Higher-Order Compact Schemes for Numerical Simulation of Incompressible Flows

Robert V. Wilson and Ayodeji O. Demuren
Old Dominion University

Mark Carpenter
NASA Langley Research Center

Institute for Computer Applications in Science and Engineering
NASA Langley Research Center
Hampton, VA

Operated by Universities Space Research Association

National Aeronautics and
Space Administration

February 1998

HIGHER-ORDER COMPACT SCHEMES FOR NUMERICAL SIMULATION OF INCOMPRESSIBLE FLOWS*

ROBERT V. WILSON[†], AYODEJI O. DEMUREN[†] AND MARK CARPENTER[‡]

Abstract. A higher order accurate numerical procedure has been developed for solving incompressible Navier-Stokes equations for 2D or 3D fluid flow problems. It is based on low-storage Runge-Kutta schemes for temporal discretization and fourth and sixth order compact finite-difference schemes for spatial discretization. The particular difficulty of satisfying the divergence-free velocity field required in incompressible fluid flow is resolved by solving a Poisson equation for pressure. It is demonstrated that for consistent global accuracy, it is necessary to employ the same order of accuracy in the discretization of the Poisson equation. Special care is also required to achieve the formal temporal accuracy of the Runge-Kutta schemes. The accuracy of the present procedure is demonstrated by application to several pertinent benchmark problems.

Key words. compact schemes, incompressible flow simulation

Subject classification. Fluid Mechanics

1. Introduction. For direct numerical simulation (DNS) of fluid flow problems, it is generally accepted that higher-order accurate methods must be used to minimize dissipation and dispersion errors. As the flow Reynolds number increases so do the ranges of temporal and spatial scales which must be resolved. Thus, the number of grid points-per-wavelength (PPW) required by the numerical scheme for approximation of the flow equations to within acceptable tolerances of dissipation and dispersion errors effectively limits the smallest scales that can be computed accurately, and thereby also the maximum Reynolds number. Spectral methods require the fewest PPW, namely two, and are therefore ideal for computations of flows with periodic boundary conditions. For more general flow problems finite-difference methods are desirable. Lele [1] has analyzed the resolution qualities of several finite-difference schemes. In general, resolution increased, i.e., fewer PPW, the larger the computational grid stencil, and implicit compact schemes had better resolution than regular explicit schemes of the same order of accuracy and computational stencil. Further, high resolution properties could be improved by optimization of coefficients, but at the expense of the formal order of accuracy. A fourth-order compact scheme was devised with nearly the resolution quality of spectral methods, but retaining the flexibility of finite-difference methods. Haras and Ta'asan [2] analyzed the accuracy of various compact schemes, and demonstrated that resolution efficiency was not synonymous with minimum truncation error over the whole range of wavelengths present in a problem. The suggested that a scheme should be optimized for global accuracy rather than resolution efficiency. Hu et al. [3] have also shown that, in applications of interest to computational acoustics, temporal resolution could be improved by optimization of the coefficients of any multi-stage, Runge-Kutta, time-advancing approximation scheme.

In large eddy simulation (LES) of turbulent flows, the goal is not to resolve all the scales in the flow, but only the larger scales. Effects of the unresolved smaller scales are approximated with sub-grid scale (SGS) models. Therefore, second-order, central-difference schemes are often used [4,5]. A further justification is that effects of truncation errors may be comparable to uncertainties inherent in the SGS models. However, questions remain as to how large the "large" scales are, and how many PPW are required to resolve the smallest scale in the range. Further, what are the consequences of inadequate resolution of such second-order schemes. It is obvious that if LES is to be used in computational acoustics, dispersion errors are unacceptable [6]. Higher-order-accurate (greater than second-order) methods guarantee much better convergence towards grid independence, along with better wave-number resolution. In addition, implicit (compact) finite-difference schemes require narrower computational grid stencils, have better fine-scale resolution and yield better global accuracy than explicit finite-difference schemes with the same formal order of accuracy. Therefore, the present study focuses on the use of higher-order (fourth and sixth) compact schemes for the simulation of incompressible fluid flow problems.

The lack of an evolution equation for the pressure presents particular difficulty in the computation of incompressible flows, which is absent in compressible flows. In the latter, the Navier-Stokes equations, along with the continuity equation and the energy equation present evolution equations for the five variables, namely three momentum components, density and enthalpy, which can be advanced numerically with the same temporal and spatial discretization schemes. The pressure can then be obtained from an equation of state [7]. But in the former, an auxiliary equation has to be derived for the pressure which is then solved to satisfy the divergence-free velocity-field condition required for incompressibility [8,9,10]. Bell et al. [8] proposed a second-order projection method. Henshaw [9] derived a Poisson equation for pressure which was solved by a fourth-order explicit finite-difference method. It was found necessary to introduce a damping term to reduce divergence errors. In their finite-difference formulation for 2D problems, Joslin et al. [10] used an influence matrix method to solve the Poisson equation for pressure. But this method is of order N^2 for 2D problems and order N^3 for 3D problems, so that memory requirements and computation work quickly become prohibitive for 3D computations. If any of the directions has periodic boundaries, then mixed finite-difference/spectral methods could be used [10,11], resulting in an Helmholtz equation for pressure which could be solved more cheaply. In the present study, interest is mainly in a flexible numerical scheme which could be used for non-regular computational domains and non-uniform computational grids, and with consistent treatment of all equations. Therefore, the Poisson equation for the pressure will be approximated with the same compact finite-difference scheme as used in the Navier-Stokes equations. Extension to irregular grids in physical space simply requires transformation of the equations onto a regular grid in computational space. The metrics of the transformation must be computed with the same compact finite-difference scheme to guarantee a consistent level of accuracy.

2. Analysis.

2.1. Governing Differential Equations.
The partial differential equations governing the incompressible fluid flow are the Navier-Stokes equations which can be written in Cartesian tensor form, for dimensionless variables as:

$$(1) \qquad \frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{\partial P}{\partial x_i} + \frac{1}{Re_L} \frac{\partial^2 u_i}{\partial x_j \partial x_j}$$

where, u_i are the Cartesian velocity components in the Cartesian coordinate directions x_i, P is the pressure and Re_L is the Reynolds number based on the characteristic length, L. These equations must be solved in conjunction with the continuity equation:

$$(2) \qquad \frac{\partial u_i}{\partial x_i} = 0$$

which expresses the divergence-free velocity condition. In 2D, i or $j = 1,2$ and in 3D, i or $j = 1,2,3$. Einstein's summation rule for repeated indices is presumed.

2.2. Temporal Discretization. The time advancement scheme for the momentum equations should possess several qualities such as low dispersion and dissipation errors over a wide range of step sizes, low memory storage requirements, and a relatively large stability envelope. A family of low-storage, RK schemes proposed by Williamson [12] possesses these desirable qualities. The schemes are low-storage in the sense that only two storage locations (one for the time derivative and one for the variable itself) are required for the time advancement. In comparison, a third-order fully implicit scheme would require four storage locations.

The Navier-Stokes equations (1) are discretized temporally with explicit Runge-Kutta (RK) schemes, and spatially with implicit compact finite difference schemes. The momentum equation is advanced from time level, n, to $n+1$, in Q substages using either a third- or fourth-order explicit RK scheme. The advancement from substage, M, to $M+1$, is defined by:

$$(3) \qquad u_i^{M+1} = u_i^M + b^{M+1} \Delta t \left(H_i^M - \delta_{x_i} P_i^M \right)$$

where Δt is the time step, b^{M+1} is a coefficient of the RK scheme, and u_i^M represents the x_i velocity component at the M_{th} substage. The substage, $M = 0$, is equivalent to the n_{th} time level and the substage, $M = Q$-1, is equivalent to the $n+1_{th}$ time level. The term, $H_i^M = -u_j \delta_{x_j} u_i^M + \frac{1}{Re_L} \delta_{x_j x_j} u_i^M + a^M H_i^{M-1}$ is the sum of the convection and diffusion terms, plus accumulation from the previous sub-stage. δ_{x_j} and $\delta_{x_j x_j}$ are compact first and second derivative operators, respectively, to be addressed in the next sub-section. The terms on the right hand side (RHS) of Eq. (3) are assumed known from the previous sub-stage or from the initial conditions at $t = 0$. The calculation of the pressure is accomplished by solving a Poisson equation at each sub-stage such that the continuity is enforced. Since the pressure, P^M, is calculated before the advancement of Eq. (3), u_i^{M+1}, can be calculated explicitly.

The low-storage requirement is accomplished by continuously overwriting the storage location for the time derivatives and unknown variables at each sub-stage:

$$(4) \qquad \hat{H}_i^M \leftarrow a^M \hat{H}_i^{M-1}$$

3

$$(5) \qquad u_i^{M+1} \leftarrow u_i^M + b^{M+1} \Delta t \hat{H}_i^M$$

where $\hat{H}_i^M = H_i^M - [\partial P^M / \partial x_i]$ and the notation $\leftarrow$ is used to indicate that the storage locations, $\hat{H}_i^{M-1}$, u_i^M are overwritten by $\hat{H}_i^M$, u_i^{M+1}, respectively. Tables 1 and 2 give values of the coefficients, a^M and b^M for some low-storage, three-stage-third-order and five-stage-fourth-order schemes, respectively.

Table 1. Coefficients of a three-stage-third-order Runge-Kutta scheme, from Lowery and Reynolds [13]

M	a^M	b^M
1	0	0.500
2	-0.68301270	0.91068360
3	-1.33333333	0.36602540

Table 2. Coefficients of a five-stage-fourth-order Runge-Kutta scheme, from Carpenter and Kennedy [14]

M	a^M	b^M
1	0	0.14965902
2	-0.41789047	0.37921031
3	-1.19215169	0.82295502
4	-1.69778469	0.69945045
5	-1.51418344	0.15305724

The stability characteristics of RK schemes can be analyzed by considering the model equation:

$$(6) \qquad \frac{\partial \phi}{\partial t} = \hat{H}(\phi, t)$$

where ϕ is the generic unknown to be advanced in time and $\hat{H}$ is the time derivative which contains the spatial terms of the governing equation. Equation (6) is transformed from physical space to wavenumber space by decomposing ϕ into Fourier modes:

$$(7) \qquad \phi = \tilde{\phi}(t) e^{ikx}$$

where $\tilde{\phi}(t)$ is the Fourier coefficient of ϕ, $i = \sqrt{-1}$, and k is the wavenumber. Substituting Eq. (7) into Eq. (6) yields:

$$(8) \qquad \frac{\partial}{\partial t} \tilde{\phi} = \lambda \tilde{\phi}$$

where λ (a complex number) is the Fourier symbol of the spatial operator $\hat{H}$. The RK scheme is used to expand the term on the LHS of Eq. (8). This gives the amplification factor, $G = \tilde{\phi}^{n+1}/\tilde{\phi}^n$ for the third-order scheme as:

(9)
$$G = 1 + (\lambda \Delta t) + \frac{1}{2}(\lambda \Delta t)^2 + \frac{1}{6}(\lambda \Delta t)^3$$

It can be shown that all three-stage, third-order RK schemes have the same amplification factor given in Eq. (9). Analytical solution of Eq. (8) gives the exact amplification factor, G_e:

(10)
$$G_e = e^{\lambda \Delta t}$$

Comparing Eqs. (9) and (10), one sees that the three-stage, third-order RK scheme is a polynomial approximation to the exact solution to third-order. Similarly, the five-stage, fourth-order RK scheme has amplification factor:

(11)
$$G = 1 + (\lambda \Delta t) + \frac{1}{2}(\lambda \Delta t)^2 + \frac{1}{6}(\lambda \Delta t)^3 + \frac{1}{24}(\lambda \Delta t)^4 + \frac{1}{200}(\lambda \Delta t)^5$$

The stability of the RK schemes is shown graphically in Fig. 1 by plotting the $|G| = 1$ contour of Eq. (9) for the three-stage, third-order scheme and Eq. (11) for the five-stage, fourth-order scheme. A selection of $\lambda \Delta t$ which lies in the interior of the closed curve yields $|G| < 1$, i.e. the scheme is stable. Outside the closed curve, $|G| > 1$ and the scheme is unstable. If the Fourier symbol of the spatial operator, λ, is purely imaginary (for example the 1-D convection equation) an inspection of Fig. 1 reveals that the region, $-L_I < \lambda \Delta t < L_I$, is stable. If λ is purely real (for example the 1-D diffusion equation) the region, $-L_R < \lambda \Delta t < 0$, is stable. The stability limits for these two extreme cases are given in Table 3 for the third- and fourth-order RK schemes. Hence, the fourth-order scheme would allow time steps roughly twice that of the third-order scheme.

Temporal stability analysis of the Navier-Stokes equations is modeled after the stability of the convection-diffusion equation for which λ is, in general, complex.

Table 3. Stability limits of Runge-Kutta schemes for purely imaginary (L_I) or real (L_R) spatial operators

Spatial Operator	third-order, three stage	fourth-order, five stage
Imag, L_I	1.73	3.34
Real, L_R	2.51	4.65

2.3. Spatial Discretization. The numerical approximations to the spatial derivatives appearing in the momentum equations, Eq. (3), are given in this section. Standard second-order finite difference approximations to first derivative terms suffer from large dispersion errors. Spectral methods offer exact differentiation for resolved modes but suffer from high cost and low flexibility in that simple domains and boundary conditions are required for their implementation. In this study, high-order compact finite differences are preferred due to the combination of high-accuracy, flexibility, and relative operation count.

2.3.1. First Derivative Terms.

The first derivative terms appearing in the governing equations are approximated using fourth- and sixth-order compact finite difference schemes described by Lele [1]. Higher accuracy derives from the implicit treatment of derivatives, via:

$$(12) \qquad \alpha\phi'_{i-1} + \phi'_i + \alpha\phi'_{i+1} = \frac{a}{2\Delta x}(\phi_{i+1} - \phi_{i-1}) + \frac{b}{4\Delta x}(\phi_{i+2} - \phi_{i-2})$$

(or in matrix form: $A_x\phi' = B_x\phi$ or $\phi' = A_x^{-1}B_x\phi$), where $\Delta x = L_x/(N_x - 1)$, N_x is the number of grid points, ϕ'_i represents the first derivative of the generic variable ϕ_i with respect to x, and α, a, b are the coefficients of the compact scheme which determine the accuracy. Similar expressions are used for derivatives with respect to the y and z directions. For the fourth-order scheme: $\alpha = 1/4$, $a = 3/2$, and $b = 0$, and for the sixth-order scheme: $\alpha = 1/3$, $a = 14/9$, and $b = 1/9$. The LHS of Eq. (12) contains the unknown derivatives at grid points i and $i \pm 1$ while the RHS contains the known functional values ϕ_i at the grid points $i \pm 1$ and $i \pm 2$. A_x is a tridiagonal $N_x \times N_x$ matrix and B_x is a tridiagonal matrix for the fourth-order scheme and pentadiagonal matrix for the sixth-order scheme. In contrast, A_x will be diagonal in an explicit finite-difference scheme.

