JOURNAL OF COMPUTATIONAL PHYSICS 95, 313–338 (1991)

A Hybrid Numerical Method for Three-Dimensional Spatially-Developing Free-Shear Flows

JEFFREY C. BUELL

Center for Turbulence Research, MS202A-1,
NASA Ames Research Center, Moffett Field, California 94035

Received May 22, 1989; revised February 23, 1990

A new algorithm has been developed and implemented for solving the three-dimensional incompressible Navier–Stokes equations on a domain that is infinite in the vertical (y) direction, finite in the streamwise (x) direction, and homogeneous in the spanwise (z) direction. A mapped spectral method is used in y, a classical Fourier method is used in z, and high-order compact finite differencing is used in x. A projection method is discussed that ensures exact conservation of mass and satisfaction of the boundary conditions at infinity. The new aspects of these schemes are described, test cases to validate the code are presented, and results for two- and three-dimensional mixing layers are given.

1. INTRODUCTION

We present here a new algorithm for solving the three-dimensional time-dependent incompressible Navier–Stokes equations on domains typical of free-shear problems. The primary difficulty of solving these equations on nonperiodic domains is ensuring conservation of mass at each point in time. This is not difficult for the compressible equations since the equation for density is dynamical and can be advanced in time like the energy and momentum equations. Yet, the continuity constraint is also the primary *advantage* (in terms of numerical efficiency) of the incompressible equations over the compressible system. This constraint alone allows the number of dynamical variables that must be carried to be reduced from five to two. The amount of computer memory, disk space, and arithmetic operations needed can be reduced by similar proportions. Therefore, unless the goal is to study the effect of compressibility itself, one should solve the incompressible equations and strive to extract all the possible efficiencies from doing so.

There are several schemes in current usage that ensure continuity. For two-dimensional (2D) flows, streamfunction-vorticity or pure streamfunction formulations are effective and easily implemented, especially for second-order finite differencing. Comte *et al.* [1] used the former in conjunction with central differencing to study temporally- and spatially-growing mixing layers. For complex geometries, one usually resorts to primitive-variable finite difference methods. Probably the most popular subclass of these are second-order accurate staggered-grid methods

0021-9991/91 $3.00

581/95/2-5

(e.g., see Kim and Moin [2], and references therein). When properly implemented, the numerical divergence of the gradient is identical to the Laplacian operator, both in the interior and adjacent to the boundaries. This allows one to formulate a Poisson equation for pressure (or related variable) that automatically yields a divergence-free velocity field. Davis and Moore [3] used this type of scheme in conjunction with upwind differencing in their 2D simulations of jets and mixing layers. One of the things we will show here is that extra numerical diffusion (in the form of upwinding or added artificial dissipation) is not necessary for free shear flows at moderate Reynolds numbers.

A staggered-grid method with upwind differencing was used also in the three-dimensional (3D) mixing layer simulations of Lowery [4], but only in the streamwise direction. It was combined with the mapped spectral method of Cain *et al.* [5] in the vertical direction and a Fourier method in the spanwise direction [6]. The idea is that finite difference methods are more suitable for use with inflow/outflow boundary conditions and that spectral methods should be used when the boundary conditions permit [7]. However, his implementation of the mapped spectral method produced flows that satisfied continuity and the boundary conditions only approximately. The continuity problem is indicative of the difficulty in general with primitive-variable formulations in finding a scalar field (pressure) whose gradient forces the velocity field to be divergence-free. (See Tuckerman [8] and references therein for analyses of primitive-variable spectral methods.) Despite these problems, Lowery's algorithm turned out to be a good starting point for the development of the one presented here.

