerical results (see section 4) show that this multigrid algorithm solves to the level of truncation error (i.e., residual norm less than the truncation error norm) for the problems with known (smooth) solutions, independent of the mesh size h . Almost any change (e.g., fourth-order initial FMG interpolation) destroys this property.

3.3 Vorticity equation

It might be hoped that obtaining fourth-order accuracy in the streamfunction equation (5) as described here might be enough to increase the overall accuracy of the model. Unfortunately, Burgess and Fulton[11] observed that this is not the case: we must seek a higher-order discretization of the remaining equation for vorticity. Thus, we turn next to the discretization of the spatial terms in the vorticity equation (2) to produce the term G^h in (4). We focus on how to approximate the Jacobian

$$J(\zeta, \psi) = \frac{\partial \zeta}{\partial x} \frac{\partial \psi}{\partial y} - \frac{\partial \psi}{\partial x} \frac{\partial \zeta}{\partial y} \quad (17)$$

with fourth-order accuracy, since the approximation of the remaining terms by fourth-order finite differences (uncentered where needed) is straightforward.

The second-order version of the model uses the Jacobian of Arakawa[1], which can be written as

$$J_1(\zeta, \psi) = \frac{1}{3} \left[J^{++}(\zeta, \psi) + J^{+\times}(\zeta, \psi) + J^{\times+}(\zeta, \psi) \right]. \quad (18)$$

Here

$$\begin{aligned}
J_{i,j}^{++}(\zeta, \psi) &= \frac{1}{4h^2} [(\zeta_{i+1,j} - \zeta_{i-1,j}) (\psi_{i,j+1} - \psi_{i,j-1}) \\
&\quad - (\zeta_{i,j+1} - \zeta_{i,j-1}) (\psi_{i+1,j} - \psi_{i-1,j})], \tag{19}
\end{aligned}$$

$$\begin{aligned}
J_{i,j}^{+\times}(\zeta, \psi) &= \frac{1}{4h^2} [\zeta_{i+1,j} (\psi_{i+1,j+1} - \psi_{i+1,j-1}) - \zeta_{i-1,j} (\psi_{i-1,j+1} - \psi_{i-1,j-1}) \\
&\quad - \zeta_{i,j+1} (\psi_{i+1,j+1} - \psi_{i-1,j+1}) + \zeta_{i,j-1} (\psi_{i+1,j-1} - \psi_{i-1,j-1})], \tag{20}
\end{aligned}$$

and

$$\begin{aligned}
J_{i,j}^{\times+}(\zeta, \psi) &= \frac{1}{4h^2} [\zeta_{i+1,j+1} (\psi_{i,j+1} - \psi_{i+1,j}) - \zeta_{i-1,j-1} (\psi_{i-1,j} - \psi_{i,j-1}) \\
&\quad - \zeta_{i-1,j+1} (\psi_{i,j+1} - \psi_{i-1,j}) + \zeta_{i+1,j-1} (\psi_{i+1,j} - \psi_{i,j-1})], \tag{21}
\end{aligned}$$

This second-order discretization of Jacobian is both *compact* (involving only nearest-neighbor points) and *conservative* (exactly conserving discrete analogues of vorticity, enstrophy, and kinetic energy). It is natural to ask whether a fourth-order analogue with these same two properties exists; unfortunately, the answer is no, as shown in Appendix B.

A fourth-order discrete Jacobian which maintains conservation by sacrificing compactness was given by Arakawa[1]. This can be written in the form

$$J_4(\zeta, \psi) := 2J_1(\zeta, \psi) - J_2(\zeta, \psi) = J(\zeta, \psi) + O(h^4). \tag{22}$$

Here $J_1(\zeta, \psi)$ is the usual second-order Arakawa's Jacobian given by (18) and

$$J_2(\zeta, \psi) = \frac{[J_{i,j}^{\times\times}(\zeta, \psi) + J_{i,j}^{\times*}(\zeta, \psi) + J_{i,j}^{*\times}(\zeta, \psi)]}{3} \tag{23}$$

is second-order non-compact conservative Jacobian, where

$$\begin{aligned}
J_{i,j}^{\times\times}(\zeta, \psi) &= \frac{1}{8h^2} [(\zeta_{i+1,j+1} - \zeta_{i-1,j-1}) (\psi_{i-1,j+1} - \psi_{i+1,j-1}) \\
&\quad - (\zeta_{i-1,j+1} - \zeta_{i+1,j-1}) (\psi_{i+1,j+1} - \psi_{i-1,j-1})], \tag{24}
\end{aligned}$$

$$\begin{aligned}
J_{i,j}^{\times*}(\zeta, \psi) &= \frac{1}{8h^2} [\zeta_{i+1,j+1} (\psi_{i,j+2} - \psi_{i+2,j}) - \zeta_{i-1,j-1} (\psi_{i-2,j} - \psi_{i,j-2}) \\
&\quad - \zeta_{i-1,j+1} (\psi_{i,j+2} - \psi_{i-2,j}) + \zeta_{i+1,j-1} (\psi_{i+2,j} - \psi_{i,j-2})], \tag{25}
\end{aligned}$$

and

$$\begin{aligned}
J_{i,j}^{*\times}(\zeta, \psi) &= \frac{1}{8h^2} [\zeta_{i+2,j} (\psi_{i+1,j+1} - \psi_{i+1,j-1}) - \zeta_{i-2,j} (\psi_{i-1,j+1} - \psi_{i-1,j-1}) \\
&\quad - \zeta_{i,j+2} (\psi_{i+1,j+1} - \psi_{i-1,j+1}) + \zeta_{i,j-2} (\psi_{i+1,j-1} - \psi_{i-1,j-1})]. \quad (26)
\end{aligned}$$

This fourth-order discretization of the Jacobian also conserves the square of vorticity, enstrophy and kinetic energy. However, this discretization not only uses the nearest-neighbor grid points, but also involves the additional grid points $(i+2, j)$, $(i-2, j)$, $(i, j+2)$, $(i, j-2)$. Thus, the Arakawa fourth-order Jacobian can be applied to all interior points in the grid *except* those along the first interior lines (adjacent to each boundary). At those points, we use the second-order discretization (18); at the boundary points, we use the first-order discretization described by Fulton [4]. Since the set of points where lower accuracy is used is small, this has negligible effect on the overall accuracy.

4 Numerical Results

This section describes results of numerical tests verifying the fourth-order discretization of the Helmholtz problem and the Jacobian, as well as numerical results showing the convergence and efficiency of the higher-order model as a whole.

4.1 Numerical tests for the Helmholtz problem

To test the fourth-order multigrid algorithm for the Helmholtz problem, we specified ψ as

$$\psi(x, y) = \psi_0 \cos\left(\frac{\pi x}{L_x}\right) \cos\left(\frac{\pi y}{L_y}\right), \quad (27)$$

where ψ_0 is constant, and L_x and L_y are x -scale and y -scale of the environmental flow, respectively. With $m = 1$ and $\gamma = 0$ in the Helmholtz problem (5), the corresponding vorticity is

$$\zeta(x, y) = -\pi^2 \left(\frac{1}{L_x^2} + \frac{1}{L_y^2}\right) \psi_0 \cos\left(\frac{\pi x}{L_x}\right) \cos\left(\frac{\pi y}{L_y}\right). \quad (28)$$

Using the values $L_x = 2048$ km and $L_y = 1638.4$ km, we ran the model and computed the residual, truncation error, and solution error for different values of the mesh size h . We measured these quantities using the l_2 norm

$$\|f\|_2 = \sqrt{h^2 \sum_i \sum_j f_{i,j}^2}, \quad (29)$$

where $f_{i,j} = f(x_i, y_j)$.