A comparison of explicit central difference and implicit compact approximations to the first derivative is given in Table 4. It is seen that the implicit treatment of the derivative allows for more "compact" stencils, for given order. Also, the coefficient of the leading truncation-error term is reduced by a factor of 4 for the fourth-order scheme and 9 for the sixth-order scheme in comparison to explicit central difference schemes of the same order.

Table 4. Comparison of explicit central difference and implicit compact approximations to the first derivative

Scheme	Truncation error	Stencil Size
fourth-order central	$(-4/5!)(\Delta x)^4\phi^{(5)}$	5
fourth-order compact	$(-1/5!)(\Delta x)^4\phi^{(5)}$	3
sixth-order central	$(-36/7!)(\Delta x)^6\phi^{(7)}$	7
sixth-order compact	$(-4/7!)(\Delta x)^6\phi^{(7)}$	5

The set of Eq. (13) at all grid points results in a tridiagonal system of algebraic equations and that is solved efficiently by factoring the LHS into a lower/upper (LU) system once at the beginning of the simulation. The LU factors are stored and then used to solve Eq. (12) for the unknown derivatives.

The resolution properties of the numerical approximation to the first derivative can be analyzed by transforming the 1-D convection equation from physical to wavenumber space[1]. For explicit finite-difference schemes, $\alpha = 0$, the numerical wavenumber is given by:

$$(13) \qquad k^* = \frac{-i}{\Delta x} \sum_{l=-N}^{N} a_l e^{ilk\Delta x}$$

6

while for the tridiagonal compact scheme, the numerical wavenumber is given by:

$$(14) \qquad k^* = \frac{1}{\Delta x} \left[\frac{a \sin(k\Delta x) + \frac{b}{2} \cos(2k\Delta x)}{1 + 2\alpha \cos(k\Delta x)} \right]$$

Note that in general the numerical wavenumber, k*, is complex while the exact wavenumber, k, is real. For the numerical approximation to yield an exact solution, the following two conditions must be met:

$$(15) \qquad \textit{Real}(k^*) \ = \ k$$

$$(16) \qquad \textit{Imag}(k^*) \ = \ 0$$

Deviations from Eq. (15) indicate dispersion errors due to odd derivative terms appearing in the truncation error, and deviations from Eq. (16) indicate dissipation errors due to even derivative terms appearing in the truncation error. The real and imaginary parts of Eqs. (13) and (14) are plotted separately in Fig. 2, for several differencing schemes. All four approximations do a reasonable job of approximating the exact wavenumber (i.e. very low dispersion errors) at very low wavenumbers ($k\Delta x \to 0$), but they all do a poor job at very high wavenumbers ($k\Delta x \to \pi$). For intermediate wavenumbers, the fourth- and sixth-order compact schemes provide a much better approximation to the exact wavenumber over a greater range of wavenumbers than the explicit schemes. The second-order central difference scheme yields a poor approximation to the exact wavenumber for all but the very lowest wavenumbers ($k\Delta x < 0.5$). Due to symmetry, central difference schemes always have real wavenumbers, hence they contain no dissipation errors. Only the third-order upwind scheme has numerical dissipation errors. Spectral methods yield exact differentiation for all modes which can be resolved on the specified grid and thus correspond to the exact relationship for $k\Delta x$ in Fig. 2.

Table 5 lists some quantitative measures of resolution for the five schemes. The wavenumber, k_c^*, defines the region of acceptable accuracy, i.e. for $0 < k < k_c^*$, $\left| k^* \Delta x - k \Delta x \right| < 0.01$. Modes with $k > k_c^*$, are not accurately resolved. The quantity, k_{max}^*, defines the maximum value for the modified wavenumber, i.e. for $k < k_{max}^*$, the slope of the curve is zero. Also listed is the number of spatial grids points per wavelength, $PPW = 2\pi/(k_c^* \Delta x)$, to accurately resolve a given mode. From the estimate of PPW, roughly five times as many points are required for the second-order central difference scheme to achieve the same accuracy as the compact schemes.

7

Table 5. Resolution measures of various numerical approximations to the first derivative

Spatial Scheme	$k_c^*\Delta x$	$k_{max}^*\Delta x$	Points per wavelength
second-order central	0.22	1.00	28.6
third-order upwind	0.44	1.27	14.3
fourth-order compact	1.11	1.73	5.6
sixth-order compact	1.55	2.00	4.1
spectral	π	π	2

For non-periodic boundaries, one-sided finite difference expressions are required to close the system of equations at the boundary points; $i = 1$ and $i = N$ for the fourth-order scheme and $i = 1, 2$ and $i = N\text{-}1$, N for the sixth-order scheme. A third-order compact boundary scheme is used at $i = 1$ and $i = N$ with the fourth-order interior scheme:

$$(17) \qquad \phi'_1 + \alpha_{bs}\phi'_2 = \frac{1}{\Delta x} \sum_{i=1}^{3} a_{bs_i}\phi_i$$

where $\alpha_{bs} = 2$ and $a_{bs_1} = -5/2$, $a_{bs_2} = 2$, $a_{bs_3} = 1/2$ are the coefficients of the third-order boundary scheme. A similar equation is used at $i = N$.

For the sixth-order scheme, a boundary and near-boundary scheme are required for closure since the interior stencil is pentadiagonal. A fifth-order explicit boundary scheme is used at points, $i = 1$ and $i = N$:

$$(18) \qquad \phi'_1 = \frac{1}{\Delta x} \sum_{i=1}^{8} a_{bs_i}\phi_i$$

with coefficients:

$$a_{bs_1} = -296/105 \qquad a_{bs_5} = -215/12$$

$$a_{bs_2} = 415/48 \qquad a_{bs_6} = 791/80$$

$$a_{bs_3} = -125/8 \qquad a_{bs_7} = -25/8$$

$$a_{bs_4} = 985/48 \qquad a_{bs_8} = 145/336$$

A different fifth-order explicit near boundary scheme is used at points, $i = 2$ and $i = N\text{-}1$:

(19)
$$\phi'_2 = \frac{1}{\Delta x} \sum_{i=1}^{8} a_{nb_i} \phi_i$$

with coefficients:

$$a_{nb_1} = -3/16 \qquad a_{nb_5} = 115/144$$

$$a_{nb_2} = -211/180 \qquad a_{nb_6} = -1/3$$

$$a_{nb_3} = 109/48 \qquad a_{nb_7} = 23/240$$

$$a_{nb_4} = -35/24 \qquad a_{nb_8} = -1/72$$

The boundary schemes given by Eqs (18) and (19) were shown to be asymptotically stable by Carpenter [15]. Similar equations for the boundary and near boundary schemes are used at points $i = N$ and $i = N\text{-}1$.

2.3.2. Second Derivative Terms. The second derivative terms present in the viscous terms of the momentum equation and the Laplacian operator of the Poisson equation for pressure are approximated using fourth- and sixth-order compact finite differences. Again, higher accuracy is achieved by treating the derivative implicitly:

$$\alpha\phi''_{i-1} + \phi''_i + \alpha\phi''_{i+1} = \frac{a}{(\Delta x)^2}(\phi_{i+1} - 2\phi_i + \phi_{i-1})$$

(20)
$$+ \frac{b}{4(\Delta x)^2}(\phi_{i+2} - 2\phi_i + \phi_{i-2})$$

(or in matrix form: $A_{xx}\phi'' = B_{xx}\phi$ or $\phi'' = A_{xx}^{-1}B_{xx}\phi$), where ϕ''_i represents the second derivative of the generic variable ϕ_i with respect to x, and α, a, b are the coefficients of the compact scheme. For the fourth-order scheme: $\alpha = 1/10$, $a = 6/5$, and $b = 0$, and for the sixth-order scheme: $\alpha = 2/11$, $a = 12/11$, and $b = 3/11$. The tridiagonal system of algebraic equations for the second derivatives is solved for in the same manner as the first derivatives. A comparison of explicit central difference and implicit compact approximations to the second derivative is given in Table 6. As with the first derivative, the implicit treatment of the second derivative results in a smaller stencil size for a given order. The lead-

9

ing truncation error term for the compact formulation is reduced by factors of 2 for the fourth-orde scheme and 4 for the sixth-order scheme, in comparison to explicit central difference schemes of the same order.

Table 6. Comparison of explicit central difference and implicit compact approximations of the second derivative

Scheme	Truncation error	Stencil Size
fourth-order central	$(-8/6!)(\Delta x)^4 \phi^{(6)}$	5
fourth-order compact	$(-3.6/6!)(\Delta x)^4 \phi^{(6)}$	3
sixth-order central	$(-72/8!)(\Delta x)^6 \phi^{(8)}$	7
sixth-order compact	$(-16.7/8!)(\Delta x)^6 \phi^{(8)}$	5

For non-periodic boundaries, one-sided finite differences are required to close the system of equations. At i = 1 and i = N, a third-order compact boundary scheme is used:

$$(21) \qquad \phi''_1 + \alpha_{bs}\phi''_2 = \frac{1}{(\Delta x)^2}\sum_{i=1}^{4} a_{bs_i}\phi_i$$

where $\alpha_{bs} = 11$ and $a_{bs_1} = 13$, $a_{bs_2} = -27$, $a_{bs_3} = 15$, and $a_{bs_4} = -1$ are the coefficients of the third-order boundary scheme. For the sixth-order scheme, a near boundary scheme is required at i = 2 and i = N-1. The fourth-order interior scheme is used at these points since only a three-point stencil is needed.

A similar analysis of the 1-D diffusion equation is used to investigate the resolution qualities of the proposed compact approximation to the second derivative. The tridiagonal compact scheme represented by Eq. (20) yields a numerical wavenumber:

$$(22) \qquad (k^*)^2 = \frac{1}{(\Delta x)^2}\left[\frac{2a[1 - cos(k\Delta x)] + \frac{b}{2}[1 - cos(2k\Delta x)]}{1 + 2\alpha cos(k\Delta x)}\right]$$

The dissipation errors for the explicit second-order central difference, and fourth- and sixth-order compact schemes are shown in Fig. 3. We see that the numerical wavenumber for the compact scheme more closely approximates the exact wavenumber over a wider range of wavenumbers. Quantitative measures of resolution power for the various schemes are given in Table 7. Estimates of the PPW show that roughly twice as many points are required for the explicit second-order central difference to produce the same accuracy as the compact scheme.

10

Table 7. Resolution measures of various numerical approximations to the second derivative

Spatial Scheme	$(k_c^* \Delta x)$	$(k_{max}^* \Delta x)^2$	Points per wavelength
second-order central	0.57	4.00	11.0
fourth-order compact	1.14	6.00	5.5
sixth-order compact	1.52	6.86	4.1
spectral	π	π^2	2

2.4. Enforcement of the Continuity Equation and Poisson Equation for Pressure.

An examination of the governing equations reveals four scalar equations (continuity and three scalar components of the momentum equation) in terms of four unknowns (three velocity components and pressure). Time derivatives for the velocity components in the momentum equation are used to march those equations in time. However, in an incompressible flow, no such time derivative exists for pressure, but the continuity equation gives an additional constraint on the velocity field, in order to enforce the divergence-free condition. The current approach overcomes this problem by taking the numerical divergence of the discretized momentum equation and substituting in the discretized continuity equation. This results in a Poisson equation for pressure which is solved to ensure that the velocity field is divergence free at the M+1$_{th}$ substage. The problem does not arise in a compressible flow since the density appears as a natural choice for the fourth variable, and the continuity equation contains its time derivative to be used for time-advancement. The pressure is then obtained from an equation of state.

Applying the divergence operator δ_{x_j} to the discretized momentum Eq. (3) gives:

(23)
$$\frac{1}{b^{M+1} \Delta t}\left\{\delta_{x_j}(u_i^{M+1} - u_i^M)\right\} = \delta_{x_j} H_i^M - \delta_{x_j}\delta_{x_j} P^M$$

The term, $\delta_{x_j} u_i^{M+1}$, represents the discretized continuity equation at the $M + 1$ sub-stage and is set to zero to enforce the divergence-free condition. The term, $\delta_{x_j} u_i^M$, represents the divergence of the velocity field at the previous substage M. This term should also be zero, but in practice, it is retained to "kill off" any accumulation from previous substages due to lack of convergence, etc. The term, $\delta_{x_j} H_i^M$, is the source term of the Poisson equation and represents gradients of the convection and diffusion terms which are known from the previous sub-stage. The term, $\delta_{x_j}\delta_{x_j} P^M$, represents the discretized Laplacian operator on the pressure. Hence, the Poisson equation to be solved for the pressure is:

(24)
$$\nabla^2 P^M = \delta_{x_j}\left[H_i^M + \frac{u_i^M}{b^{M+1} \Delta t}\right]$$

The solution details follow.

3. Solution of the Poisson equation for pressure. A significant amount of the total computational time required for the solution of the incompressible Navier-Stokes equations (as much as half) is devoted to the enforcement of the continuity equation/solution of pressure. This stems from the fact that evolution equations exist for the velocity components (i.e. the momentum equations) while none exist for the pressure. Instead, an elliptic equation must be solved for the pressure which involves the solution of a system of equations and is expensive. Solutions methods for elliptic equations generally fall into two categories - direct or iterative. Direct methods usually involve some form of Gaussian elimination where the coefficient matrix is first factored into an upper and lower matrix and then the solution is computed using back substitution. The operation count and memory requirements for this procedure can be prohibitively large for the solution of systems involving a large number of unknowns ($\sim 10^6$ in typical 3-D problems). The alternative to a direct solution is an iterative procedure where an initial approximation to the solution is used to yield an improved solution. This process is repeated until the solution is converged to within a desired tolerance. The operation count and memory requirements of most iterative methods are less than that of Gaussian elimination. Therefore, the iterative solution procedure is used in this study to solve the Poisson equation for pressure. The details of this procedure are outlined in this chapter.

3.1. Discretized Laplacian Operator. The discrete Poisson equation for pressure which was derived in Sect. 2.4 is given by:

$$(25) \qquad \delta_{x_i} \delta_{x_i} P^M = \delta_{x_i} \left[H_i^M + \frac{u_i^M}{b^{M+1} \Delta t} \right]$$

The LHS of Eq. (25) represents a discretized Laplacian operator composed of two applications of the first derivative operator, δ_{x_i}. It is well known that using two first derivative operators to represent the Laplacian operator on non-staggered grids can lead to an "odd-even" decoupling of the solution. Indeed, with standard second-order central differencing for the first derivative operator, the solution at even grid points completely decouples from the odd grid points, leading to non-physical results. One remedy is to introduce terms of the same order as the truncation error which in effect replaces the two first derivative operators with a single second derivative operator. This couples the solution at odd and even grid points while maintaining the same formal order of accuracy. The Laplacian operator is discretized using a single second derivative operator to prevent possible decoupling and Eq. (25) becomes:

$$(26) \qquad \delta_{xx_i} P^M = \delta_{x_i} \left[H_i^M + \frac{u_i^M}{b^{M+1} \Delta t} \right]$$

where $\delta_{xx_i} P^M$ represents the discrete Laplacian of pressure and is discretized using the compact second derivative operator given by Eq. (20).