Another class of schemes in 3D involves applying the curl operator to the momentum equations in order to eliminate the pressure and reduce the number of dynamical variables to two. This class is sometimes referred to as vector potential or vector streamfunction methods (2D streamfunction schemes are a special case of this class). These are usually restricted to simpler geometries. One method takes two of the components of the curl operation to obtain two vorticity equations. This method has been used in Cartesian coordinates (e.g., see Murdock [9]) and is particularly useful in cylindrical coordinates [10]. Another method, applicable in Cartesian and spherical coordinates, takes one component of the curl operation and the same component of the curl operator applied twice to the momentum equations. The particular component chosen is the inhomogeneous direction (if there is only one), or the direction with the most complex boundary conditions. At this point, one may introduce solenoidal velocity fields based on similar operations [8, 11], or retain more primitive variables (i.e., the appropriate components of the velocity and vorticity [12]). The former is conceptually simpler (especially for linear stability problems), but the latter appears to require fewer arithmetic operations and so it is the approach we take here. We note that, as far as the equation formulation is concerned, the inflow/outflow boundary conditions considered here cause the same difficulties (and yield to the same methods) as the rigid-wall conditions considered in the papers referred to above. An important advantage of this class of schemes is that continuity is either imposed directly or replaced with solenoidal velocity fields;

there is no extra restriction on the properties of the numerical approximation of the Poisson equation, as there is in primitive-variable formulations. Furthermore, given a set of boundary conditions, all vector potential methods are equivalent analytically. However, with some of these methods the right number of boundary conditions are associated with each equation, while the other methods require solving the entire system of equations simultaneously with all the boundary conditions. While the latter may be accomplished with the use of a capacitance matrix method, it is always more work than the former. Thus, in practice, the choice of formulation is dictated by the ease of implementation of boundary conditions.

The governing equations, boundary conditions, and general issues are discussed in the next section. The numerical method, based on high-order compact finite differencing in the streamwise direction and spectral methods in the other two, is described in Section 3. Three analytic tests of the resulting code are given in Section 4. In Section 5, examples of results from 2D and 3D simulations of mixing layers illustrate the capabilities of the method. Conclusions are presented in the last section.

2. Governing Equations and Boundary Conditions

A general schematic and coordinate system for the class of free-shear problems we would like to solve is shown in Fig. 1. A mean inflow velocity profile, $U_0(y)$, is specified at $x=0$. Perturbations added at the inflow will then develop in the streamwise direction (x). The domain extends to infinity in the vertical (y) direction and is assumed to be homogeneous in the spanwise (z) direction. At $y=\pm\infty$, finite entrainment velocities (that are independent of the horizontal coordinates and time) are specified. We assume $U_0(y)$ contains a length scale δ_ω and a velocity scale ΔU. For the mixing layer profile shown, δ_ω is the vorticity thickness and

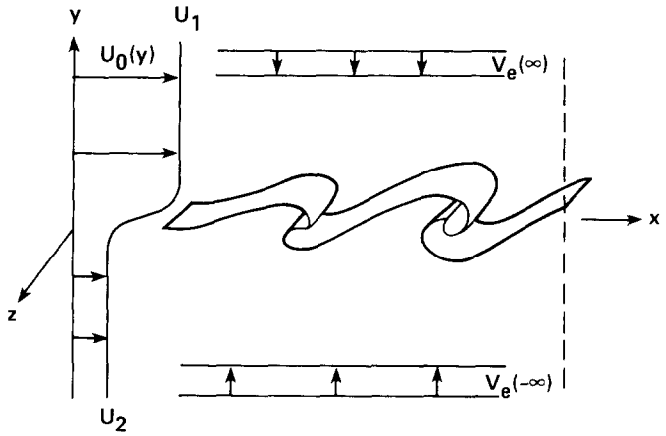

FIG. 1. Problem geometry and coordinate system.

$\Delta U = U_1 - U_2$ is the freestream velocity difference. Nondimensionalizing the Navier–Stokes and continuity equations with these scales yields

$$\frac{\partial \mathbf{U}}{\partial t} = \mathbf{H} - \nabla\left(p + \frac{1}{2}\mathbf{U}\cdot\mathbf{U}\right) + \frac{1}{\mathrm{Re}}\nabla^2\mathbf{U}, \tag{1}$$

$$\nabla\cdot\mathbf{U} = 0, \tag{2}$$

where the advection terms in rotational form are written as $\mathbf{H} = \mathbf{U}\times\boldsymbol{\omega}$, the vorticity is $\boldsymbol{\omega} = \nabla\times\mathbf{U}$ and the Reynolds number is $\mathrm{Re} = \delta_\omega(U_1 - U_2)/\nu$. An additional parameter for mixing layer profiles is the velocity ratio, defined here as $\Lambda = U_2/U_1$. We note that a larger velocity ratio ($\Lambda \to 1$) makes the layer more temporal in nature [13], while a small ratio may yield a negative streamwise velocity at the exit plane.