Table 1 lists the norms of the truncation error $\tau^h = L^h\psi - F^h$, residual r^h , and solution error $\psi - \tilde{\psi}^h$ for various values of the mesh size h . Also listed is the ratio $\|\tau^{2h}\|/\|\tau^h\|$ of

Table 1: Fourth-order Helmholtz solver: residuals and errors

h (km)	$\ \tau^h\ $	ratio	$\ r^h\ $	$\ \psi - \tilde{\psi}^h\ $	$\ \psi - \psi^h\ $	ratio
128	2.888(-4)	–	1.113(-4)	3.990(7)	3.499(7)	–
64	1.869(-5)	15.45	7.025(-6)	2.604(6)	2.196(6)	15.32
32	1.185(-6)	15.77	4.040(-7)	1.654(5)	1.374(5)	15.74
16	7.453(-8)	15.90	2.417(-8)	1.039(4)	8.591(3)	15.92
8	4.636(-9)	16.08	1.484(-9)	6.457(2)	5.327(2)	16.09

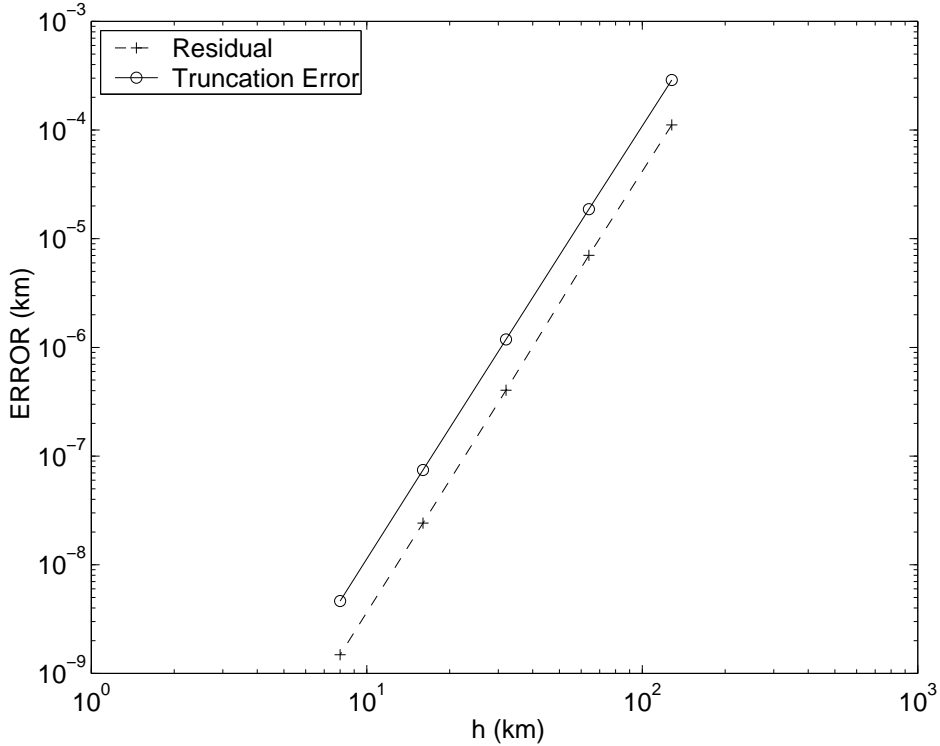


Figure 1: Fourth-order Helmholtz solver: residuals and errors

truncation errors and the corresponding ratio of solution errors. These ratios show that both the discretization and the resulting solution have accuracy $O(h^4)$. The table also shows that the problem was solved to the level of truncation error ($\|r^h\| \leq \|\tau^h\|$), and that $\|\psi - \tilde{\psi}^h\| \approx \|\psi - \psi^h\|$. Figure 1 shows the same conclusion graphically: the norm of the residual less than the norm of the truncation error. We tried many different cases, i.e., other values of L_x and L_y and other specified solutions ψ (results not shown). Almost every case is solved to the truncation error; in the few cases which are not, the approximate solution is still very close the solution.

4.2 Numerical tests for the Jacobian

To verify the accuracy of the Jacobians (18) and (22) as coded in the model, we specified ζ and ψ as

$$\psi(x, y) = -u_0 y + \frac{v_1}{k} \sin(kx) - \frac{u_1}{l} \sin(lx), \quad (30)$$

and

$$\zeta(x, y) = -kv_1 \sin(kx) + lu_1 \sin(lx), \quad (31)$$

and substituted ψ and ζ into the second-order and fourth-order Jacobians for different mesh sizes h . For these tests we used the domain $[0, 2] \times [0, 2]$ and the values $k = 3.8\pi$, $l = \pi/2$,

$u_0 = 2$, $u_1 = 8$, and $v_1 = 3$. To verify the order of Jacobian we computed the effective order $P := \log(\|J - J^{2h}\|/\|J - J^h\|) / \log 2$, where J is the continuous Jacobian, J^h is the discrete Jacobian on the grid h , measuring the results in both the l_∞ and l_2 norms.

In Tables 2 and 3, we give the results in the l_∞ norm. The results in the l_2 norm are shown in Table 4. The tables include three kinds of results: P_1 means the order of truncation error computed over all points, P_2 omits the boundary points, and P_3 omits the boundary and first interior points. Recall that at the boundary points we use the same Jacobian as in [4], which is only first-order. Thus, the values of P_1 in Tables 2 and 3 are about 1. Similarly, since the fourth-order Jacobian (22) is noncompact, it cannot be applied at the first interior lines, so we must use the second-order Jacobian (18) there. Thus, the values of P_2 in Tables 2 and 3 are about 2. The values of P_3 in Table 2 is also about 2, and P_3 for the fourth-order Jacobian in Table 3 is about 4. These l_∞ norm results show that the Jacobians have the expected order pointwise.

Similarly, Table 4 gives the values of P_3 for the second-order and fourth-order Jacobians in the l_2 norm. Since these results show overall second- and fourth-order convergence, respectively, we conclude that the effects of lower accuracy on and adjacent to the boundaries are small.