The effect of replacing the two first derivative operators with the single second derivative Laplacian operator is shown by rewriting Eq. (26) as

$$(27) \qquad \delta_{x_ix_i} \tilde{P}^M = \delta_{x_i}\left(H_i^M + \frac{u_i^M}{b^{M+1}\Delta t}\right) - R$$

where an iterative solution is assumed and $\tilde{P}^M$ is the current iterate of pressure and R is the residual imbalance due to incomplete convergence. Taking the divergence of the momentum equation Eq. (3) after the iterative solution of the pressure and time advancement gives:

$$(28) \qquad \delta_{x_i}^2 \tilde{P}^M = \delta_{x_i}\left(H_i^M + \frac{u_i^M}{b^{M+1}\Delta t}\right) - \delta_{x_i}\left(\frac{u_i^{M+1}}{b^{M+1}\Delta t}\right)$$

Subtracting Eqs. (27) from (28), and rearranging gives the expression for the divergence:

$$(29) \qquad \delta_{x_i}u_i^{M+1} = -b^{M+1}\Delta t\left\{\delta_{x_i}^2 - \delta_{x_ix_i}\right\}\tilde{P}^M + R$$

Equation (29) shows that the imbalance of the continuity equation is due to two contributions; (i) the difference between the two different Laplacian operators which is $O(\Delta x^m)$ where m is greater than the order of accuracy of the Laplacian operators (it is easy to show that it is fourth-order for explicit second-order operators), and (ii) the residual imbalance due to incomplete iterative convergence. Equation (29) also shows that the residual imbalance, R, does not need to be driven to machine zero, only below the level of the difference in the two Laplacian operators. However, the convergence tolerance for terminating the iterative solution of the Poisson equation must be sufficiently small for the solution to be insensitive to the stopping criterion. As the grid is refined, the tolerance must be reduced in accordance with the order of accuracy of the scheme. For example, when using the sixth-order scheme on relatively fine grids (see Table 18), it is necessary to converge the solution to machine zero (10^{-14}) in order for the solution errors to reduce at a sixth-order rate.

Equation (26) at all interior grid points results in a system of equations that is solved at each sub-stage of the time advancement scheme. For simplicity, the system of equations is described for the 2-D Poisson equation with periodic boundaries on a uniform grid, for the fourth-order tridiagonal scheme defined in Sec. 2.3. The solution procedure is easily extended to the 3-D Poisson equation, as presented in Appendix A. Extension to general curvilinear grids is presented in Appendix B.

The set of Eq. (26) can be written in the form of a system of equations as:

$$(30) \qquad AP = [A_{xx}^{-1}B_{xx} + A_{yy}^{-1}B_{yy}]P = F$$

13

$$A_{xx} = \begin{bmatrix} \bar{L} & & & & & & \bar{L} \\ & \bar{L} & & & & & \\ & & \circ & & & & \\ & & & \circ & & & \\ & & & & \circ & & \\ & & & & & \bar{L} & \\ \bar{L} & & & & & & \bar{L} \end{bmatrix}, \quad \text{where } \bar{L} = \begin{bmatrix} 1 & \alpha & & & & & \alpha \\ \alpha & 1 & \alpha & & & & \\ & & \circ & & & & \\ & & & \circ & & & \\ & & & & \circ & & \\ & & & & \alpha & 1 & \alpha \\ \alpha & & & & & \alpha & 1 \end{bmatrix}$$

(*nj x nj* block matrix) (*ni x ni* scalar matrix)

$$A_{yy} = \begin{bmatrix} \bar{I} & \alpha\bar{I} & & & & & \alpha\bar{I} \\ \alpha\bar{I} & \bar{I} & \alpha\bar{I} & & & & \\ & & \circ & & & & \\ & & & \circ & & & \\ & & & & \circ & & \\ & & & & \alpha\bar{I} & \bar{I} & \alpha\bar{I} \\ \alpha\bar{I} & & & & & \alpha\bar{I} & \bar{I} \end{bmatrix}, \text{ where } \bar{I} \text{ is the } ni \times ni \text{ identity matrix}$$

(*nj x nj* block matrix)

$$B_{xx} = \frac{a}{h_x^2} \begin{bmatrix} \bar{R} & & & & & & \bar{R} \\ & \bar{R} & & & & & \\ & & \circ & & & & \\ & & & \circ & & & \\ & & & & \circ & & \\ & & & & & \bar{R} & \\ \bar{R} & & & & & & \bar{R} \end{bmatrix}, \quad \text{where } \bar{R} = \begin{bmatrix} -2 & 1 & & & & & 1 \\ 1 & -2 & 1 & & & & \\ & & \circ & & & & \\ & & & \circ & & & \\ & & & & \circ & & \\ & & & & 1 & -2 & 1 \\ 1 & & & & & 1 & -2 \end{bmatrix}$$

(*nj x nj* block matrix) (*ni x ni* scalar matrix)

$$B_{yy} = \frac{a}{h_y^2} \begin{bmatrix} -2\bar{I} & \bar{I} & & & & & -2\bar{I} \\ \bar{I} & -2\bar{I} & \bar{I} & & & & \\ & & \circ & & & & \\ & & & \circ & & & \\ & & & & \circ & & \\ & & & & \bar{I} & -2\bar{I} & \bar{I} \\ -2\bar{I} & & & & & \bar{I} & -2\bar{I} \end{bmatrix}$$

(*nj x nj* block matrix)

$$P = \begin{bmatrix} \bar{P}_1 & \bar{P}_2 & \circ & \circ & \circ & \bar{P}_{nj} \end{bmatrix}^T, \qquad \text{where } \bar{P}_j = \begin{bmatrix} P_{1,j} & P_{2,j} & \circ & \circ & \circ & P_{ni,j} \end{bmatrix}^T$$

(*nj x 1* block vector) (*ni x 1* scalar vector)

$$F = \begin{bmatrix} \bar{F}_1 & \bar{F}_2 & \circ & \circ & \circ & \bar{F}_{nj} \end{bmatrix}^T, \qquad \text{where } \bar{F}_j = \begin{bmatrix} F_{1,j} & F_{2,j} & \circ & \circ & \circ & F_{ni,j} \end{bmatrix}^T$$

(*nj x 1* block vector) (*ni x 1* scalar vector)

where $P_{i,j}$ and $F_{i,j} = \delta_{x_i}[H_i + u_i/(b^{M+1}\Delta t)]_{i,j}$ are the pressure and source term at the i,j grid point, respectively. The symbols, $ni = N_x + 1$, $nj = N_y + 1$ denote the number of grid points in the x, y directions, respectively. Values at $i = N_x + 1$ and $j = N_y + 1$ locations are replaced by values at $i = 1$ and $j = 1$ since

the boundaries are periodic. This change results in a non-zero element in the upper right and lower left corners of the coefficient matrices. The sixth-order compact scheme (with pentadiagonal RHS) can be written in the same manner by including the $i\pm2$ and $j\pm2$ terms in the B_{xx} and B_{yy} matrices.

For non-periodic boundaries, the second derivative boundary scheme given by Eq. (21) is used at the boundary points. In addition, the unknown pressures at the boundaries are replaced with their boundary values and those terms are moved and added to the RHS of Eq. (26). For Dirichlet boundary conditions, such as freestream conditions, this procedure is straightforward. Neumann boundary conditions, such as those applied at inflow and outflow planes, require that the pressure gradient at the boundary be discretized using a first derivative scheme (Eq. 17):

$$(31) \qquad \left(\frac{\partial P}{\partial x_I}\right)_{I,j} = \frac{I}{\Delta x}\sum_{i=1}^{3} a_{bs_i} P_{i,j}$$

where the subscripts "1,j" are used to denote the inflow plane for example. The boundary pressure, $P_{1,j}$ is then solved for:

$$(32) \qquad P_{I,j} = \frac{I}{a_{bs_I}}\left[\Delta x\left(\frac{\partial P}{\partial x_I}\right)_{I,j} - \sum_{i=2}^{3} a_{bs_i} P_{i,j}\right]$$

Equation (32) is then used to substitute for the boundary pressures in Eq. (30). The first term on the RHS of Eq. (32) is known from the boundary condition and is moved and added to the RHS of Eq. (30). The second term on the RHS of Eq. (32) contains the unknown pressures, $P_{2,j}$ and $P_{3,j}$, so they are kept on the LHS of Eq. (30) and corresponding terms of the coefficient matrix, A are modified accordingly. The resulting system of equations contains only the interior pressures as unknowns, $P_{i,j}$, $2 \le i \le N_x - 1$ and $2 \le j \le N_y - 1$.

Equation (30) results in a "cross" type stencil at the i, j node in which all points along lines passing through the central node contribute to the stencil. The coefficients of this stencil are implicitly defined in the sense that the matrix operations, $[A_{xx}^{-1}B_{xx} + A_{yy}^{-1}B_{yy}]$, in Eq. (30) must be performed to determine their values.

Multiplying Eq. (30) by $A_{yy}A_{xx}$ gives:

$$(33) \qquad [A_{yy}B_{xx} + A_{yy}A_{xx}A_{yy}^{-1}B_{yy}]P = A_{yy}A_{xx}F$$

It is easy to show by inspection that matrices A_{xx} and A_{yy} commute, i.e. $A_{yy}A_{xx} = A_{xx}A_{yy}$, so that Eq. (33) simplifies to:

$$(34) \qquad [A_{yy}B_{xx} + A_{xx}B_{yy}]P = A_{yy}A_{xx}F$$

With the coefficients of the fourth-order compact approximation, the second derivative results in an explicit nine-point, "grid" type stencil for the LHS and RHS of Eq. (34):

$$(35) \quad \left[-2a\left(\frac{1}{h_x^2} + \frac{1}{h_y^2}\right)\right] P_{i,j} +$$

$$\left[-2a\left(\frac{\alpha}{h_x^2} - \frac{1}{2h_y^2}\right)\right][P_{i,j+1} + P_{i,j-1}] + \left[-2a\left(\frac{\alpha}{h_y^2} - \frac{1}{2h_x^2}\right)\right][P_{i+1,j} + P_{i-1,j}] +$$

$$\left[a\alpha\left(\frac{1}{h_x^2} + \frac{1}{h_y^2}\right)\right][P_{i+1,j+1} + P_{i+1,j-1} + P_{i-1,j+1} + P_{i-1,j-1}] = F_{i,j} +$$

$$\alpha[F_{i,j+1} + F_{i,j-1} + F_{i+1,j} + F_{i-1,j}] + \alpha^2[F_{i+1,j+1} + F_{i+1,j-1} + F_{i-1,j+1} + F_{i-1,j-1}]$$

With the coefficients of the sixth-order compact approximation, the second derivative results in the same nine-point stencil for the RHS and an explicit twenty-one point stencil on the LHS:

$$(36) \quad \left[-2\left(a + \frac{b}{4}\right)\left(\frac{1}{h_x^2} + \frac{1}{h_y^2}\right)\right] P_{i,j} +$$

$$\left[\frac{-2\alpha}{h_x^2}\left(a + \frac{b}{4}\right) + \frac{a}{h_y^2}\right][P_{i,j+1} + P_{i,j-1}] + \left[\frac{-2\alpha}{h_y^2}\left(a + \frac{b}{4}\right) + \frac{a}{h_x^2}\right][P_{i+1,j} + P_{i-1,j}]$$

$$\left[\frac{b}{4h_y^2}\right][P_{i,j+2} + P_{i,j-2}] + \left[\frac{b}{4h_x^2}\right][P_{i+2,j} + P_{i-2,j}] +$$

$$\left[\frac{b\alpha}{4h_x^2}\right][P_{i+2,j+1} + P_{i+2,j-1} + P_{i-2,j+1} + P_{i-2,j-1}] +$$

$$\left[\frac{b\alpha}{4h_y^2}\right][P_{i+1,j+2} + P_{i+1,j-2} + P_{i-1,j+2} + P_{i-1,j-2}] +$$

$$\left[a\alpha\left(\frac{1}{h_x^2} + \frac{1}{h_y^2}\right)\right][P_{i+1,j+1} + P_{i+1,j-1} + P_{i-1,j+1} + P_{i-1,j-1}] = F_{i,j} +$$

$$\alpha[F_{i,j+1} + F_{i,j-1} + F_{i+1,j} + F_{i-1,j}] + \alpha^2[F_{i+1,j+1} + F_{i+1,j-1} + F_{i-1,j+1} + F_{i-1,j-1}]$$

Thus for uniform cartesian grids, the stencils of Eqs. (35) and (36) are preferred because they require fewer operations and their coefficients are explicit in contrast to the stencil of Eq. (30). The commutative property of the A_{xx} and A_{yy} matrices is valid even with non-periodic boundaries. Numerical experiments confirm that the cross-type stencil represented by Eq. (30) and the grid-type stencil of Eq. (34) give identical results.

3.2. Iteration Matrix. The system of equations represented by Eq. (30) or Eq. (34) must be solved to convergence to ensure a divergence-free velocity field, at every sub-stage of every time step. Although the matrix is sparse, direct sparse-matrix methods were found to be competitive only for fairly coarse-grid 2-D problems Therefore, iterative schemes are required, and point-relaxation methods are utilized. In this approach, only the value at the central node of the stencil, $P_{i,j}$, is treated as an unknown so that the multi-diagonal system of equations degenerates to a diagonal system for one relaxation sweep, which is trivial to solve. This process can be written in matrix notation by decomposing the matrix, A, into the sum of the diagonal, lower, and upper matrices of A:

$$(37) \qquad AP = [D - L - U]P = F$$

where the matrix, D, is the diagonal matrix of A, and the matrices, L, U, are the lower and upper matrices of A, respectively. The solution at the current iteration level, P^*, is corrected with the increment, P', to yield the solution at the next iteration level, $P = P^* + P'$. For weighted Jacobi iteration the pressure is calculated from:

$$(38) \qquad P = P^* + \omega D^{-1} R^*$$

where $0 < \omega < 1$ denotes under-relaxation. Thus, Jacobi iteration is equivalent to computing the residual ($R^* = F - A\ P^*$) of the current iterate, P^*, at all grid points followed by an update operation. In this regard, information is held and the solution is updated at all grid points simultaneously. Since the computation of the residual vector and the update of the solution vector are completely separate operations, each operation is fully vectorizable. This results in an improved computational rate when those operations are performed on vector computers.