One of the reasons we need to eliminate the gradient term in (1) is because in the numerical approximation to be used here the divergence of the gradient is not exactly identical to the Laplacian. The other reasons (as mentioned above) are related to numerical efficiency. Also, since the numerical method requires homogeneous Dirichlet boundary conditions at infinity on all computational variables, we define a perturbation velocity $\mathbf{u} = (u, v, w)$ by

$$\begin{aligned} U(x, y, z, t) &= U_0(y) + xU_e(y) + u(x, y, z, t), \\ V(x, y, z, t) &= V_e(y) + v(x, y, z, t), \\ W(x, y, z, t) &= w(x, y, z, t), \end{aligned} \tag{3}$$

where V_e is some smooth function that tends to the entrainment values at $\pm\infty$ but is otherwise arbitrary, and $U_e = -\partial V_e/\partial y$. The gradient term in (1) is eliminated by operating on the momentum equations with $\nabla\times$ and $\nabla\times\nabla\times$. Retaining the streamwise components of both and using (2) yields dynamical equations for the streamwise perturbation velocity and vorticity:

$$\frac{\partial}{\partial t}\nabla^2 u = \nabla_\perp^2 H_1 - \frac{\partial^2}{\partial x\,\partial y}H_2 - \frac{\partial^2}{\partial x\,\partial z}H_3 + \frac{1}{\mathrm{Re}}\nabla^4 U, \tag{4}$$

$$\frac{\partial}{\partial t}\omega_1 = \frac{\partial}{\partial y}H_3 - \frac{\partial}{\partial z}H_2 + \frac{1}{\mathrm{Re}}\nabla^2\omega_1, \tag{5}$$

where

$$\nabla_\perp^2 \equiv \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$$

is the "perpendicular" Laplacian. The streamwise components of the curl operations are preferred over the other two because ω_1 is the only vorticity component with boundary conditions fully defined by Dirichlet boundary conditions on the velocity. Equations (4) and (5) govern the time advancement of u and ω_1. However, $\mathbf{H}$

contains the other velocity and vorticity components which must be recovered from u, ω_1, and the boundary conditions. The spanwise velocity w is recovered from the continuity equation and the definition of ω_1, and then the vertical velocity v is obtained directly from the continuity equation:

$$\nabla_\perp^2 \omega = \frac{\partial \omega_1}{\partial y} - \frac{\partial^2 u}{\partial x \, \partial z}, \tag{6}$$

$$\frac{\partial v}{\partial y} = -\frac{\partial u}{\partial x} - \frac{\partial w}{\partial z}. \tag{7}$$

Once all the velocity components are known, the other vorticity components follow from their definitions.

Dirichlet boundary conditions are specified for $\mathbf{u}$ at the inflow ($x = 0$) and outflow ($x = L_x$) boundaries. Since (4) is a fourth-order equation, it requires a second boundary condition at each end. These are Neumann conditions obtained from continuity and the boundary conditions on v and w:

$$\frac{\partial u}{\partial x} = -\frac{\partial v}{\partial y} - \frac{\partial w}{\partial z}, \qquad x = 0, L_x. \tag{8}$$

Note that (7) and (8) are not redundant since the former will never be applied at the boundaries. Boundary conditions on ω_1 are also obtained from the conditions on v and w:

$$\omega_1 = \frac{\partial w}{\partial y} - \frac{\partial v}{\partial z}, \qquad x = 0, L_x. \tag{9}$$

Periodic boundary conditions are used in the spanwise direction over the domain $0 \leqslant z \leqslant L_z$. As implied by (3), $\mathbf{u} = 0$ at infinity.