Table 2: Effective order for second-order Jacobian (l_∞ norm)

h	P_1	P_2	P_3
$\frac{1}{16}$	–	–	–
$\frac{1}{32}$	1.3441	2.1727	2.3088
$\frac{1}{64}$	0.7557	2.1140	2.2091
$\frac{1}{128}$	0.8976	2.0634	2.1209
$\frac{1}{256}$	0.9529	2.0337	2.0655

Table 3: Effective order for fourth-Order Jacobian (l_∞ norm)

h	P_1	P_2	P_3
$\frac{1}{16}$	–	–	–
$\frac{1}{32}$	1.3441	2.1727	4.2666
$\frac{1}{64}$	0.7557	2.1140	4.1983
$\frac{1}{128}$	0.8976	2.0634	4.1180
$\frac{1}{256}$	0.9529	2.0337	4.0647

Table 4: Effective order P_3 for second- and fourth-order Jacobians (l_2 norm)

h	Second-order P_3	Fourth-order P_3
$\frac{1}{16}$	–	–
$\frac{1}{32}$	1.8544	3.8128
$\frac{1}{64}$	1.9150	3.9046
$\frac{1}{128}$	1.9554	3.9528
$\frac{1}{256}$	1.9775	3.9768

4.3 Overall model convergence

Having verified that all components of the model are indeed fourth-order accurate, we now run the whole model to verify that it is fourth-order accurate overall. Following DeMaria[10], we initialize the model with a hurricane-like vortex embedded in a larger-scale environmental flow. The vortex is defined by

$$V(r) = 2V_m \left(\frac{r}{r_m} \right) \frac{\exp[-a(r/r_m)^b]}{1 + (r/r_m)^2}, \quad (32)$$

where V is the tangential wind at the radial distance r from the vortex center (x_0, y_0) and V_m is the approximate maximum value of V near $r = r_m$ (exact when $a = 0$). The exponential factor in (32) ensures that V decreases quickly as r becomes large. The environmental flow is a zonal current with streamfunction defined by

$$\bar{\psi}(y) = \left(\frac{\bar{u}_0 L}{2\pi} \right) \cos \left(\frac{2\pi y}{L} \right). \quad (33)$$

The computational domain is a square of side length 4096 km on a β -plane centered at latitude 20° N, with the vortex centered initially at $x_0 = 768$ km and $y_0 = 768$ km. We use the values $V_m = 30 \text{ ms}^{-1}$, $r_m = 80$ km, $a = 10^{-6}$, $b = 6$, $\bar{u}_0 = 10 \text{ ms}^{-1}$, $L = 4096$ km, and $\gamma = 0$.

To quantify the model performance we use the forecast (track) error, defined as the difference in vortex position compared to that computed in a high-resolution reference run ($h = 4$ km). We run the fourth-order model with $h = 16, 32, 42.666, 51.20, 64, 128$ km, and run the second-order model with $h = 16, 32, 64, 128$ km. Figure 2 shows the forecast error as a function of h for these uniform-grid runs of the second-order and fourth-order models at 24, 48, and 72 hours and the mean errors over each 72 hour model run. The slopes of the curves verify that the overall model converges at approximately the proper rate.

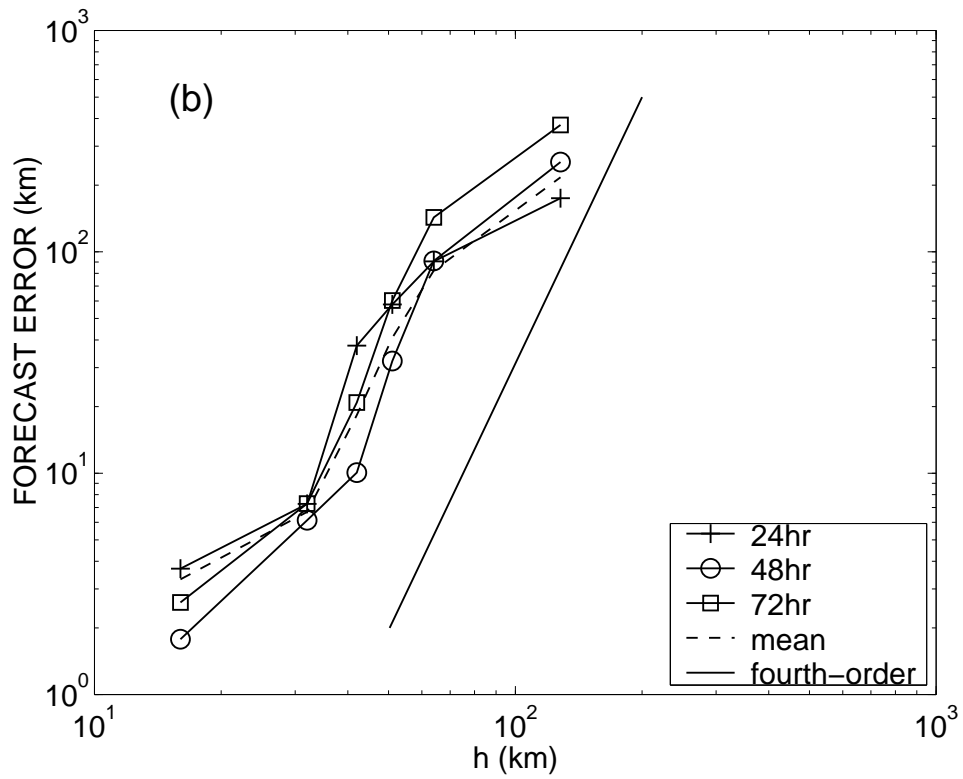
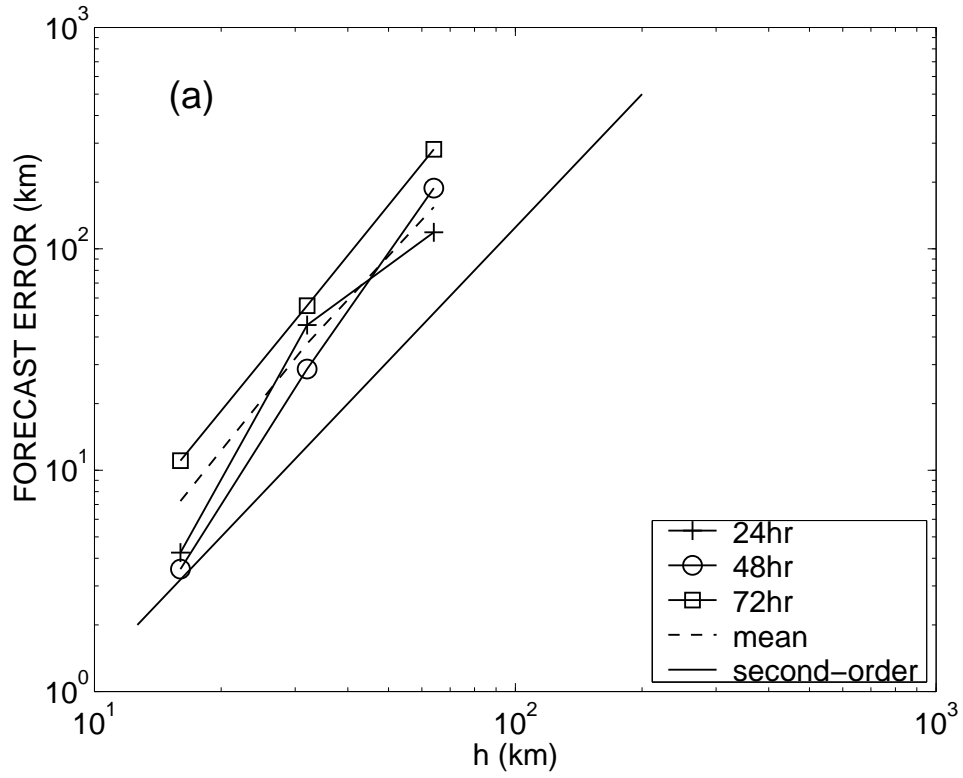


Figure 2: Overall convergence of the model:
 (a) second-order version, (b) fourth-order version.