On the other-hand, weighted Gauss-Seidel calculates the pressure from:

$$(39) \qquad P = P^* + \omega[D - L]^{-1} R^*$$

which requires that the solution vector be updated in ascending order ($P_{1,1}, P_{2,1}, \cdots P_{ni,nj}$). Since this does not vectorize very well red-black ordering is a possibility. Although the Jacobi method converged more slowly than the Gauss-Seidel method, the computational speed increase due to vectorization led us to prefer it in the present study.

3.3. Multigrid Solution. Relaxation schemes such as Jacobi and Gauss-Seidel applied on a single grid level suffer from poor convergence rates as the number of grid points increase. Multigrid methods overcome these deficiencies by utilizing a hierarchy of grids. Smooth error components are transferred to coarser grids where they appear as high frequency error components and are quickly removed by relaxation sweeps. A coarse grid correction scheme is utilized in the current study to improve the convergence rate of the pointwise relaxation scheme on a single grid. Subscripts are used to denote grid level, i.e. P_h and P_{2h} denotes the solution on the fine and coarse grids, respectively. The symbol, I_h^{2h}, is used to denote transfer from the fine to the coarse grid, while I_{2h}^{h} is used to denote transfer in the opposite direction. The algorithm for one coarse grid correction is given below and additional details can be found in Briggs [16].

(1) Smooth the current iterate, P_h^*, on the fine grid ν_1 times:

$$A_h P_h^{(1)} = F_h$$

(2) Calculate the residual on the fine grid:

$$R_h = F_h - A_h P_h^{(1)}$$

(3) Transfer (restrict) the residual to the coarse grid, where it is used as the source term for the error equation:

$$F_{2h} = I_h^{2h} R_h$$

(4) Solve for the error on the coarse grid:

$$E_{2h} = A_{2h}^{-1} F_{2h}$$

(5) Transfer (prolongate) the error to the fine grid and correct the solution:

$$P_h^{(2)} = P_h^{(1)} + I_{2h}^h E_{2h}$$

(6) Perform ν_2 post-relaxation sweeps:

$$A_h P_h = F_h$$

Standard second-order interpolation is used to transfer variables from the coarse to the fine grid, while the full weighting operator [16] is used to transfer variables in the opposite direction. Although the above algorithm utilizes only two grid levels, improved efficiency results from incorporating as many grid levels as possible. In this respect, the direct solution of the error equation in step (5) is performed on a very coarse grid requiring a small number of operations.

Since the simulations are performed on vector computers, Jacobi iteration was utilized for relaxation sweeps because it is fully vectorizable. Two pre- and two post-relaxation sweeps were performed on each grid. Through numerical experiments, the optimum relaxation factor for the uniform grid formulation was found to be, $\omega = 0.9$.

Convergence rates (CR) for the 2D wave decay problem (discussed in Sect. 4) using the 4th- and 6th-order schemes are given in Table 8 along with the number of multigrid levels used. The results demonstrate that the convergence rate of the multigrid solution technique is independent of grid spacing unlike typical single grid, iterative solution techniques where the convergence rate approaches unity as the grid is refined.

Effects of grid aspect ratios on multigrid convergence rates are shown in Table 9 for the solution of the 2D Stuart's problem using the fourth-order compact scheme. The 2D Stuarts problem, which is defined in Sect. 4, is a temporally developing mixing layer with neutral growth, which is governed by the 2D Euler equations, and for which an exact solution exists. Uniform grid spacing is used in the x coordinate direction, while a logarithmic function is used to cluster grid points in the y coordinate direction where large gradients exist due to the mixing of the high and low speed freestreams (see Figure 10):

$$(40) \qquad\qquad y = ln\left(\frac{1+\eta}{1-\eta}\right)/2c$$

where c is a clustering parameter and η is the uniformly distributed computational coordinate. The results in Table 9 show that the multigrid convergence rate deteriorates somewhat with increased grid aspect ratio.

As more grid points are moved to the mixing region through clustering, solution errors in the y coordinate direction decrease while those in x direction remain constant. Thus, the overall solution error eventually saturates with increased resolution in the y direction

.

Table 8: Multigrid convergence rates, CR, for solution of 2D wave decay.

ni x nj	number of MG levels	CR - 4th order scheme	CR - 6th order scheme
16 x 16	2	0.41	0.30
32 x 32	3	0.40	0.30
64 x 64	4	0.41	0.30
128 x 128	5	0.42	0.30

Table 9. Multigrid convergence rates, CR, for solution of 2D Stuarts problem on non-uniform grids (ni x nj = 49 x 161) at t=0.2.

Clustering parameter, c	$r = \dfrac{\Delta y_{max}}{\Delta y_{min}}$	CR	$|e_u|_2$
uniform grid	1	0.19	0.1319×10^{-5}
0.12	3.2	0.21	0.7860×10^{-6}
0.16	6.2	0.28	0.6402×10^{-6}
0.20	12.2	0.38	0.5725×10^{-6}
0.24	24.2	0.48	0.5675×10^{-6}

4. Results and discussion. The performance of the numerical formulation is tested by application to a variety of benchmark problems. Emphasis is placed on the numerical approximation of spatial derivatives. In particular, the convection terms (containing first derivatives) present the most difficulty in numerical approximation since large dispersion errors exist at high wavenumbers ($k\Delta x \sim \pi$). It is essential that the numerical approximation to the first derivative provide low dispersion errors over a large range of wavenumbers. This is especially true in 3-D simulations where reducing the required number of grid points by half in each coordinate direction leads to eight times fewer total grid points, and considerable savings in computer time

In Section 3, theory showed that compact schemes require roughly five times fewer points to accurately resolve a given mode in comparison to the standard second-order central difference approximation to the first derivative. This theory is tested by solving some practical problems ranging from the 1-D convection equation to the 2-D Navier-Stokes equations.

4.1. 1-D Convection Equation. The first problem to be solved is the 1-D convection of a Gaussian profile:

$$(41) \qquad\qquad \frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0$$

subject to: $u(x, 0) = 0.5exp\left[-\left(\frac{x}{3}\right)^2 ln(2)\right]$; $-20 \le x \le 450$; $\Delta x = 1$, $c = 1$.

This could test the time advancement scheme and the numerical approximation to the first derivative. The exact solution corresponds to the convection of the initial profile at the constant wave speed, c. The third-order RK scheme was used to advance the equation in time for all spatial schemes. In addition, the CFL number was kept small so that resulting errors are due solely to the spatial formulation. Distortion in the shape of the profile indicates dissipation and/or dispersion errors in the solution. The convection equation was solved using three approximations to the first derivative; (i) a second-order central difference, (ii) a third-order upwind, and (iii) the fourth-order compact approximation outlined in Section 2.3. The parameters and initial conditions are those proposed at the ICASE/LARC Workshop on Benchmark Problems in Computational Aeroacoustics, Hardin et al. [17]. Since the specified grid is relatively coarse, this problem provides an excellent test of the resolution power of the numerical approximation. Figure 4 shows the computed solutions at t = 400 after the profile has convected to x = 400. There is little discernible difference between the exact solution and the solution with the fourth-order compact scheme. However, the solutions with the second-order central difference and the third-order upwind approximations show greatly reduced peak values and large, dispersive waves trailing the Gaussian profile. The errors from the second-order central difference scheme are the most severe.

It is difficult to determine by inspection what portion of the error is dispersive and what portion is dissipative. The solutions are transformed into wavenumber space using a Fourier transform method and compared with the exact solution in Fig. 5 to address this issue. The graph displays the resulting complex Fourier coefficient in polar form with the amplitude displayed in Fig. 5a and the phase angle in Fig. 5b. It can be seen from Fig. 5a that the solutions computed with the second-order central difference and fourth-order compact schemes predict the correct amplitude for all modes. The amplitude of the solution computed with the third-order upwind scheme is reduced or dissipated, especially at higher wavenumbers. Figure 5b shows that the fourth-order compact scheme predicts the correct phase angle even for the highest wavenumbers.

The phase angle from the second- and third-order solutions are only correctly predicted for the very lowest wavenumbers ($k < 0.2$ for the second-order solution and $k < 0.3$ for the third-order solution). Large dispersion errors are evident at high wavenumbers. The above trends in the numerical solutions are consistent with the dissipation/dispersion error theory for the 1-D convection equation and show the resolution power of the compact schemes.

The second problem, also proposed at the ICASE/LARC Workshop on Benchmark Problems, is the solution of the 1-D convection equation in a spherical coordinate system. In Cartesian coordinates, the governing equation takes the form:

$$(42) \qquad \frac{\partial u}{\partial t} + \frac{u}{x} + \frac{\partial u}{\partial x} = 0$$

subject to: $u(x, 0) = 0$; $u(5, t) = \sin\left(\frac{\pi t}{4}\right)$; $5 \le x \le 450$; $\Delta x = 1$

Figure 6 shows the exact solution at $t = 400$ which corresponds to a damped sine wave due to the addition of the u/x term in the governing equation. Fig. 7 shows computational results for the region, $200 < x < 220$ using the third-order upwind approximation to the first derivative on three different grids and the fourth-order compact scheme, on the specified grid only, which corresponds to 8 PPW. The upwind solution with 8 PPW shows severely reduced amplitude and a phase shift relative to the exact solution. Even those with 16 PPW and 32 PPW are not very accurate. It takes roughly 64 PPW (not shown) to reproduce the exact solution with the third-order upwind approximation. On the other hand, the fourth-order compact approximation is able to reproduce the exact solution with 8 PPW.

4.2. 2-D Convection Equation. Multidimensional effects of the numerical formulation are explored by solving for the convection of an inverted cone around a circle. This problem is governed 2-D convection equation:

$$(43) \qquad \frac{\partial u}{\partial t} + c_x \frac{\partial u}{\partial x} + c_y \frac{\partial u}{\partial y} = 0$$

where $c_x = -y$ and $c_y = x$, are the convection speeds in the x and y directions, respectively. The initial conditions are that of an inverted sharp cone centered at $x, y = -0.5, 0$. The exact solution corresponds to the cone being convected counterclockwise in a circular path of radius, $r_0 = 0.5$ with a period of 2π. Distortion of the shape of the cone is an indication of dispersion and/or dissipation errors.

Figure 8 shows computed results after one revolution of the cone using (a) a third-order upwind approximation and (b) a fourth-order compact approximation to the first derivatives on a 32 x 32 grid with uniform spacing. This grid defines the shape of the cone with a maximum of 8 points in each coordinate direction. The exact shape of the cone is included to the right of the computed solution at x, y = 0.5, 0 for comparison purposes. The third-order solution (Fig 8a) shows that the sharp point of the cone is greatly diffused and that dispersion errors are evident trailing the cone. A grid of 128 x 128 (or 32 points defining the shape of the cone) must be used with the third-order upwind approximation before the shape of the cone is faithfully reproduced. The fourth-order compact solution (Fig 8b) shows that the shape of the cone is not distorted as it is convected around the circle on the 32 x 32 grid. Indeed, the only noticeable error is a very small "grid to grid" oscillation due to the absence of physical viscosity in this problem and numerical viscosity in the compact scheme.

Figure 9 shows results for the same problem after one revolution obtained by Orszag [18] using (a) second-order Arakawa finite-difference, (b) fourth-order Arakawa finite-difference, and (c) a spectral approximation to the first derivatives on a 32 x 32 grid. The finite difference solutions show errors similar to the

third-order solutions in Fig. 8. The spectral method, which provides exact differentiation for all wavenumbers representable on the 32 x 32 grid, convects the cone without distorting its shape. Thus, the solution using the compact scheme is closer to the spectral solution than the solutions obtained with conventional finite difference schemes. The higher accuracy and resolution characteristics are achieved by the implicit treatment of the derivative. Even though the stencil size of the compact scheme is finite, the implicit treatment of the derivatives makes the scheme global much like that of spectral methods.

4.3. Euler/Navier-Stokes Equations.

In the previous sections, the effect of numerical approximation on the accuracy of the convection terms was documented. In this section, the accuracy of the enforcement of the continuity equation through the solution of the Poisson equation for pressure is documented by solving the 2-D Euler/Navier-Stokes equations. Since the Navier-Stokes equations contain viscous terms, the numerical approximation to the second derivative is also tested. The test problems chosen for validation contain many features of the 3-D jets which are simulated in the current study. In this respect, the test problems are not merely academic exercises. There benchmark problems are solved; (i) a temporally-developing plane mixing layer (2-D Stuarts problem), and (ii) 2-D viscous wave decay. Problems (i) and (ii) have exact solutions.

4.3.1. Temporally-Developing Plane Mixing Layer.

Exact solutions to the Euler or Navier-Stokes equations for general flows do not exist due to the non-linearity of the convection terms. However, under special conditions exact solutions may be found. An exact solution for the temporally-developing mixing layer was first published by Stuart [19]. The initial conditions for the 2-D Stuarts problem correspond to a steady hyperbolic tangent function for the streamwise velocity component with a periodic array of vortex cores in the mixing region which cause the solution to vary in time. The wavelength of the disturbance corresponds to the neutral mode such that the disturbance is convected in the streamwise direction with no change in amplitude. The exact solution for the streamwise velocity component, u, and the transverse velocity component, v is given by:

$$u(x, y, t) = c + \frac{C \sinh(y)}{C \cosh(y) + A \cos(x - ct)}$$

$$(44) \qquad v(x, y, t) = \frac{A \sin(x - ct)}{C \cosh(y) + A \cos(x - ct)}$$

where $A = \sqrt{C^2 - 1}$ is a parameter which controls the strength of the perturbation and c is the convective speed of the mixing layer. The flow is periodic in the streamwise direction with length, $L_x = 2\pi$, $0 \le x \le 2\pi$. The flow is infinite in the transverse direction but in this study is truncated at a finite distance, $-L_y \le y \le L_y$, such that a zero-traction freestream boundary condition is well approximated. Tests which vary the transverse domain height, $2L_y$, show that $L_y = 10$ is sufficiently large to implement this boundary condition. The exact solution is shown in Fig. 10a with parameters, $c = 1$, $A = 1/2$. A uniform, cartesian grid is used for the simulations in this section. Unless otherwise specified, the third-order RK scheme is used for time advancement and time steps are sufficiently small so that spatial errors are dominant.

Figure 10b shows the numerical solution at $t = 20\pi$ (ten flow through times) on a relatively coarse grid of 13 (streamwise) x 41 (transverse) using the fourth-order compact approximation of convection terms and pressure. The solution of pressure involves the computation of the source term and the discretization of the Laplacian operator in Eq. (33). In addition, once the Poisson equation is solved for pressure, the gradient of pressure is computed which is required to advance the momentum equation in time. Therefore, the phrase "fourth-order solution of pressure" corresponds to the source and pressure gradient terms computed with the compact first derivative scheme outlined in Section 2.3.1 while the Laplacian operator is discretized using the compact second derivative scheme outlined in Section 2.3.2.