At the exit, each velocity component is required to satisfy a "convective" outflow boundary condition of the form

$$\frac{\partial \psi}{\partial t} = -c \frac{\partial \psi}{\partial x}, \tag{10}$$

where $c = (U_1 + U_2)/2$ is the nominal speed of the large structures [4]. This boundary condition is a severe approximation of the Navier–Stokes equations obtained by assuming the vortical structures are "frozen" as they leave the domain. The success of this condition relies on having a positive total streamwise velocity at the outflow boundary to "wash" any errors produced out of the domain. The opposite happens, usually with disastrous consequences, if the velocity ratio Λ is too small (say, $\Lambda < 0.15$). Note that replacing c with the local streamwise velocity makes the condition less physical, because the effects of pressure in keeping the structures "coherent" would then be neglected.

Buell and Huerre [14] showed that while (10) allows vortical structures to pass out of the domain with essentially no upstream effect on the vorticity, it does create a small-amplitude potential flow. This can be thought of as due to the sudden "impedance" change between the Navier–Stokes equations in the interior and the above condition at the outflow boundary. The incompressibility constraint allows the potential fluctuations to be communicated everywhere instantaneously. These fluctuations do not appear to interact with the shear layer in the interior (that is, the "receptivity" of the shear layer to potential fluctuations is small), but they do interact with the inflow boundary conditions to produce vortical fluctuations near the inflow. These, in turn, are amplified as they are convected downstream. This "global feedback" mechanism allows the mixing layer to be self-sustained even though mixing layers have been shown to be only (locally) convectively unstable and not absolutely unstable [15]. Clearly, better boundary conditions are needed at either the inflow or the outflow to break the feedback loop. As a practical matter, the feedback appears to have an effect similar to low-amplitude noise added to the inflow boundary conditions. We note that experimental facilities have the same problem; small changes in wind tunnel geometry either upstream or downstream of the test section can cause significant changes in the test section.

3. Numerical Method

In this section we describe the numerical method used to solve (4)–(7) with the associated boundary conditions. Since we intend to treat the terms on the right-hand sides (RHSs) of (4) and (5) explicitly, they only require methods to evaluate derivatives in each direction and a method for evaluating nonlinear terms. The LHSs of (4) and (6) need methods for the inversion of 3D and 2D Poisson equations, respectively. The recovery of v from (7) requires a y-integral. All of the calculations are performed in physical space in x and Fourier (or "wave") space in y and z, except for the evaluation of the nonlinear terms, $\mathbf{H}$. These are evaluated by transforming the velocity and vorticity fields to physical space in y and z, performing the cross product, and transforming back. The nonlinear terms are then of the same form as the viscous terms, and both may be treated like source terms.

For second-order equations like (5), the implementation of boundary conditions is usually obvious. But, when and how to implement the four boundary conditions on u for the fourth-order u-equation (4) is not as obvious. The basic principle underlying the numerical implementation of boundary conditions is the same as the reason boundary conditions are needed analytically: the spatial derivatives appearing in the problem require them. Consequently, the biharmonic operator in (4) requires four boundary conditions regardless of whether it is to be evaluated in an explicit time advancement scheme (as in this paper) or inverted in an implicit one. The role of the two Neumann conditions is similar to their role in streamfunction–vorticity methods. Together with the Dirichlet conditions they allow the computation of the Laplacian of u (or of the streamfunction) on the boundary,

which in turn is needed to calculate ∇^2 ($\nabla^2 u$) (or ∇^2 vorticity) in the interior. Trying to replace implementation of Neumann conditions in this case with one-sided differencing for the first Laplacian will yield an ill-posed problem and disastrous results. On the other hand, the last operation in the time advancement of (4) is to invert the Laplacian on the left-hand side while imposing only the two Dirichlet conditions. Yet the resulting solution u will satisfy all four boundary conditions. Indeed, the only time we need to know or care about the derivative of u at the boundary is when the biharmonic is evaluated. An examination of standard numerical methods for boundary value problems shows that they do make use of the boundary conditions in the approximation of the spatial derivatives and for no other purpose.

The next five subsections discuss the various numerical methods for the operations mentioned above.