4.4 Accuracy vs. efficiency

Asymptotically, fourth-order differencing is superior to second-order differencing: for mesh sizes h smaller than some critical value, the fourth-order solution will be more accurate. However, this does not account for the computational work involved in computing the higher-order approximation. Taking this into account, does the asymptotic superiority of fourth-order differencing pay in practice? To answer this, we ran both models with a variety of grids (uniform grids and local refinements using patches of various specified sizes). Figure 3 shows the resulting mean forecast errors as a function of the computer time (for a 72 hour model run). For the uniform-grid runs (points joined by lines in Figure 3) the fourth-order model is approximately 5–10 times faster than the second-order model (for the same level of error); for the same amount of computer time, it is at least 5 times more accurate. With local refinement (remaining points in Figure 3), fourth-order differencing shows some advantage for moderate resolution (e.g., errors of 10–30 km).

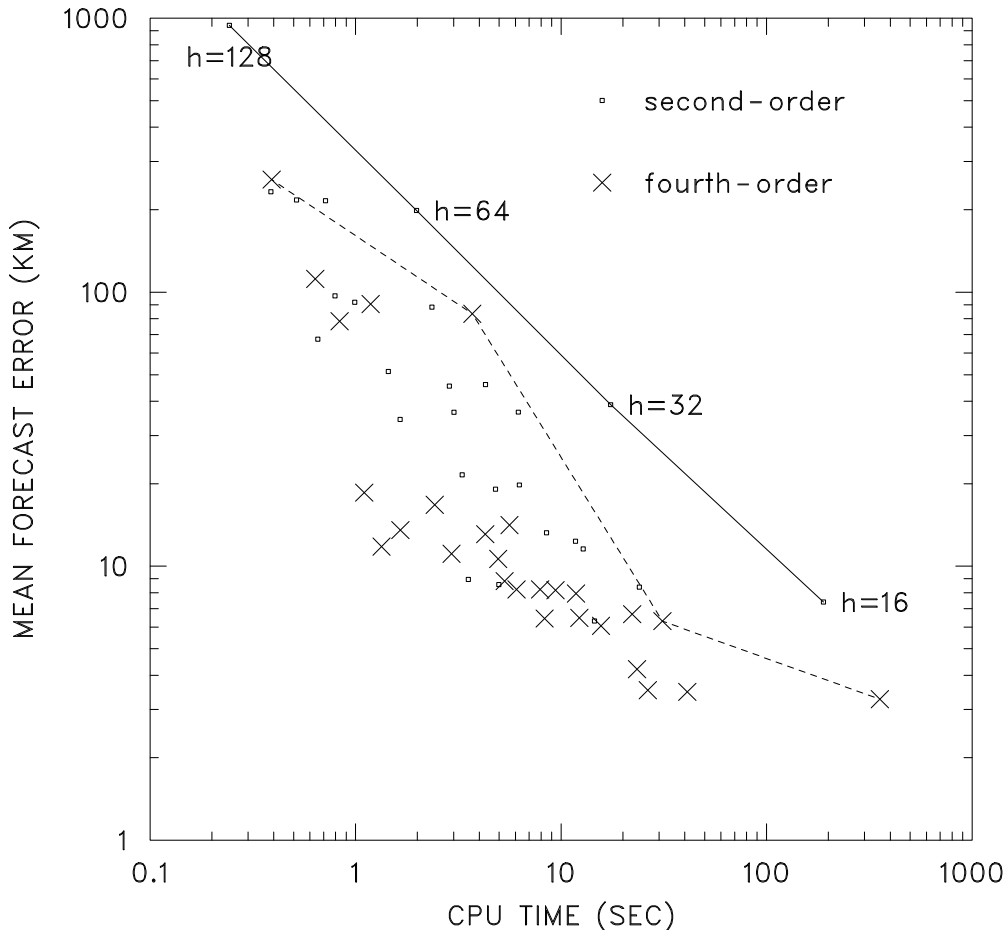


Figure 3: Forecast error vs. CPU time for second-order and fourth-order models.

In an attempt to improve the model performance, we considered the choice of operators used to transfer the fine-grid vorticity to the coarse grid at end of each time step with nested grid patches. Normally (e.g., Fig. 3) we use standard full weighting (as used for the residual transfer). Since this is only second-order accurate, we consider the results of two other methods: (1) injection and (2) fourth-order full weighting, defined by the stencil

$$I_h^{2h} = \frac{1}{1024} \begin{bmatrix} 1 & 0 & -9 & -16 & -9 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -9 & 0 & 81 & 144 & 81 & 0 & -9 \\ -16 & 0 & 144 & 256 & 144 & 0 & -16 \\ -9 & 0 & 81 & 144 & 81 & 0 & -9 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -9 & -16 & -9 & 0 & 1 \end{bmatrix}. \quad (34)$$

Figure 4 is the results with fourth-order full weighting and Fig. 5 is the results with injection. Compared to Fig. 3, we see that neither method improves the results significantly.

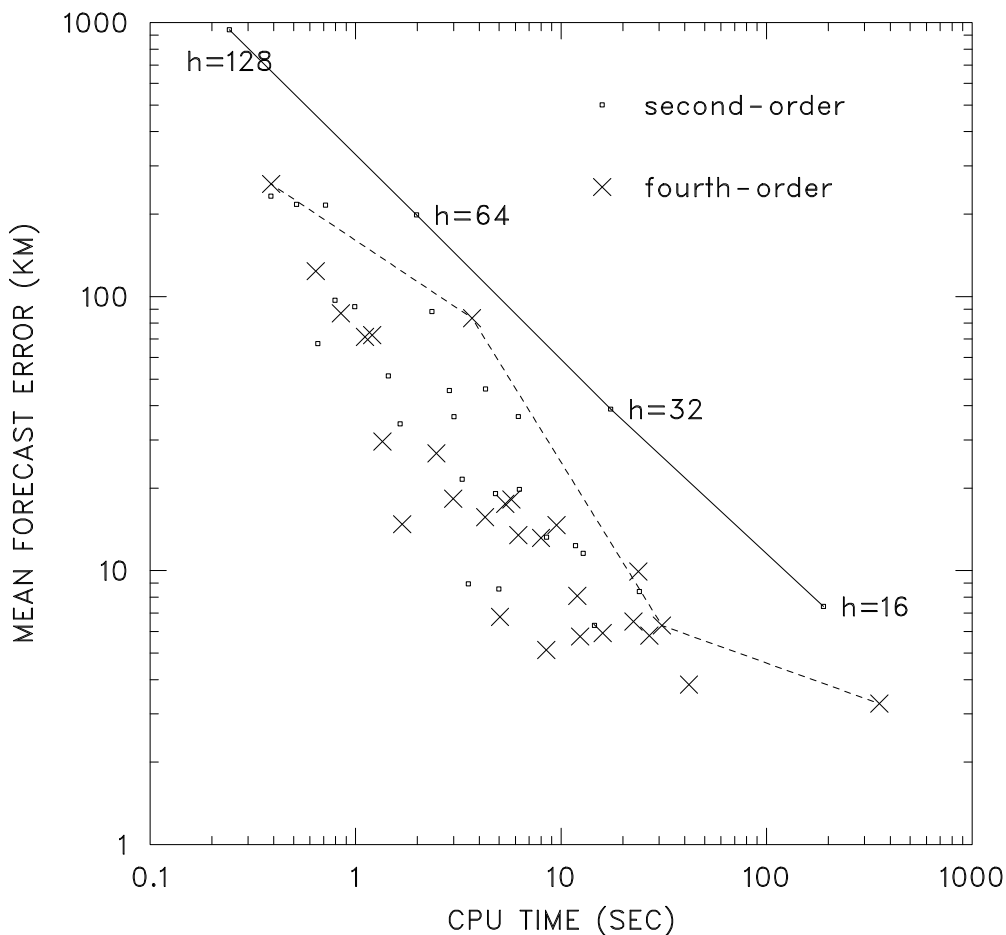


Figure 4: Forecast error vs. CPU time with fourth-Order full weighting.