Even though the grid is relatively coarse (13 streamwise points per wavelength and roughly 8 points in the mixing region at $y \sim 0$), there is little discernible difference between the exact and numerical solutions after ten flow through times. It is important to check the convergence of the error as the grid is refined to expose any coding errors, to demonstrate that the order of error convergence seen in practical computations is that predicted by a Taylor series analysis, and to gain confidence in the numerical formulation. Tables 10 and 11 give a quantitative measure of the L2 and maximum errors in the velocity components at t = 0.1 using the fourth- and sixth-order compact approximations to the convection terms and solution of pressure, respectively. Solution errors from three grids are shown where the grid spacing in the x and y directions is halved from coarsest to finest grid. The results show that the L2 and maximum errors converge at roughly the rate predicted by a Taylor series analysis as the grid is refined. The order, N, is computed using the solution error from three grids of spacing, h, 2h, and 4h:

$$(45) \qquad N = \frac{ln\left[\dfrac{\phi_h - \phi_{2h}}{\phi_{2h} - \phi_{4h}}\right]}{ln2}$$

where $\phi_h, \phi_{2h}, \phi_{4h}$ are the errors on the h, $2h$, and $4h$ grids, respectively. In using Eq. (45) it is assumed that the solution is fully resolved on all three grids and that the leading truncation error term is dominant (Demuren and Wilson [20]).

Table 10. Solution errors for 2-D Stuarts Problem at t = 0.1 using fourth-order compact approximation for convection terms and solution of pressure.

Grid (ni x nj)	Max U Error	Max V Error	L2 Norm U	L2 Norm V
13 x 41	0.18×10^{-2}	0.21×10^{-2}	0.18×10^{-3}	0.24×10^{-3}
25 x 81	0.86×10^{-4}	0.12×10^{-3}	0.80×10^{-5}	0.11×10^{-4}
49 x 161	0.47×10^{-5}	0.68×10^{-5}	0.57×10^{-6}	0.74×10^{-6}
Order, N	4.2	4.1	4.5	4.5

To address the effect of computing the pressure with a lower-order formulation, the 2-D Stuarts problem was solved using second-order central, fourth-order compact and sixth-order compact approximation of the convection terms but a second-order central difference solution of the pressure. The results of the three computations are shown in Table 12. These results show that the lower-order solution of pressure results in the overall convergence of the error being second-order, even if the convection terms receive a higher-order treatment. All terms must be discretized using higher-order approximations to achieve higher-order error

Table 11. Solution errors for 2-D Stuarts Problem at t = 0.1 using sixth-order compact approximation for convection terms and solution of pressure.

Grid(ni x nj)	Max U Error	Max V Error	L2 Norm U	L2 Norm V
13 x 41	0.73×10^{-3}	0.10×10^{-2}	0.97×10^{-4}	0.11×10^{-3}
25 x 81	0.17×10^{-4}	0.20×10^{-4}	0.12×10^{-5}	0.14×10^{-5}
49 x 161	0.15×10^{-5}	0.18×10^{-5}	0.25×10^{-6}	0.45×10^{-6}
Order, N	5.4	5.7	6.3	6.2

convergence rates. Thus, formulations presented in the literature, such as by Najjar and Tafti [21], which use higher-order differences for the convection and diffusion terms but second-order differences for the Poisson equation for pressure would only be globally second-order accurate.

Table 12. Errors with different discretization approximations for convection, but second-order discretization for pressure.

	Second-order central		Fourth-order compact		Sixth-order compact	
Grid (ni x ni)	L2 Norm U	L2 Norm V	L2 Norm U	L2 Norm V	L2 Norm U	L2 Norm V
13 x 41	0.21×10^{-2}	0.20×10^{-2}	0.20×10^{-2}	0.20×10^{-2}	0.20×10^{-2}	0.20×10^{-2}
25 x 81	0.45×10^{-3}	0.53×10^{-3}	0.42×10^{-3}	0.44×10^{-3}	0.42×10^{-3}	0.44×10^{-3}
49 x 161	0.11×10^{-3}	0.13×10^{-3}	0.11×10^{-3}	0.11×10^{-3}	0.10×10^{-3}	0.11×10^{-3}
Order (N)	2.3	2.1	2.4	2.2	2.3	2.2

.The solution of the 2-D Stuarts problem validates the numerical formulation for the enforcement of the continuity equation and the solution of the Poisson equation for pressure. In addition, it has been shown that the zero-traction freestream boundary condition for shear flows is a valid approximation provided that the freestream boundary is located a sufficiently far distance from the mixing region.

Temporal accuracy of the overall RK schemes was confirmed by performing computations for the 2-D Stuarts problem on the 49 by 161 grid and an even finer 97 by 161 grid, with several different time steps. The largest time step in each case was chosen from stability considerations. The results are shown in Tables 13 - 15. In Tables 13 and 14, the velocity field is specified from the exact solution and the vorticity is computed as a passive scalar, using the sixth-order compact scheme. In each case, the error shown contains both temporal and spatial discretization errors. As the time step is reduced to very small values the latter become dominant and error reduction stagnate as seen in column 5 of Table 13. On this grid, the spatial discretization error can be estimated to be about 0.6×10^{-6}. On the 97 by 161 grid, the spatial discretization error should reduce by about 2^6 or 64, to roughly 0.9×10^{-9}. Column 5 of Table 14 does show stagnation at about this value, which confirms the analysis. Equation (45) enables separation of temporal and spatial discretization errors by comparison of computed results from three time steps on the same grid, or three grids with the same time step. This procedure shows (column 4) that the 3-3 RK scheme is indeed third-order accurate, and the 5-4 RK scheme (column 7) is somewhat better than fourth-order accurate. The use of computed results from only two time steps (columns 3 and 6) is clearly erroneous and produces unreliable conclusions.

Table 13. Solution errors for 2-D Stuarts Problem at t = 1.0, with different RK schemes on a 49 by 161 grid.

Time Step	3-3 RK scheme			5-4 RK scheme		
	L2 Norm of Vorticity Error	N from 2 grids	N from 3 grids	L2 Norm of Vorticity Error	N from 2 grids	N from 3 grids
0.10				0.32980×10^{-5}		
0.05	0.12803×10^{-4}			0.69910×10^{-6}	2.24	
0.025	0.17157×10^{-5}	2.9		0.63575×10^{-6}	0.14	5.4
0.0125	0.66280×10^{-6}	1.4	3.4	0.63328×10^{-6}	0.01	4.7

Table 14. Solution errors for 2-D Stuarts Problem at t = 1.0, with different RK schemes on a 97 by 161 grid.

Time Step	3-3 RK scheme			5-4 RK scheme		
	L2 Norm of Vorticity Error	N from 2 grids	N from 3 grids	L2 Norm of Vorticity Error	N from 2 grids	N from 3 grids
0.05				0.1965×10^{-6}		
0.025	0.1613×10^{-5}			0.1683×10^{-7}	3.5	
0.0125	0.2019×10^{-6}	3.0		0.9496×10^{-8}	0.83	4.6
0.00625	0.2685×10^{-7}	2.9	3.0	0.9310×10^{-8}	0.02	5.3

In Table 15, the full set of Euler equations is solved with the fourth-order compact scheme and the 3-3 RK scheme. Third-order temporal accuracy is confirmed. In order to achieve this, it was necessary to obtain a divergence-free velocity field at every sub-stage of the time-stepping process.

Table 15. Solution errors for 2-D Stuarts Problem at t = 1.0 using third-order RK and fourth-order compact approximation for Euler Equations.

Grid (ni x nj)	Time Step	L2 Norm U	L2 Norm V
49 x 161	0.05	0.51630×10^{-3}	0.25821×10^{-3}
49 x 161	0.025	0.27651×10^{-5}	0.18778×10^{-4}
49 x 161	0.0125	0.24666×10^{-6}	0.17823×10^{-6}
Order, N		3.0	2.9

4.3.2. Viscous Wave Decay. The numerical treatment of viscous terms is validated by solving the 2-D viscous wave decay problem which is governed by the Navier-Stokes equations. The domain for this

problem is periodic in both the x and y directions where periodic boundary conditions are applied. The exact solution is given by:

$$(46) \qquad u(x, y, t) = -cos(x)sin(y)e^{-\left(\frac{2t}{Re}\right)}$$

$$v(x, y, t) = sin(x)cos(y)e^{-\left(\frac{2t}{Re}\right)}$$

where $Re = 20$, $L_x = L_y = 1$, and $t = 0$ gives the initial conditions. The exact solution consists of sinusoidal waves in the x and y directions which decay in time. Table 16 shows the L2 norm of the error at $t = 0.025$ using the fourth- and sixth-order compact approximations for convection and diffusion terms and the solution of pressure. The results are compared to the fourth-order, Essentially Non-Oscillatory (ENO) scheme from Weinan and Shu[22]. The error converges at fourth- and sixth-order rates thus validating the numerical treatment of the viscous terms and again validating the convection terms and the solution of pressure. The error of the ENO scheme converges at a fourth-order rate, but is more than two orders of magnitude greater than the fourth-order compact results. The error magnitude of the sixth-order compact formulation on the 128 x 128 grid has reached the round-off error level ($\sim 10^{-13}$) of the Cray supercomputer, indicating that extremely accurate results are obtained on average-sized grids.

Table 16. Solution errors for 2-D viscous wave decay.

Grid (ni x nj)	4th oa compact	6th oa compact	3rd(4th) oa ENO
16 x 16	0.14×10^{-6}	0.10×10^{-7}	-
32 x 32	0.77×10^{-8}	0.15×10^{-9}	0.53×10^{-5}
64 x 64	0.47×10^{-9}	0.27×10^{-11}	0.32×10^{-6}
128 x 128	0.71×10^{-10}	0.11×10^{-12}	0.20×10^{-7}
Order, N	4.0	6.0	4.0

REFERENCES

[1] S. K. LELE, Compact Finite Difference Schemes with Spectral Like Resolution, Journal of Computational Physics, Vol. 103, pp. 16-42, 1992.

[2] Z. HARAS AND S. TAASAN, Finite-Difference Schemes for Long-Time Integration, Journal of Computational Physics, Vol. 114, pp. 265-279, 1994.

[3] F. Q. HU, M. Y. HUSSAINI, AND J. MANTHEY, Low-Dissipation and -Dispersion Runge-Kutta Schemes for Computational Acoustics, ICASE Report 94-102, and Journal of Computational Physics, Vol. 124, 1, pp.177-191, 1996

[4] K. AKSELVOLL AND P. MOIN, Large Eddy Simulation of Turbulent Confined Coannular Jets and Turbulent Flow over a Backward Facing Step, Report TF-63, Department of Mechanical Engineering, Stanford University, 1995.

[5] W. RODI, J. H. FERZIGER, M. BREUER, AND M. POURQUIE, Status of Large Eddy Simulation: Results of

a Workshop, ASME, Journal of Fluids Engineering, Vol. 119, 2, pp. 248-262, 1997.

[6] C. K. W. TAM AND J. C. WEBB, *Dispersion-Relation-Preserving Schemes for Computational Acoustics,* Journal of Computational Physics, Vol. 103, pp. 262-281, 1993.

[7] G. ERLEBACHER, M.Y. HUSSAINI, AND C.-W. SHU, *Interaction of a Shock with a Longitudinal Vortex,* Journal of Fluid Mechanics, Vol. 337, 25, pp. 129-154, 1997.

[8] J. B. BELL, P. COLELLA AND H. M. GLAZ, *A Second-Order Projection Method for Incompressible Navier-Stokes Equations,* Journal of Computational Physics, 85, pp. 257-283, 1989.

[9] W. D. HENSHAW, *A Fourth-Order Accurate Method for the Incompressible Navier-Stokes Equations on Overlapping Grids,* Journal of Computational Physics, Vol. 113, pp. 13-25, 1994.

[10] R. D. JOSLIN, C. L. STREETT AND C-L. CHANG, *Validation of Three-Dimensional Incompressible Spatial Direct Numerical Simulation Code,* NASA Technical Paper 3205, 1992.

[11] R. K. MADABHUSHI, S. BALACHANDRA, AND S. P. VANKA, *A Divergence-Free Chebyshev Collocation Procedure for Incompressible Flows with Two Non-Periodic Directions,* Journal of Computational Physics, Vol 105, 2, pp. 199-206, 1994.

[12] J. WILLIAMSON, *Low Storage Runge-Kutta Schemes,* Journal of Computational Physics, Vol. 35, pp. 48-56, 1980.

[13] S.L. LOWERY, AND W. C. REYNOLDS *Numerical Simulation of a Spatially-Developing Mixing Layer,* Report TF-26, Department of Mechanical Engineering, Stanford University, 1986.

[14] M. H. CARPENTER AND C.A. KENNEDY, *Fourth-Order, 2N-Storage Runge-Kutta Schemes,* NASA TM 109112, 1994.

[15] M.H. CARPENTER, D. GOTTLIEB AND S. ABARBANEL, *The Stability of Numerical Boundary Treatments for Compact High-Order Finite Difference Schemes,* Journal of Computational Physics, Vol. 180, 2, 1991.

[16] W. L. BRIGGS, *A Multigrid Tutorial,* Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, 1987.

[17] J. C. HARDIN, J. R., RISTORCELLI, AND C. K. W. TAM, *ICASE/LARC Workshop on Benchmark Problems in Computational Aeroacoustics CAA,* 1995.

[18] S.A. ORSZAG, *Numerical Simulation of Incompressible Flows within Simple Boundaries: Accuracy,* Journal of Fluid Mechanics, Vol. 49, Part 1, pp. 75-112, 1971.

[19] J.T. STUART, *On Finite Amplitude Oscillations in Laminar Mixing Layers,* Journal of Fluid Mechanics, Vol. 29, Part 3, pp. 417-440, 1967.

[20] A.O. DEMUREN, AND R.V. WILSON, *Estimating Uncertainty in Computations of Two-Dimensional Separated Flows,* Journal of Fluids Engineering, Vol. 116, pp. 216-220, 1994.

[21] F.M. NAJJAR, AND D.K. TAFTI, *Study of Discrete Test Filters and Finite Difference Approximations for the Dynamic Subgrid-Scale Stress Model,* Physics of Fluids, Vol. 8, No. 4, pp. 1076-1088, 1996.

[22] E. WEINAN, AND C.W. SHU, *A Numerical Resolution Study of High Order Essentially Non-Oscillatory Schemes Applied to Incompressible Flow,* ICASE Report 92-39, 1992.