3.1. *Approximation of Streamwise Derivatives*

The first and second x-derivatives in (4)–(7) are approximated with new Padé finite difference formulas due to Lele [16, 17]. An approximation for the fourth derivative is not needed since the biharmonic in (4) is evaluated as two successive Laplacians. Most of the following is a specialization of the general formulas presented in [17] to simpler formulas. A one-parameter family of fourth-order accurate schemes for the first derivative of a function $f(x)$ is given by

$$f'_{j-1} + \alpha f'_j + f'_{j+1} = \frac{1+2\alpha}{3\Delta x}(f_{j+1} - f_{j-1}) + \frac{4-\alpha}{12\Delta x}(f_{j+2} - f_{j-2}), \tag{11}$$

where the prime denotes the numerical derivative, the subscript j, $0 \leqslant j \leqslant J$, refers to the grid point number, and $\Delta x = L_x/J$. The classical fourth-order Padé formula is obtained by setting $\alpha = 4$, while $\alpha = 3$ (which is used here) gives a sixth-order scheme.

The "order" of a scheme pertains to its behavior only for vanishingly small wavenumbers. As useful as this concept is, we need a measure of the accuracy of a scheme for moderate to large wavenumbers. If we define an effective wavenumber k_e by differentiating a periodic function,

$$f \equiv \exp(ikx), \qquad f' = ik_e \exp(ikx), \tag{12}$$

then the accuracy of various schemes for all wavenumbers may be compared. Inserting (12) into (11) and into the standard second- and fourth-order central differencing formulas yields the curves shown in Fig. 2. Fourier methods are exact for (12), thus $k_e = k$ (the diagonal line in Fig. 2). The dominant error in these approximations is dispersion, which is related to the departure of k_e from k. Clearly the Padé approximations are superior to the central differencing ones; the former allows significant energy to exist in modes up to half the spatial Nyquist frequency ($k_N \equiv \pi/\Delta x$) without much dispersion. Furthermore, the sixth-order scheme is

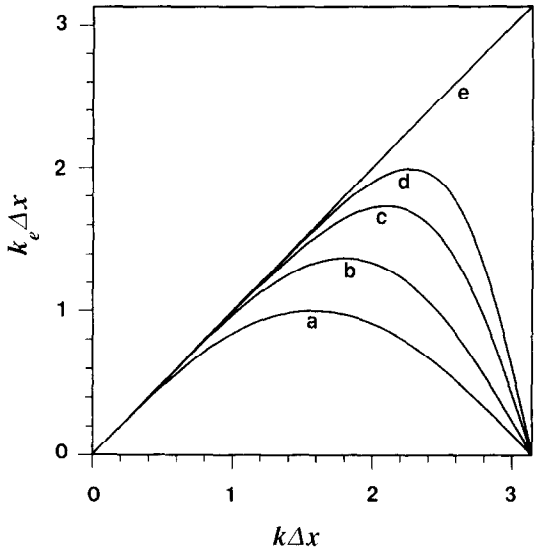

FIG. 2. Effective wavenumber for numerical approximations of the first derivative: (a) second-order central differencing; (b) fourth-order central differencing; (c) Eq. (7) with $\alpha = 4$ (classical Padé); (d) Eq. (7) with $\alpha = 3$ (sixth order); and (e) Fourier (exact).

significantly more accurate than the classical Padé scheme. None of the five schemes produce any numerical dissipation since k_e is real in all cases.

By increasing the bandwidth of either side of (11), a multiple-parameter family of fourth-order schemes can be constructed [17]. This family contains members that are accurate up to $k = 0.9k_N$, approaching Fourier methods in accuracy. It is not clear at what point increases in complexity begin to outweigh increases in accuracy, or what the importance is of aliasing errors. The answers to these questions will determine whether one should upgrade the present algorithm to include more accurate finite difference approximations.