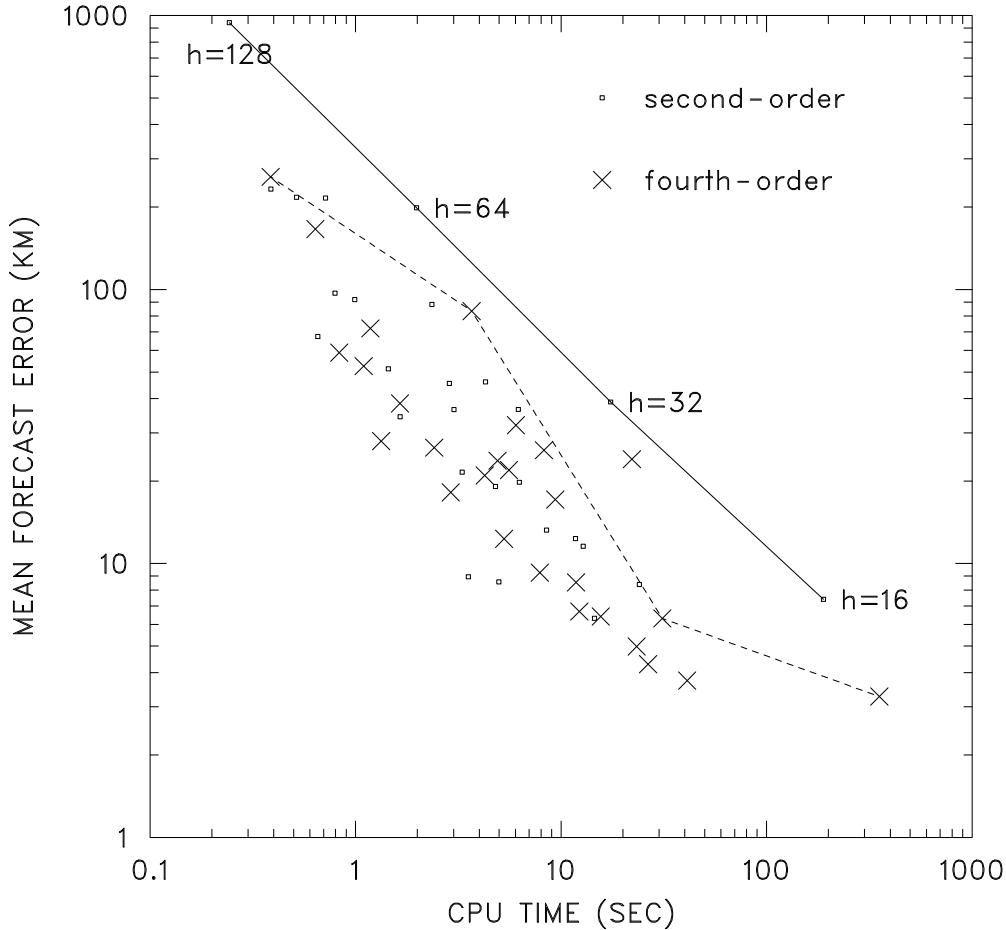


Figure 5: Forecast error vs. CPU time with injection.

5 Conclusions

We have developed a fourth-order version of the MUDBAR model. The Helmholtz problem for the streamfunction is solved by an FMG multigrid method, using fourth-order compact discretization on the finest level in each V-cycle, sixth-order initial interpolation, Gauss-Seidel relaxation with red-black ordering, and two V(1,1) cycles per level. Numerical results showed that this algorithm solves the Helmholtz problem to the level of truncation error. For the vorticity equation, we applied the fourth-order conservative Jacobian of Arakawa; this is not compact, but no compact version exists as shown in Appendix B. Numerical results verified that the whole model is fourth-order accurate. Compared to the second-order model[5], the fourth-order model provides higher accuracy (or efficiency) when using a single uniform grid; when used with local mesh refinement, fourth-order accuracy provides less benefit.

References

- [1] Arakawa, A., 1966: Computational design for long-term numerical integration of the equations of fluid motion: Two-dimensional incompressible flow. Part I. *J. Comput. Phys.*, **1**, 119–143.
- [2] Berger, M. J. and J. Oliger, 1984: Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comput. Phys.*, **53**, 484–512.
- [3] Brandt, A., 1977: Multi-level adaptive solutions to boundary-value problems. *Math. Comput.*, **31**, 333–390.
- [4] Fulton, S. R., 1997: A comparison of multilevel adaptive methods for hurricane track prediction. *Elec. Trans. Num. Anal.*, **6**, 120-132.
- [5] Fulton, S. R., 2000: An adaptive multigrid barotropic tropical cyclone track model. *Mon. Wea. Rev.*, in press.
- [6] Schaffer, S., 1984: Higher-Order Multi-Grid Methods. *Math. Comput.*, **43**, 89–115.
- [7] Fulton, S. R. , P. E. Ciesielski, and W. H. Schubert, 1986: Multigrid Methods for Elliptic Problems: A Review. *Monthly Weather Review*, **114**, 943–959.
- [8] Taft, R., 1987: A comparison of two multigrid methods for solving the Poisson Problem. *Atmos. Sci.*, Paper No. 415, Dept. of Atmospheric Science, Colorado State University.
- [9] Brandt, A., 1984: *Multigrid Techniques: 1984 Guide with applications to fluid dynamics*, 176 pp.
- [10] DeMaria, M., 1985: Tropical cyclone track prediction with a barotropic model. *Mon. Wea. Rev.*, **113**, 1199–1210.
- [11] Burgess, N. M., and S. R. Fulton, 1999: Experiments with Higher Order Multigrid Methods. *Technical Report No. 99-02, Department of Mathematics and Computer Science*, Clarkson University.

A FMG interpolation

This Appendix details the sixth-order initial interpolation used in the FMG algorithm. For the fine-grid points which coincide with coarse-grid points, the values are simply copied from the coarse grid to fine grid. For the remaining fine-grid points, the values are obtained by one-dimensional interpolations in x and y , using sixth-order interpolation where possible as follows.