APPENDIX A - Discretization of 3D Poisson equation. In Sect. 3, the solution of the 2D Poisson equation for pressure using fourth-order compact finite differencing was developed and a reduced, nine-point, grid-type stencil was presented. In this appendix, the details of the solution procedure for the 3D Poisson equation on uniform grids are presented. The Poisson equation for pressure was given by Eq. (26):

$$\text{(A.1)} \qquad \delta_{xx_i} P^M = \delta_{x_i}\left[H_i^M + \frac{u_i^M}{b^{M+1}\Delta t} \right]$$

Compact finite differencing is used for the discrete derivative operators in Eq. (A.1). For the 3D Poisson equation on uniform grids, Eq. (A.1) can be written in the form of a system of equations as:

$$\text{(A.2)} \qquad \tilde{A}P = [\tilde{A}_{xx}^{-1}\tilde{B}_{xx} + \tilde{A}_{yy}^{-1}\tilde{B}_{yy} + \tilde{A}_{zz}^{-1}\tilde{B}_{zz}]\tilde{P} = \tilde{F}$$

where the tilde is used to denote 3D block matrices which are defined using the 2D block matrices given in Sect. 3.

$$\tilde{A}_{xx} = \begin{bmatrix} A_{xx} & & & & & A_{xx} \\ & A_{xx} & & & & \\ & & \circ & & & \\ & & & \circ & & \\ & & & & \circ & \\ & & & & & A_{xx} \\ A_{xx} & & & & & A_{xx} \end{bmatrix}, \quad \tilde{A}_{yy} = \begin{bmatrix} A_{yy} & & & & & A_{yy} \\ & A_{yy} & & & & \\ & & \circ & & & \\ & & & \circ & & \\ & & & & \circ & \\ & & & & & A_{yy} \\ A_{yy} & & & & & A_{yy} \end{bmatrix},$$

$$\tilde{A}_{zz} = \begin{bmatrix} \bar{I} & \alpha\bar{I} & & & & \alpha\bar{I} \\ \alpha\bar{I} & \bar{I} & \alpha\bar{I} & & & \\ & & \circ & & & \\ & & & \circ & & \\ & & & & \circ & \\ & & & \alpha\bar{I} & \bar{I} & \alpha\bar{I} \\ \alpha\bar{I} & & & & \alpha\bar{I} & \bar{I} \end{bmatrix}$$

$$\tilde{B}_{xx} = \begin{bmatrix} B_{xx} & & & & & B_{xx} \\ & B_{xx} & & & & \\ & & \circ & & & \\ & & & \circ & & \\ & & & & \circ & \\ & & & & & B_{xx} \\ B_{xx} & & & & & B_{xx} \end{bmatrix}, \quad \tilde{B}_{yy} = \begin{bmatrix} B_{yy} & & & & & B_{yy} \\ & B_{yy} & & & & \\ & & \circ & & & \\ & & & \circ & & \\ & & & & \circ & \\ & & & & & B_{yy} \\ B_{yy} & & & & & B_{yy} \end{bmatrix}, ,$$

28

$$\tilde{B}_{zz} = \frac{a}{h_z^2} \begin{bmatrix} -2\tilde{I} & \tilde{I} & & & & & -2\tilde{I} \\ \tilde{I} & -2\tilde{I} & \tilde{I} & & & & \\ & & \circ & & & & \\ & & & \circ & & & \\ & & & & \circ & & \\ & & & & \tilde{I} & -2\tilde{I} & \tilde{I} \\ -2\tilde{I} & & & & & \tilde{I} & -2\tilde{I} \end{bmatrix}$$

where the above are nk x nk block matrices, nk is equal to the number of grid points in the z direction, and I is the $(ni$ x $nj)$ x $(ni$ x $nj)$ identity matrix. The 2D block matrices A_{xx}, A_{yy}, B_{xx}, and B_{yy} were defined in Sect. 3. The block vectors, $\tilde{P}$ and $\tilde{F}$ are assembled from the 2D vectors, P and F, defined in Sect. 3, $\tilde{P} = \begin{bmatrix} P_1 & P_2 & \circ & \circ & \circ & P_{nk} \end{bmatrix}^T$, $\tilde{F} = \begin{bmatrix} F_1 & F_2 & \circ & \circ & \circ & F_{nk} \end{bmatrix}^T$. The sixth-order compact scheme (with pentadiagonal RHS) can be written in the same manner by including the $i\pm2$, $j\pm2$, and $k\pm2$ terms in the above matrices. The coefficients of this stencil are implicitly defined in the sense that the matrix operation,

$[\tilde{A}_{xx}^{-1}\tilde{B}_{xx} + \tilde{A}_{yy}^{-1}\tilde{B}_{yy} + \tilde{A}_{zz}^{-1}\tilde{B}_{zz}]$, must be performed to determine their values. A more compact, "grid" type stencil is defined by multiplying Eq. (A.2) by $\tilde{A}_{zz}\tilde{A}_{yy}\tilde{A}_{xx}$:

(A.3) $\qquad\qquad [\tilde{A}_{zz}\tilde{A}_{yy}\tilde{B}_{xx} + \tilde{A}_{zz}\tilde{A}_{xx}\tilde{B}_{yy} + \tilde{A}_{yy}\tilde{A}_{xx}\tilde{B}_{zz}]\tilde{P} = \tilde{A}_{zz}\tilde{A}_{yy}\tilde{A}_{xx}\tilde{F}$

where the commutation properties, $A_{yy}A_{xx} = A_{xx}A_{yy}$, $A_{zz}A_{xx} = A_{xx}A_{zz}$, and $A_{zz}A_{yy} = A_{yy}A_{zz}$ have been used. Using tridiagonal compact finite differencing for the matrices in Eq. (A.3) results in an explicit 27-point stencil for the LHS and RHS of Eq. (A.3). Because of the symmetry of the second derivative, there are only 8 unique coefficients in the RHS and LHS of 27-point stencil which are given in Table A.1. The stencil represented by Eq. (A.3) does not require a separate near boundary scheme for Dirichlet boundary conditions since the stencil is three points wide.

Table A.1 Coefficients of the LHS 27-point explicit stencil for 3D Poisson equation using fourth-order compact scheme.

Location	Coeff.s of LHS	Coeff.s of RHS
i,j,k	$-2a(1/h_x^2 + 1/h_y^2 + 1/h_z^2)$	1
$i \pm 1, j, k$	$a(1/h_x^2 - 2\alpha/h_y^2 - 2\alpha/h_z^2)$	α
$i, j \pm 1, k$	$a(-2\alpha/h_x^2 + 1/h_y^2 - 2\alpha/h_z^2)$	α
$i, j, k \pm 1$	$a(-2\alpha/h_x^2 - 2\alpha/h_y^2 + 1/h_z^2)$	α
$i \pm 1, j \pm 1, k$	$a\alpha(1/h_x^2 + 1/h_y^2 - 2\alpha/h_z^2)$	α^2
$i \pm 1, j, k \pm 1$	$a\alpha(1/h_x^2 - 2\alpha/h_y^2 + 1/h_z^2)$	α^2
$i, j \pm 1, k \pm 1$	$a\alpha(-2\alpha/h_x^2 + 1/h_y^2 + 1/h_z^2)$	α^2
$i \pm 1, j \pm 1, k \pm 1$	$a\alpha^2(1/h_x^2 + 1/h_y^2 + 1/h_z^2)$	α^3

APPENDIX B - Solution procedure for curvilinear grids. The procedure for solution of the governing equations on general grids using compact finite differencing is outlined in the this appendix. The approach taken is to transform the spatial gradients from physical space to a computational space with uniform grid spacing where higher-order, compact finite differencing is used. The velocity components are not transformed and are defined in the Cartesian coordinate system.

First derivatives in the Cartesian coordinate system are expressed in terms of derivatives in the computational coordinate system using the chain rule:

(B.1)
$$\frac{\partial \phi}{\partial x_j} = \frac{\partial \varepsilon_m}{\partial x_j}\frac{\partial \phi}{\partial \varepsilon_m}$$

The Laplacian operator in the Cartesian coordinate system is expressed in terms of gradients in the computational space:

(B.2)
$$\frac{\partial \phi}{\partial x_j \partial x_j} = \frac{\partial \varepsilon_m}{\partial x_j}\frac{\partial}{\partial \varepsilon_m}\left(\frac{\partial \varepsilon_n}{\partial x_j}\frac{\partial \phi}{\partial \varepsilon_n}\right)$$

$$= m_{11}\frac{\partial^2 \phi}{\partial \varepsilon_1 \partial \varepsilon_1} + m_{22}\frac{\partial^2 \phi}{\partial \varepsilon_2 \partial \varepsilon_2} + m_{12}\frac{\partial^2 \phi}{\partial \varepsilon_1 \partial \varepsilon_2} + m_1\frac{\partial \phi}{\partial \varepsilon_1} + m_2\frac{\partial \phi}{\partial \varepsilon_2}$$

where: $m_{11} = J^2(x_\eta^2 + y_\eta^2)$, $m_{22} = J^2(x_\varepsilon^2 + y_\varepsilon^2)$

, $m_{12} = -2J^2[x_\varepsilon x_\eta + y_\varepsilon y_\eta]$, $J = \dfrac{1}{(x_\varepsilon y_\eta - y_\varepsilon x_\eta)}$,

$m_1 = J[-y_\eta(m_{11}x_{\varepsilon\varepsilon} + m_{22}x_{\eta\eta} + m_{12}x_{\varepsilon\eta}) + x_\eta(m_{11}y_{\varepsilon\varepsilon} + m_{22}y_{\eta\eta} + m_{12}y_{\varepsilon\eta})]$,

$m_2 = J[y_\varepsilon(m_{11}x_{\varepsilon\varepsilon} + m_{22}x_{\eta\eta} + m_{12}x_{\varepsilon\eta}) - x_\varepsilon(m_{11}y_{\varepsilon\varepsilon} + m_{22}y_{\eta\eta} + m_{12}y_{\varepsilon\eta})]$

Spatial gradients of the computational coordinates appearing in Equations (B.2) are the metrics from a direct transformation and are defined in terms of the indirect metrics:

$$
(B.3) \qquad
\begin{bmatrix}
\varepsilon_x & \varepsilon_y & \varepsilon_z \\
\eta_x & \eta_y & \eta_z \\
\xi_x & \xi_y & \xi_z
\end{bmatrix}
=
\begin{bmatrix}
x_\varepsilon & x_\eta & x_\xi \\
y_\varepsilon & y_\eta & y_\xi \\
z_\varepsilon & z_\eta & z_\xi
\end{bmatrix}^{-1}
$$

Compact finite differencing is used to evaluate the gradients of the Cartesian coordinates in computational space that appear on the RHS of Eq. (B.4).

Equations (B.2) and (B.3) are used to express spatial gradients appearing in the continuity and momentum equations. The continuity equation (2) in the computational coordinate system is given by:

$$
(B.4) \qquad \frac{\partial \varepsilon_m}{\partial x_j} \frac{\partial u_j}{\partial \varepsilon_m} = 0
$$

while the momentum equation (1) is given by

$$
(B.5) \qquad \frac{\partial u_i}{\partial t} + u_j \frac{\partial \varepsilon_m}{\partial x_j} \frac{\partial u_i}{\partial \varepsilon_m} = -\frac{\partial \varepsilon_m}{\partial x_i} \frac{\partial p}{\partial \varepsilon_m} + \frac{1}{Re_D} \frac{\partial \varepsilon_m}{\partial x_j} \frac{\partial}{\partial \varepsilon_m} \left(\frac{\partial \varepsilon_n}{\partial x_j} \frac{\partial u_i}{\partial \varepsilon_n} \right)
$$

Discretizing Eq (B.6) temporally with the RK scheme and spatially with compact finite differencing gives:

$$
(B.6) \qquad u_i^{M+1} = u_i^M + b^M \Delta t \left[H_i^M - \left(\delta_{x_i} \varepsilon_m \right) \left(\delta_{\varepsilon_m} P^M \right) \right]
$$

where
$$
H_i^M = -u_j^M (\delta_{x_j} \varepsilon_m)(\delta_{\varepsilon_m} u_i^M) + \frac{1}{Re_D}(m_{11}\delta_{\varepsilon\varepsilon} + m_{22}\delta_{\eta\eta} + m_{12}\delta_{\varepsilon\eta} + m_1\delta_\varepsilon + m_2\delta_\eta)u_i^M
$$

Taking the divergence of Eq. (B.6) gives the Poisson equation for pressure:

$$
(B.7) \qquad (m_{11}\delta_{\varepsilon\varepsilon} + m_{22}\delta_{\eta\eta} + m_{12}\delta_{\varepsilon\eta} + m_1\delta_\varepsilon + m_2\delta_\eta) P^M
$$

$$
= (\delta_{x_i} \varepsilon_m) \delta_{\varepsilon_m} \left[H_i^M + \frac{u_i^M}{b^{M+1} \Delta t} \right]
$$

Explicit stencils for Eq. (B.7) are difficult to find due to the presence of the metric terms which prevent commutation of the derivative operators. Solution techniques for Eq. (B.7) are based on treating the pressure, P^M, as well as it second derivatives, $\delta_{\varepsilon\varepsilon}P^M$ and $\delta_{\eta\eta}P^M$, as unknowns to be solved simultaneously:

$$
(B.8) \qquad A_{\varepsilon\varepsilon}(\delta_{\varepsilon\varepsilon} P^M) = B_{\varepsilon\varepsilon} P^M
$$