At and near the boundaries lower-order formulas must be used. For $j = 0$ and $j = J$, compact one-sided third-order approximations are used:

$$f'_0 + 2f'_1 = \frac{1}{2\Delta x}(-5f_0 + 4f_1 + f_2),$$
$$f'_J + 2f'_{J-1} = \frac{1}{2\Delta x}(5f_J - 4f_{J-1} - f_{J-2}). \tag{13}$$

To evaluate $\partial u/\partial x$ on the RHS of (7), the above are replaced with analytical expressions, since we have boundary conditions on this particular derivative. At $j = 1$ and $j = J - 1$ (11) is used with $\alpha = 4$, and at $j = 2$ and $j = J - 2$ (11) is used but with $\alpha' = (16 + 32\alpha)/(40 - \alpha)$ substituted for α. As discussed in [17], these modifications ensure both stability and numerical conservation for equations of the form

$(\partial/\partial t)u = (\partial/\partial x) f(u)$. Unlike wall-bounded flows, there is little penalty for using less-accurate formulas at the boundaries of free-shear flows. At the inflow the streamwise gradients are small, and at the outflow extra numerical errors are washed out of the domain.

For the second derivative, Lele [16, 17] presents a large bandwidth formula which we specialize to a one-parameter family of fourth-order accurate formulas:

$$f''_{j-1} + \gamma f''_j + f''_{j+1} = \frac{4(\gamma-1)}{3\Delta x^2}(f_{j-1} - 2f_j + f_{j+1}) + \frac{10-\gamma}{12\Delta x^2}(f_{j-2} - 2f_j + f_{j+2}). \tag{14}$$

Here, $\gamma = 10$ is the classical Padé formula and $\gamma = \frac{11}{2}$ produces the sixth-order version. The effective wavenumber is defined by

$$f \equiv \exp(ikx), \qquad f'' = -k_e^2 \exp(ikx). \tag{15}$$

In Fig. 3, k_e^2 is plotted versus k for the two central-differencing schemes, three values of γ, and the Fourier method (which yields a parabola). For the evaluation of the viscous terms, k_e^2 is proportional to the numerical dissipation. Among the finite difference schemes plotted, we see that $\gamma = 4$ comes closest to reproducing the physical dissipation and is the value we use here. It is slightly over-dissipative at moderately high wavenumbers and under-dissipative (like the other schemes) at the

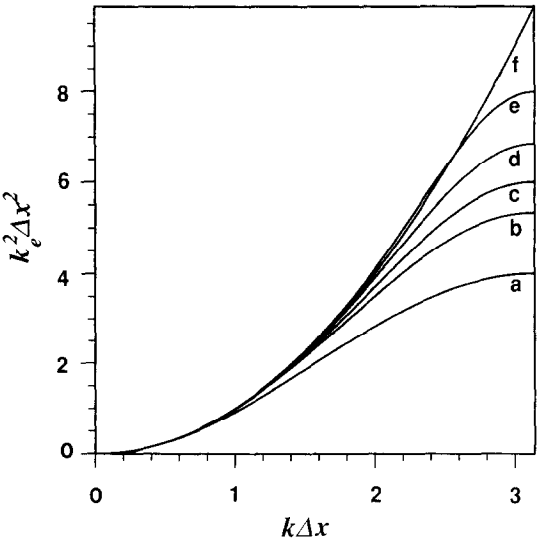

FIG. 3. Effective wavenumber for numerical approximations of the second derivative: (a) second-order central differencing; (b) fourth-order central differencing; (c) Eq. (10) with $\gamma = 10$ (classical Padé); (d) Eq. (10) with $\gamma = 5.5$ (sixth order); (e) Eq. (10) with $\gamma = 4$; and (f) Fourier (exact).

highest wavenumbers. As pointed out by S. Lele, the preferred member of a fourth-order family of schemes is not necessarily the sixth-order one. However, the sixth-order schemes for both the first and second derivatives coincide with the ones that have the highest values of $k_e(k)$, with the constraint that $k_e(k) \leqslant k$.