If the coarse grid contains at least six points in a given direction, we use sixth-order interpolation. At the first interior grid points this takes the form

$$\psi^h \approx \frac{1}{256}[63 \quad \bullet \quad 315 \quad -210 \quad 126 \quad -45 \quad 7]\psi^{2h} \quad (35)$$

or

$$\psi^h \approx \frac{1}{256}[7 \quad -45 \quad 126 \quad -210 \quad 315 \quad \bullet \quad 63]\psi^{2h} , \quad (36)$$

where ψ^h and ψ^{2h} are the approximate solutions on the fine and coarse grid, respectively. Here, the numbers in the stencils denote the weights applied at the coarse-grid points and the bullet \bullet shows the location of the resulting fine-grid value. Likewise, at the third interior grid points we use

$$\psi^h \approx \frac{1}{256}[-7 \quad 105 \quad \bullet \quad 210 \quad -70 \quad 21 \quad -3]\psi^{2h} \quad (37)$$

or

$$\psi^h \approx \frac{1}{256}[-3 \quad 21 \quad -70 \quad 210 \quad \bullet \quad 105 \quad -7]\psi^{2h} , \quad (38)$$

and at the remaining interior grid points we use

$$\psi^h \approx \frac{1}{256}[3 \quad -25 \quad 150 \quad \bullet \quad 150 \quad -25 \quad 3]\psi^{2h} . \quad (39)$$

If the coarse grid contains four or five points in a given direction, we use cubic interpolation: at the first interior grid points we use

$$\psi^h \approx \frac{1}{16}[5 \quad \bullet \quad 15 \quad -5 \quad 1]\psi^{2h} \quad (40)$$

or

$$\psi^h \approx \frac{1}{16}[1 \quad -5 \quad 15 \quad \bullet \quad 5]\psi^{2h} , \quad (41)$$

and at the remaining interior grid points we use

$$\psi^h \approx \frac{1}{16}[-1 \quad 9 \quad \bullet \quad 9 \quad -1]\psi^{2h} . \quad (42)$$

Finally, if the coarse grid contains three points in a given direction, we use the quadratic interpolation

$$\psi^h \approx \frac{1}{8}[3 \quad \bullet \quad 6 \quad -1]\psi^{2h} \quad (43)$$

or

$$\psi^h \approx \frac{1}{8}[-1 \quad 6 \quad \bullet \quad 3]\psi^{2h} \quad (44)$$

B Compact conservative Jacobians

This Appendix reviews and extends the arguments of Arakawa[1] to find all discretizations of the Jacobian which are both *compact* (involving only nearest-neighbor points) and *conservative* (exactly conserving discrete analogues of vorticity, enstrophy, and kinetic energy). We show that these discrete Jacobians form a one-parameter family; however, all are only second-order accurate, and the classical Arakawa Jacobian is the optimum choice.

Following [1], we can write a relatively general discrete form of the Jacobian as

$$J_{i,j}(\zeta, \psi) = \sum_{i',j'} \sum_{i'',j''} c_{i',j';i'',j''} \zeta_{i+i',j+j'} \psi_{i+i'',j+j''}. \quad (45)$$

Here, i and j index the point (x_i, y_j) on a uniform grid Ω_h of mesh size h where the Jacobian is to be computed, $\zeta_{i+i',j+j'}$ is the vorticity at a neighboring grid point $(x_{i+i'}, y_{j+j'})$, $\psi_{i+i'',j+j''}$ is the streamfunction at a neighboring grid point $(x_{i+i''}, y_{j+j''})$, and the sums extend over whatever points near (i, j) which we choose to use. Since we assume a uniform Cartesian grid here, the coefficients $c_{i',j';i'',j''}$ may be assumed independent of the point (i, j) at which (45) is applied. Our goal is to choose these coefficients so that the resulting discretization has the desired properties.

One essential property is *consistency*: comparing the continuous Jacobian J to its discretization $J_{i,j}$, we need $|J_{i,j}(\zeta, \psi) - J(\zeta, \psi)(x_i, y_j)| \rightarrow 0$ as $h \rightarrow 0$ for all suitably smooth functions ζ and ψ . Since $J(\zeta, \psi) = 0$ when either ζ or ψ is constant, we will require $J_{i,j}(1, \psi) = 0$ and $J_{i,j}(\zeta, 1) = 0$ for all suitably smooth functions ζ and ψ . In view of (45), these conditions lead to the constraints

$$\sum_{i',j'} c_{i',j';i'',j''} = 0 \quad (\text{each } i'', j'') \quad (46)$$

and

$$\sum_{i'',j''} c_{i',j';i'',j''} = 0 \quad (\text{each } i', j'). \quad (47)$$

Additionally, since $J(x, y) = 1$, we will require $J_{i,j}(x, y) = 1$. Using (45)–(47) and the substitutions $x_i = x_0 + ih$ and $y_j = y_0 + jh$, this condition leads to the constraint

$$h^2 \sum_{i',j'} \sum_{i'',j''} i' j'' c_{i',j';i'',j''} = 1. \quad (48)$$

Arakawa[1] showed that in addition to consistency, the discrete Jacobian must mimic the conservation properties of the continuous Jacobian for the resulting solutions to be of much value. The continuous Jacobian satisfies

$$\int_{\Omega} J(\zeta, \psi) dx dy = 0, \quad (49)$$

$$\int_{\Omega} \zeta J(\zeta, \psi) dx dy = 0, \quad (50)$$

and

$$\int_{\Omega} \psi J(\zeta, \psi) dx dy = 0 \quad (51)$$

when ψ is constant on the boundary of the domain Ω ; from (2) we see these correspond (at least when $m = 1$, $\beta = 0$, and $\nu = 0$) to conservation of vorticity, enstrophy (square vorticity), and kinetic energy, respectively.

To formulate a discrete analogue of (49) for conservation of vorticity, we write (45) in the form

$$J_{i,j}(\zeta, \psi) = \sum_{i',j'} a_{i,j;i+i',j+j'} \zeta_{i+i',j+j'}, \quad (52)$$

where

$$a_{i,j;i+i',j+j'} := \sum_{i'',j''} c_{i',j';i'',j''} \psi_{i+i'',j+j''}. \quad (53)$$

From (52) we see that $a_{i,j;i+i',j+j'} \zeta_{i+i',j+j'}$ is the contribution to the rate of change of vorticity at the point (i, j) due to interaction with the point $(i+i', j+j')$; similarly, $a_{i+i',j+j';i,j} \zeta_{i,j}$ is the contribution to the rate of change of vorticity at the point $(i+i', j+j')$ due to interaction with the point (i, j) . Considering the case where $\zeta = 0$ except at the point (i, j) , we see that conservation of vorticity leads to the condition

$$\sum_{i',j'} a_{i+i',j+j';i,j} = 0 \quad (\text{each } i, j). \quad (54)$$

Substituting from (53) leads to

$$\sum_{i',j'} \sum_{i'',j''} c_{-i',-j';i'',j''} \psi_{i+i'+i'',j+j'+j''} = 0 \quad (\text{each } i, j), \quad (55)$$

which we can rewrite by setting $\hat{i} = i' + i''$ and $\hat{j} = j' + j''$ to obtain

$$\sum_{\hat{i},\hat{j}} \left(\sum_{i',j'} c_{-i',-j';\hat{i}-i',\hat{j}-j'} \right) \psi_{i+\hat{i},j+\hat{j}} = 0 \quad (\text{each } i, j). \quad (56)$$

Since this must hold for all choices of ψ , we conclude that the term in parentheses must vanish for all \hat{i} and \hat{j} . Relabeling \hat{i} and \hat{j} as i'' and j'' , respectively, leads to

$$\sum_{i',j'} c_{-i',-j';i''-i',j''-j'} = 0 \quad (\text{each } i'', j''), \quad (57)$$

which is the condition on the coefficients $c_{i',j';i'',j''}$ which ensures conservation of vorticity.