31

(B.9)
$$A_{\eta\eta}(\delta_{\eta\eta}P^M) = B_{\eta\eta}P^M$$

(B.10)
$$(m_{11}\delta_{\varepsilon\varepsilon} + m_{22}\delta_{\eta\eta})P^M$$

$$= (\delta_{x_i}\varepsilon_m)\delta_{\varepsilon_m}\left[H_i^M + \frac{u_i^M}{b^M\Delta t}\right] - (m_{12}\delta_{\varepsilon\eta} + m_1\delta_\varepsilon + m_2\delta_\eta)\tilde{P}^M$$

where $A_{\varepsilon\varepsilon}$, $B_{\varepsilon\varepsilon}$, $A_{\eta\eta}$, and $B_{\eta\eta}$ are the compact matrices for the second derivative scheme defined in Section 2.3.2. The first and mixed derivative terms have been placed on the RHS of Eq. (B.10) and are lagged at the previous iteration. Iterative techniques outlined in Sect. 3 are then used to solve the system of equations (B.8) - (B.10) for the pressure and its second derivatives.
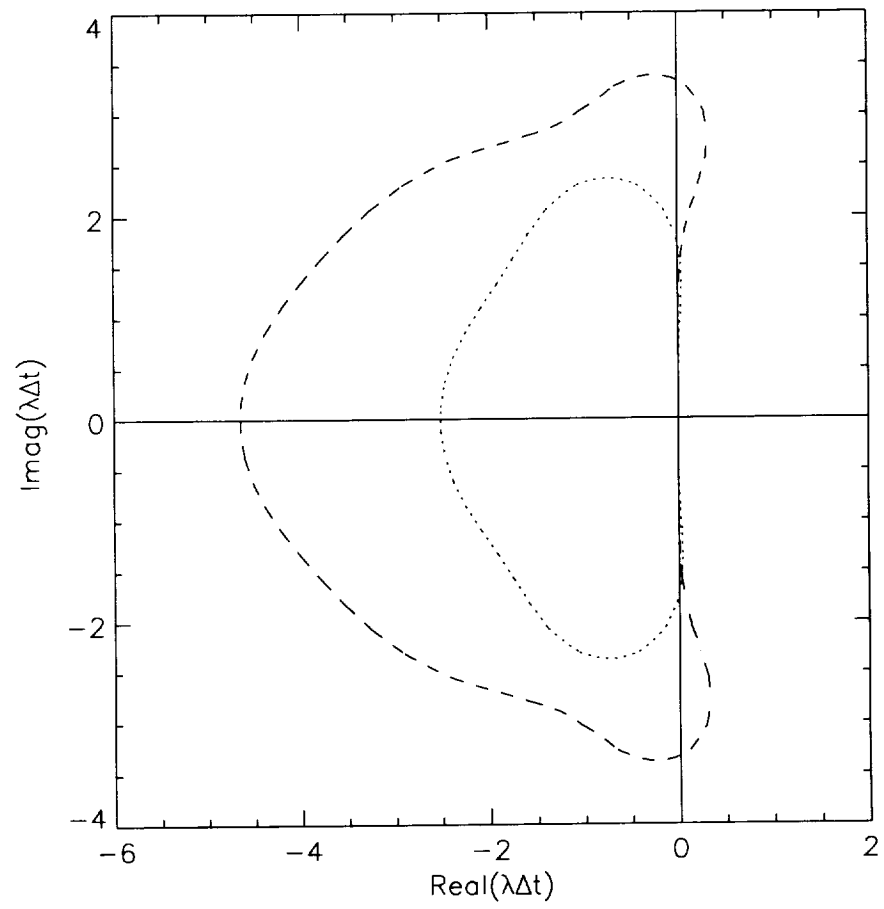
FIG 1   *Stability diagram for Runge Kutta schemes (dotted line, third-order; dashed line, fourth-order).*
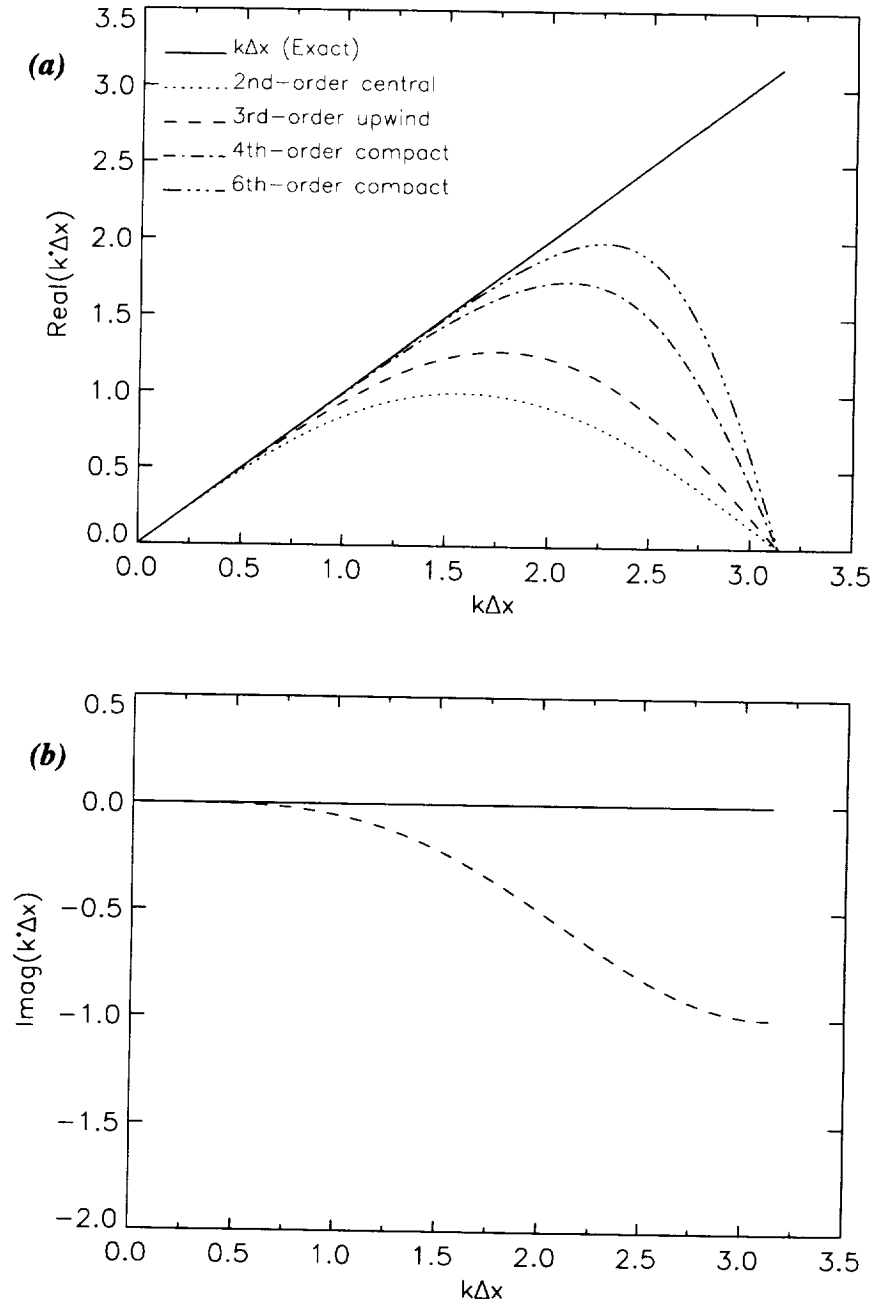
FIG 2   *Finite differencing error for numerical approximations of the first derivative (a) dispersion errors, (b) dissipation errors (zero except for 3rd-order upwind).*
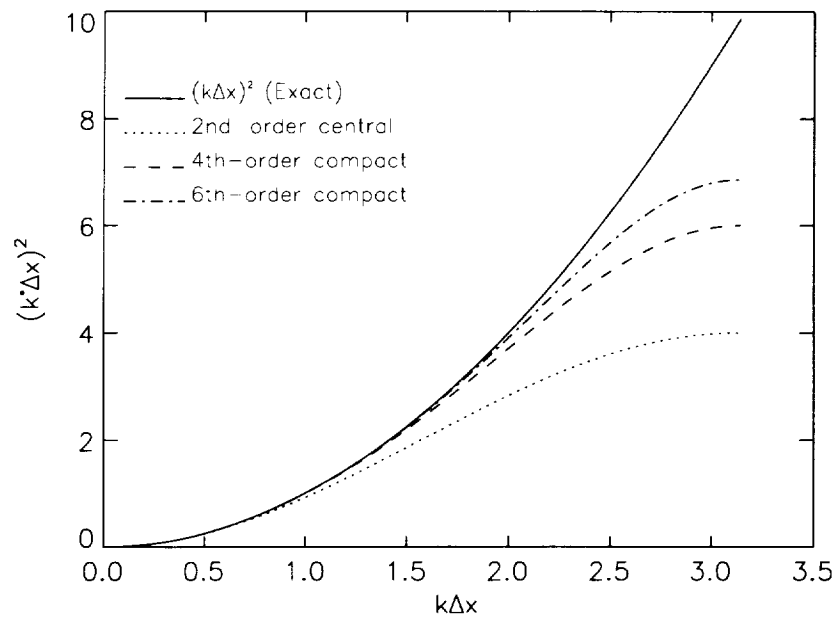
FIG 3   *Finite differencing error for numerical approximations of the second derivative.*
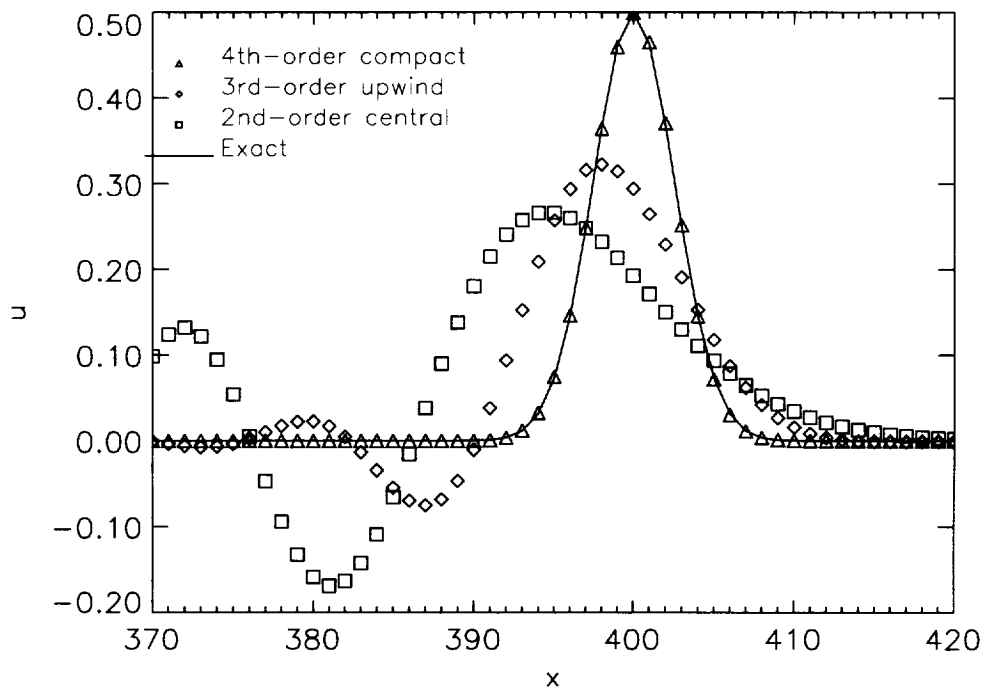
FIG 4   *Solution to the 1-D convection equation at t = 400 in physical space for*
            various finite difference approximations of the first derivative term.
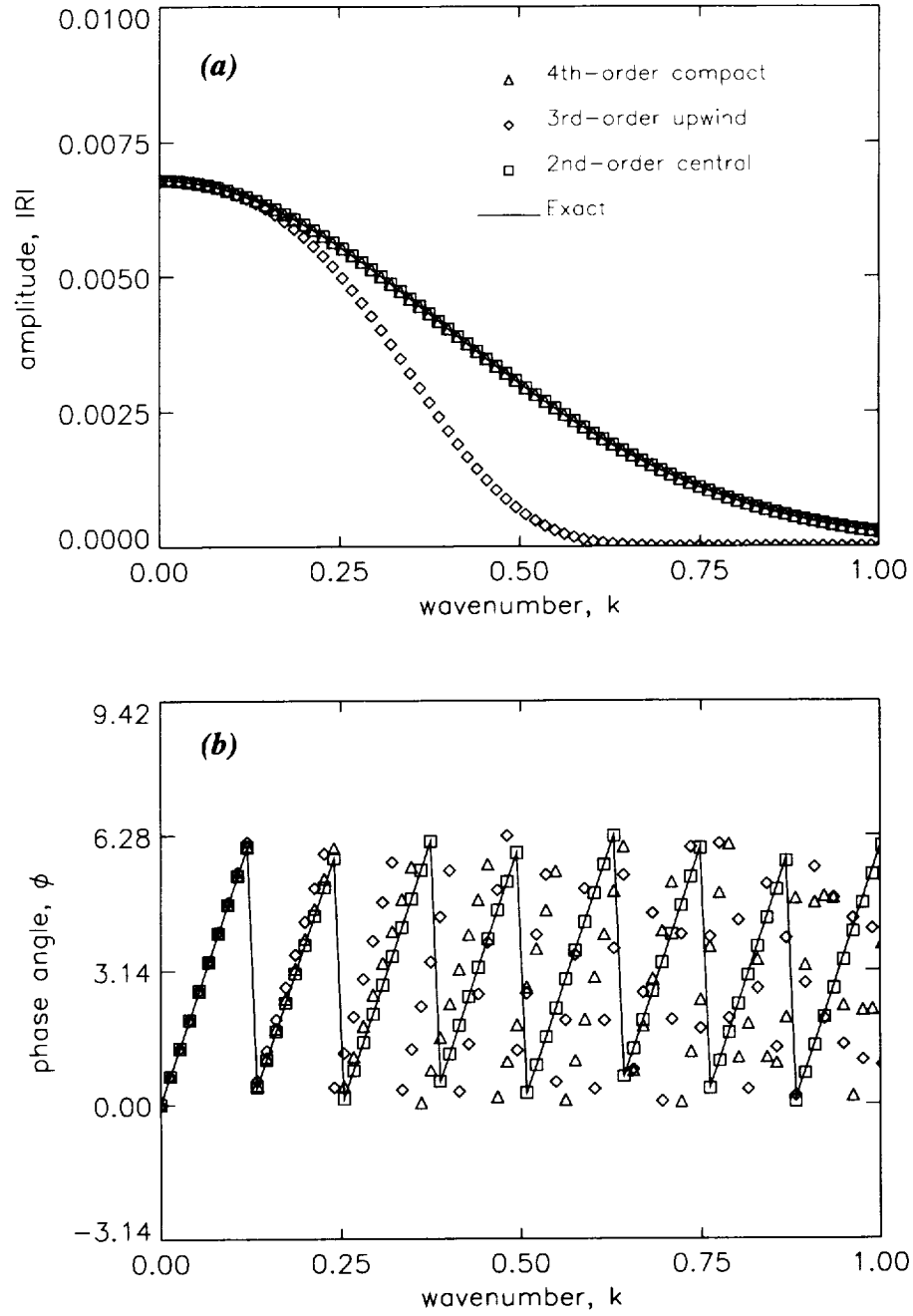
FIG 5   *Solution to the 1-D convection equation at $t = 400$ in wavenumber space for various finite difference approximations of the first derivative term, (a) amplitude and (b) phase angle.*
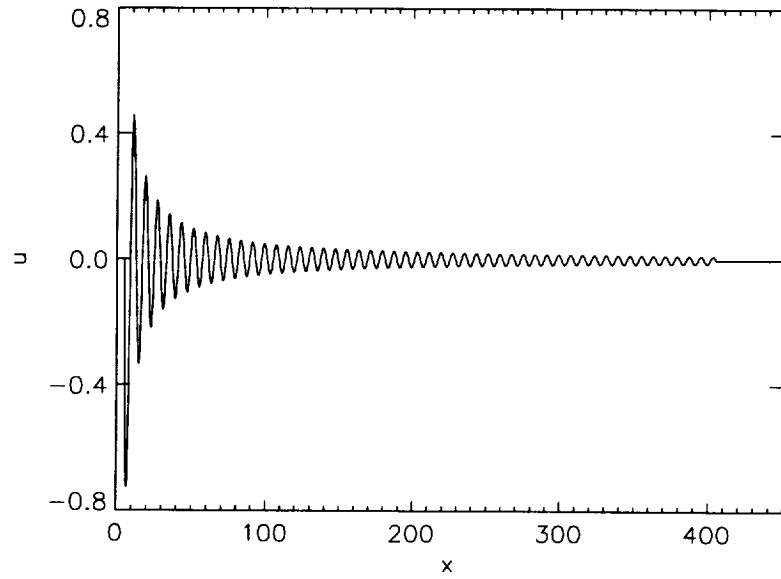
FIG 6   Exact solution to the spherical wave problem at t = 400.
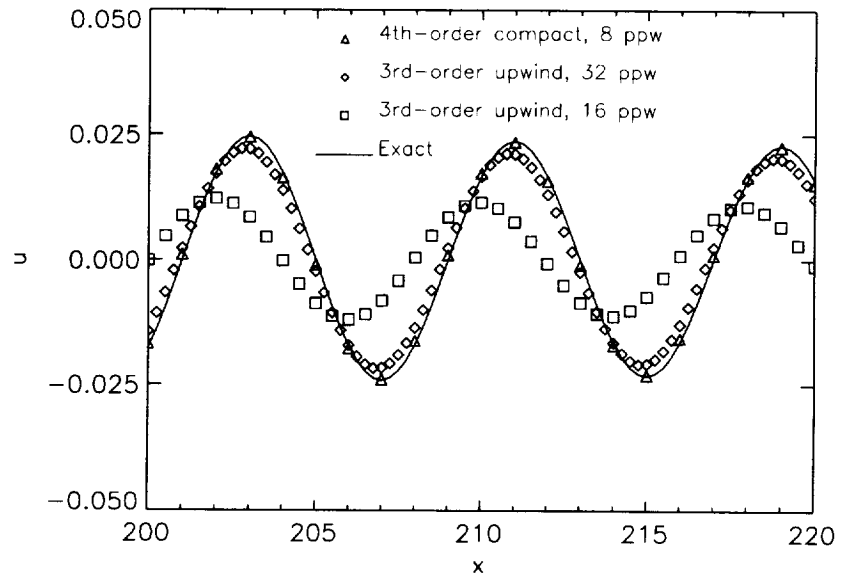
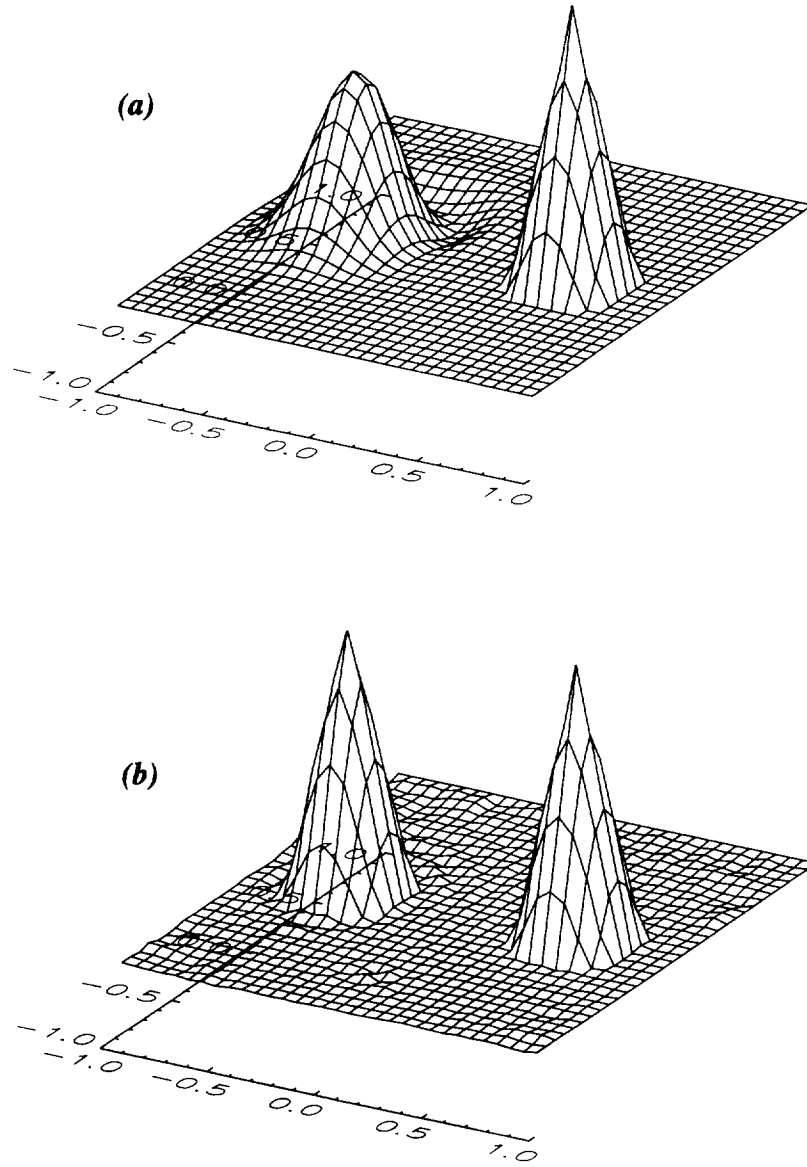FIG 7   Numerical solution of the 1-D spherical wave problem at t = 400 for the region, 200 < x < 220.