Similarly, to formulate a discrete analogue of (50) for conservation of enstrophy, we multiply (52) by $\zeta_{i,j}$ to obtain

$$\zeta_{i,j} J_{i,j}(\zeta, \psi) = \sum_{i',j'} a_{i,j;i+i',j+j'} \zeta_{i,j} \zeta_{i+i',j+j'}. \quad (58)$$

From (58) we see that $a_{i,j;i+i',j+j'} \zeta_{i,j} \zeta_{i+i',j+j'}$ is the contribution to the rate of change of enstrophy at the point (i, j) due to interaction with the point $(i+i', j+j')$; similarly,

$a_{i+i',j+j';i',j'}\zeta_{i+i',j+j'}\zeta_{i,j}$ is the contribution to the rate of change of enstrophy at the point $(i+i',j+j')$ due to interaction with the point (i,j) . Since these two must cancel to avoid false production of enstrophy, we see that conservation of enstrophy leads to the condition

$$a_{i,j;i+i',j+j'} = -a_{i+i',j+j';i,j} \quad (\text{each } i, j; i', j'). \quad (59)$$

Substituting from (53) leads to

$$\sum_{i'',j''} c_{i',j';i'',j''} \psi_{i+i'',j+j''} = - \sum_{i'',j''} c_{-i',-j';i'',j''} \psi_{i+i'+i'',j+j'+j''} \quad (\text{each } i, j; i', j'), \quad (60)$$

which we can rewrite by setting $\hat{i} = i' + i''$ and $\hat{j} = j' + j''$ to obtain

$$\sum_{i'',j''} c_{i',j';i'',j''} \psi_{i+i'',j+j''} = - \sum_{\hat{i},\hat{j}} c_{-i',-j';\hat{i}-i',\hat{j}-j'} \psi_{i+\hat{i},j+\hat{j}} \quad (\text{each } i, j; i', j'). \quad (61)$$

Relabeling \hat{i} and \hat{j} as i'' and j'' , respectively, we see that since (61) must hold for all choices of ψ , we must have

$$c_{i',j';i'',j''} = -c_{-i',-j';i''-i',j''-j'} \quad (\text{each } i', j'; i'', j''), \quad (62)$$

which is the condition on the coefficients $c_{i',j';i'',j''}$ which ensures conservation of enstrophy. As a special case, we note that with $i' = j' = 0$, (62) reduces to

$$c_{0,0;i'',j''} = 0 \quad (\text{each } i'', j''), \quad (63)$$

which says that any discrete Jacobian which conserves enstrophy cannot involve the value of $\zeta_{i,j}$ when applied at the point (i,j) . It is interesting to note that by using (47) and (62), we can derive (57); that is, for a consistent discretization, conservation of enstrophy suffices to ensure conservation of vorticity. However, the converse is not true in general: a consistent discretization may conserve vorticity without conserving enstrophy.

Similarly, to formulate a discrete analogue of (51) for conservation of kinetic energy, we write (45) in the form

$$J_{i,j}(\zeta, \psi) = \sum_{i'',j''} b_{i,j;i+i'',j+j''} \psi_{i+i'',j+j''}, \quad (64)$$

where

$$b_{i,j;i+i'',j+j''} := \sum_{i',j'} c_{i',j';i'',j''} \zeta_{i+i',j+j'}. \quad (65)$$

Proceeding exactly as before, we find that conservation of kinetic energy requires

$$b_{i,j;i+i'',j+j''} = -b_{i+i'',j+j'';i,j} \quad (\text{each } i, j; i'', j''), \quad (66)$$

which in turn implies the condition

$$c_{i',j';i'',j''} = -c_{i'-i'',j'-j'';-i'',-j''} \quad (\text{each } i', j'; i'', j''), \quad (67)$$

which is the condition on the coefficients $c_{i',j';i'',j''}$ which ensures conservation of kinetic energy. As a special case, we note that with $i'' = j'' = 0$, (67) reduces to

$$c_{i',j';0,0} = 0 \quad (\text{each } i', j'), \quad (68)$$

which says that any discrete Jacobian which conserves kinetic energy cannot involve the value of $\psi_{i,j}$ when applied at the point (i, j) .

Alternatively, we can note that the continuous Jacobian satisfies the antisymmetry property $J(\zeta, \psi) = -J(\psi, \zeta)$, so we can require the discrete Jacobian to satisfy

$$J_{i,j}(\zeta, \psi) = -J_{i,j}(\psi, \zeta) \quad (\text{each } i, j). \quad (69)$$

Substituting from (45) then yields the condition

$$c_{i',j';i'',j''} = -c_{i'',j'';i',j'} \quad (\text{each } i', j'; i'', j''). \quad (70)$$

From this we see that for an antisymmetric discretization, the first two consistency conditions are equivalent: using (70), (46) implies (47). Similarly, an antisymmetric discretization which conserves enstrophy also conserves kinetic energy (and vice versa): using (70), (62) implies (67).

The discrete Jacobian and some of the conditions derived above can be expressed conveniently in matrix form. To do so, we index the points at which the values of ζ and/or ψ used in the discrete Jacobian are evaluated as

$$(i + i_k, j + j_k), \quad k = 1, 2, \dots, n. \quad (71)$$

Here we assume the points are distinct, i.e., $(i_k, j_k) = (i_l, j_l)$ only if $k = l$, and in view of (63) and (68) we can assume $(i_k, j_k) \neq (0, 0)$ for all k . Then defining

$$\vec{\zeta}_{i,j} = [\zeta_{i+i_1, j+j_1}, \zeta_{i+i_2, j+j_2}, \dots, \zeta_{i+i_n, j+j_n}]^T \quad (72)$$

and

$$\vec{\psi}_{i,j} = [\psi_{i+i_1, j+j_1}, \psi_{i+i_2, j+j_2}, \dots, \psi_{i+i_n, j+j_n}]^T, \quad (73)$$

we can write (45) in the form

$$J_{i,j}(\zeta, \psi) = \frac{1}{h^2} \vec{\zeta}_{i,j}^T A \vec{\psi}_{i,j} \quad (74)$$

where A is the $n \times n$ matrix with entries

$$A_{k,l} = h^2 c_{i_k, j_k; i_l, j_l}, \quad k, l = 1, 2, \dots, n. \quad (75)$$

The consistency conditions (46) and (47) then correspond to the column and row sum conditions

$$\sum_{k=1}^n A_{k,l} = 0 \quad (l = 1, 2, \dots, n) \quad (76)$$

and

$$\sum_{l=1}^n A_{k,l} = 0 \quad (k = 1, 2, \dots, n), \quad (77)$$

respectively, and the antisymmetry condition (70) takes the matrix form

$$A^T = -A. \quad (78)$$

Table 5: Indexing of nearest-neighbor points for compact Jacobians

k	1	2	3	4	5	6	7	8
i_k	1	-1	0	0	1	-1	1	-1
j_k	0	0	1	-1	1	-1	-1	1

The remaining conditions (48), (57), (62), and (67) all imply conditions on the matrix entries, but the specific form of these conditions depends on the ordering of the points (71).

We turn now to *compact* discretizations, i.e., those involving only nearest-neighbor points. For concreteness, we index those points using the ordering given in Table 5, with which (72) and (73) become

$$\vec{\zeta} = [\zeta_{i+1,j}, \zeta_{i-1,j}, \zeta_{i,j+1}, \zeta_{i,j-1}, \zeta_{i+1,j+1}, \zeta_{i-1,j-1}, \zeta_{i+1,j-1}, \zeta_{i-1,j+1}]^T, \quad (79)$$

and

$$\vec{\psi} = [\psi_{i+1,j}, \psi_{i-1,j}, \psi_{i,j+1}, \psi_{i,j-1}, \psi_{i+1,j+1}, \psi_{i-1,j-1}, \psi_{i+1,j-1}, \psi_{i-1,j+1}]^T. \quad (80)$$

The condition (62) for conservation of enstrophy then implies that A must have the structure

$$A = \begin{bmatrix} 0 & 0 & a & b & c & 0 & d & 0 \\ 0 & 0 & -c & -d & 0 & -b & 0 & -a \\ e & f & 0 & 0 & g & 0 & 0 & h \\ -g & -h & 0 & 0 & 0 & -f & -e & 0 \\ i & 0 & j & 0 & 0 & 0 & 0 & 0 \\ 0 & -j & 0 & -i & 0 & 0 & 0 & 0 \\ k & 0 & 0 & l & 0 & 0 & 0 & 0 \\ 0 & -l & -k & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (81)$$

Applying the antisymmetry condition (78)—which together with conservation of enstrophy ensures conservation of kinetic energy as discussed above—forces $a = -e = -l$, $b = g = -j$, $c = f = -i$, and $d = -h = -k$, so (81) reduces to

$$A = \begin{bmatrix} 0 & 0 & a & b & c & 0 & d & 0 \\ 0 & 0 & -c & -d & 0 & -b & 0 & -a \\ -a & c & 0 & 0 & b & 0 & 0 & -d \\ -b & d & 0 & 0 & 0 & -c & a & 0 \\ -c & 0 & -b & 0 & 0 & 0 & 0 & 0 \\ 0 & b & 0 & c & 0 & 0 & 0 & 0 \\ -d & 0 & 0 & -a & 0 & 0 & 0 & 0 \\ 0 & a & d & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (82)$$

The consistency condition (76) [or (77)] then implies that $c = -b$ and $d = -a$. Finally, the consistency condition (48) then implies that $6h^2(a - b) = 1$, reducing the number of degrees

of freedom to one. If we define $\alpha = 6h^2(a + b)$, we can write the matrix for the resulting discrete Jacobian as

$$A = A_1 + \alpha \hat{A}, \quad (83)$$

where

$$A_1 = \frac{1}{12} \begin{bmatrix} 0 & 0 & 1 & -1 & 1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 1 & 0 & 1 & 0 & -1 \\ -1 & 1 & 0 & 0 & -1 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 & 0 & -1 & 1 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (84)$$

corresponds to the Arakawa Jacobian (18) and

$$\hat{A} = \frac{1}{12} \begin{bmatrix} 0 & 0 & 1 & 1 & -1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 1 & 0 & -1 & 0 & -1 \\ -1 & -1 & 0 & 0 & 1 & 0 & 0 & 1 \\ -1 & -1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (85)$$

Thus, the conservative compact Jacobians form a one-parameter family, with the second-order Arakawa Jacobian corresponding to the choice $\alpha = 0$.

To derive the truncation error for the discrete Jacobian specified by (74) and (83), we expand ζ and ψ in Taylor series around the point (x_i, y_j) . After some algebra we obtain

$$\begin{aligned} J_{i,j}(\zeta, \psi) &= J(\zeta, \psi) \\ &+ \frac{h^2}{6} \left[\frac{\partial \zeta}{\partial x} \frac{\partial^3 \psi}{\partial y^3} - \frac{\partial \zeta}{\partial y} \frac{\partial^3 \psi}{\partial x^3} + \frac{\partial \psi}{\partial y} \frac{\partial^3 \zeta}{\partial x^3} - \frac{\partial \psi}{\partial x} \frac{\partial^3 \zeta}{\partial y^3} \right. \\ &+ \left(\frac{\partial \zeta}{\partial x} \frac{\partial^3 \psi}{\partial x^2 \partial y} - \frac{\partial \zeta}{\partial y} \frac{\partial^3 \psi}{\partial x \partial y^2} \right) + \left(\frac{\partial^2 \zeta}{\partial x^2} - \frac{\partial^2 \zeta}{\partial y^2} \right) \frac{\partial^2 \psi}{\partial x \partial y} \\ &- \left(\frac{\partial \psi}{\partial x} \frac{\partial^3 \zeta}{\partial x^2 \partial y} - \frac{\partial \psi}{\partial y} \frac{\partial^3 \zeta}{\partial x \partial y^2} \right) - \left(\frac{\partial^2 \psi}{\partial x^2} - \frac{\partial^2 \psi}{\partial y^2} \right) \frac{\partial^2 \zeta}{\partial x \partial y} \\ &\left. + \frac{\alpha h^2}{12} \left[\frac{\partial^2 \zeta}{\partial y^2} \frac{\partial^2 \psi}{\partial x^2} - \frac{\partial^2 \zeta}{\partial x^2} \frac{\partial^2 \psi}{\partial y^2} \right] \right. \\ &\left. + 2 \left(\frac{\partial \zeta}{\partial y} \frac{\partial^3 \psi}{\partial x^2 \partial y} - \frac{\partial \zeta}{\partial x} \frac{\partial^3 \psi}{\partial x \partial y^2} + \frac{\partial \psi}{\partial x} \frac{\partial^3 \zeta}{\partial x \partial y^2} - \frac{\partial \psi}{\partial y} \frac{\partial^3 \zeta}{\partial x^2 \partial y} \right) \right] + O(h^4). \end{aligned} \quad (86)$$

Since the $O(h^2)$ terms in (86) do not vanish for any choice of α , all compact conservative Jacobians are second-order accurate: there is no compact conservative Jacobian of the form (45) with fourth-order accuracy.

To find the optimum choice of the parameter α , consider the case where the streamfunction consists of a single wave

$$\psi = -UY + A \sin(kX), \quad (87)$$

where $X = x \cos(\theta) + y \sin(\theta)$ and $Y = -x \sin(\theta) + y \cos(\theta)$ are coordinates rotated by the angle θ as considered by Arakawa[1]. The corresponding vorticity is

$$\zeta = -k^2 A \sin(kX). \quad (88)$$

Making these substitutions in (86) leads to the truncation error

$$U \frac{\partial \zeta}{\partial X} \frac{k^2 h^2}{6} \left[(\cos^2 \theta + \sin^2 \theta)^2 - \alpha \cos \theta \sin \theta \right] + O(h^4). \quad (89)$$

As Arakawa pointed out, since the first term in the brackets is independent of θ , the corresponding discrete Jacobian—to leading order—is isotropic. However, the second term does not share this property (and it actually increases the truncation error for some values of θ). For example, the special cases $\alpha = 1$ and $\alpha = -1$ lead to particularly simple compact conservative Jacobians, but each exhibits a directional bias: the former omits values at the points $(i + 1, j + 1)$ and $(i - 1, j - 1)$, and the latter omits values at the points $(i + 1, j - 1)$ and $(i - 1, j + 1)$. Thus, the optimum choice of α (in the sense of generating a truncation error which is minimum and isotropic) is $\alpha = 0$, giving the Arakawa Jacobian.