FIG 8   *Numerical solution of the rotating cone problem after one revolution on a 32 x 32 grid (a) third-order upwind scheme, (b) fourth-order compact scheme. Numerical solution is shown to the left, exact solution to the right.*
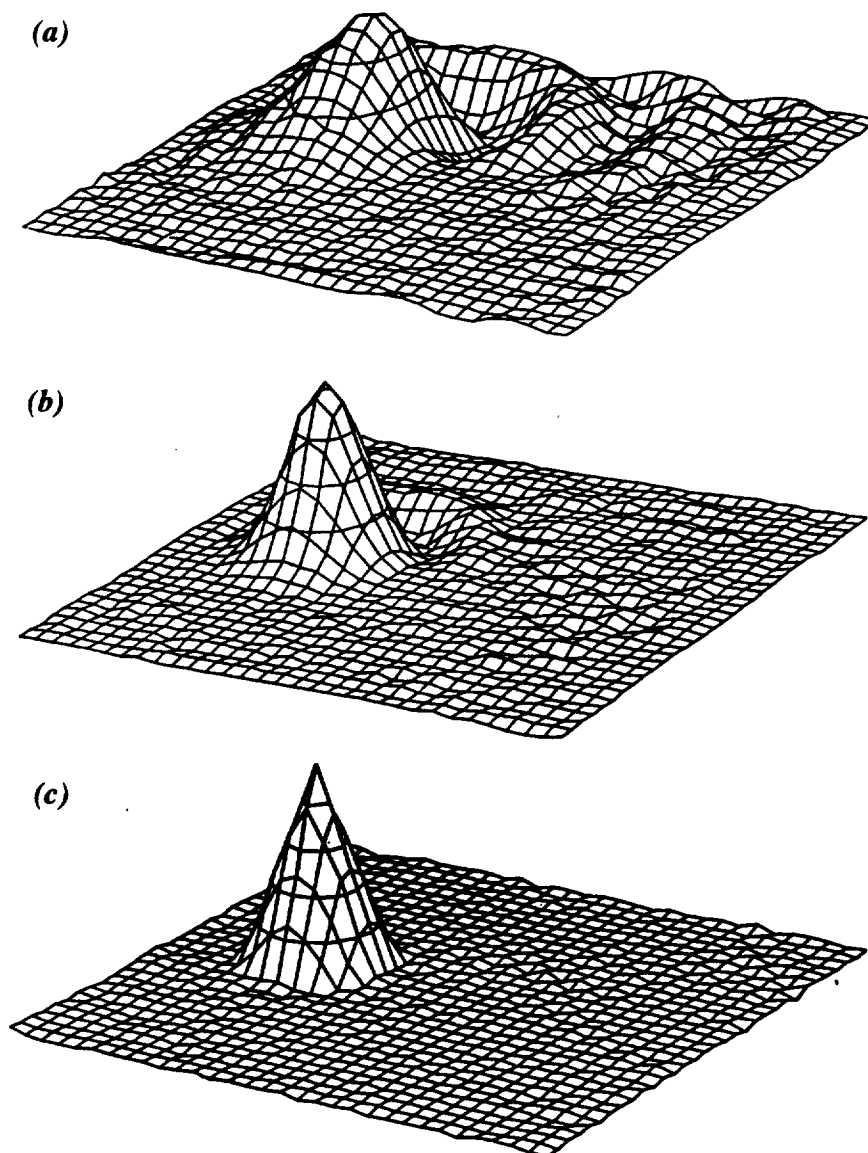
FIG 9   *Numerical solution of the rotating cone problem after one revolution on a 32 x 32 grid from Orszag (1971), (a) second-order Arakawa scheme, (b) fourth-order Arakawa scheme, and (c) spectral methods.*
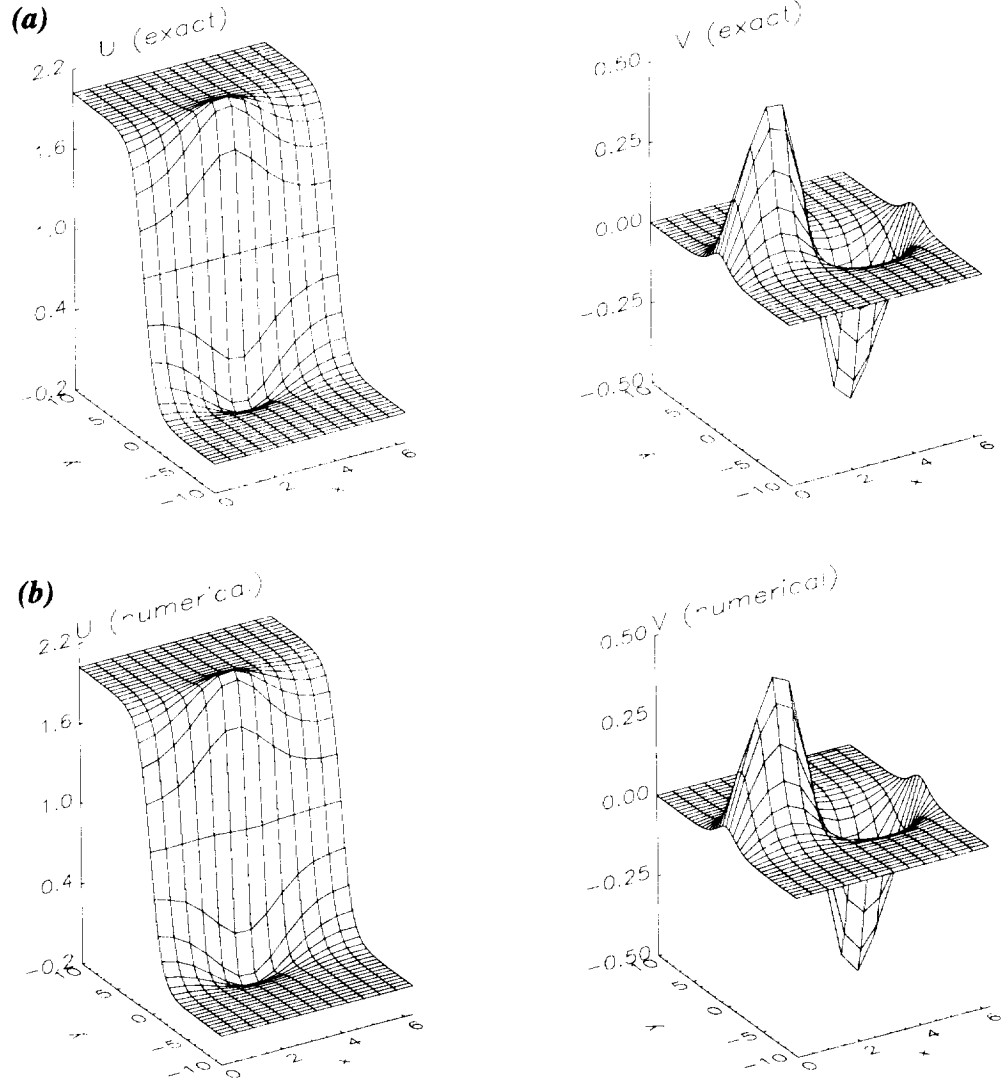
FIG 10   Solution of the Stuart's problem, (a) Exact solution and (b) numerical solution using fourth-order compact scheme on a 13 x 41 grid at   t = 20π   .

REPORT DOCUMENTATION PAGE

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	February 1998	Contractor Report

4. TITLE AND SUBTITLE

Higher-Order Compact Schemes for Numerical Simulation of Incompressible Flows

5. FUNDING NUMBERS

C NAS1-19480
WU 505-90-52-01

6. AUTHOR(S)

Robert V. Wilson
Ayodeji O. Demuren
Mark Carpenter

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Institute for Computer Applications in Science and Engineering
Mail Stop 403, NASA Langley Research Center
Hampton, VA 23681-2199

8. PERFORMING ORGANIZATION REPORT NUMBER

ICASE Report No. 98-13

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Langley Research Center
Hampton, VA 23681-2199

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA/CR-1998-206922
ICASE Report No. 98-13

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified–Unlimited
Subject Category 34
Distribution: Nonstandard
Availability: NASA-CASI (301)621-0390

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

A higher order accurate numerical procedure has been developed for solving incompressible Navier-Stokes equations for 2D or 3D fluid flow problems. It is based on low-storage Runge-Kutta schemes for temporal discretization and fourth and sixth order compact finite-difference schemes for spatial discretization. The particular difficulty of satisfying the divergence-free velocity field required in incompressible fluid flow is resolved by solving a Poisson equation for pressure. It is demonstrated that for consistent global accuracy, it is necessary to employ the same order of accuracy in the discretization of the Poisson equation. Special care is also required to achieve the formal temporal accuracy of the Runge-Kutta schemes. The accuracy of the present procedure is demonstrated by application to several pertinent benchmark problems.

14. SUBJECT TERMS

compact schemes, incompressible flow simulation

15. NUMBER OF PAGES

46

16. PRICE CODE

A03

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified		

NSN 7540-01-280-5500