At the boundaries one of the following two sets of third-order formulas is used:

$$\begin{aligned} f_0'' + 11f_1'' &= \frac{1}{\Delta x^2}(13f_0 - 27f_1 + 15f_2 - f_3), \\ f_J'' + 11f_{J-1}'' &= \frac{1}{\Delta x^2}(13f_J - 27f_{J-1} + 15f_{J-2} - f_{J-3}) \end{aligned} \tag{16}$$

or

$$\begin{aligned} f_0'' + 2f_1'' &= -\frac{3}{\Delta x}\left.\frac{df}{dx}\right|_{x=0} - \frac{3}{2\Delta x^2}(f_0 - f_2), \\ f_J'' + 2f_{J-1}'' &= \frac{3}{\Delta x}\left.\frac{df}{dx}\right|_{x=L_x} - \frac{3}{2\Delta x^2}(f_J - f_{J-2}). \end{aligned} \tag{17}$$

The latter pair of formulas is used when boundary conditions on both the function and its derivative are available, as is the case for u. The former pair is used in all other cases. Note that well-posedness *requires* the use of (17) when the fourth derivative is to be calculated, but its use is optional if just the second derivative is needed. At $j = 1$ and $j = J - 1$, (14) is used with $\gamma = 10$.

3.2. *Approximation of Vertical Derivatives and Integrals*

In this section we briefly review the method of Cain *et al.* [5] and describe new modifications to it. The physical domain $-\infty \leqslant y \leqslant \infty$ is mapped to the unit interval $0 \leqslant \zeta \leqslant 1$, using

$$y = -\beta \cot(\pi\zeta),$$

where β is the mapping parameter. One of the properties of this mapping is that a uniform grid on ζ yields an algebraically-stretched grid for large y, viz., $\Delta y \sim y^2$ as $y \to \infty$. This property is necessary to resolve the algebraically-decaying potential flow associated with vortical structures. Grosch and Orszag [18] and Metcalfe *et al.* [19] compared calculations for several problems based on both algebraically- and exponentially-stretched grids and found that the former is significantly more accurate when a potential flow must be resolved. For the calculations reported here, we find that the best results obtain when only half of the grid points in y contain vortical flow. With two homogeneous directions, it is possible to represent the potential flow with a small number of functions and to use successfully a mapping with an exponentially-stretched grid for large y. This results in significant gains in efficiency; upwards of 80% of the grid points may contain vortical flow [20, 21]. It should be possible to apply such ideas to flows with more inhomogeneous directions, but this has not yet been investigated.

Given a function $f(y)$, y-derivatives are found from

$$\frac{df}{dy}=\frac{d\zeta}{dy}\frac{df}{d\zeta}=\frac{1}{\pi\beta}\sin^2(\pi\zeta)\frac{df}{d\zeta}. \tag{18}$$

If f is represented by a finite trigonometric series, then it is a trivial matter to write down its derivative analytically as another finite series using (18). However, because of the $\sin^2$ term in (18) the latter will have two more terms than the former. Since we want the derivative series to contain no more terms than the function series, an approximation must be made. One possible approximation was used in [5], but it produces nonzero derivatives at infinity and leads to flows with nonzero divergence. While these errors are of the same order of magnitude as the last terms in the series (which should be small), the method would be more satisfactory if the boundary conditions (no fluctuations at infinity) and conservation of mass are satisfied exactly.

First, consider a function f that can be represented exactly by a finite sine series,

$$f(y)=\sum_{m=0}^{M}\hat{f}_m\sin(\pi m\zeta). \tag{19}$$

We want an approximate derivative of the form

$$\frac{df}{dy}\cong g(y)=\sum_{m=0}^{M}\hat{g}_m\cos(\pi m\zeta) \tag{20}$$

that satisfies the above constraints and can be calculated from

$$\hat{\mathbf{g}}=\mathbf{B}\hat{\mathbf{f}}, \tag{21}$$

where $\mathbf{B}$ is a tridiagonal matrix of order $M+1$ (when reordered properly; assuming M is even, the $M/2$ odd modes decouple from the $M/2+1$ even modes). The elements of a matrix $\mathbf{B}$ satisfying all these constraints are given in the Appendix. It is easily verified that any derivative calculated with this approximation is zero at infinity and that the derivative is analytic if $\hat{f}_{M-1}=\hat{f}_M=0$. If f is defined by a cosine series, then the matrix $\mathbf{A}$ yields the sine series approximation to the derivative ($\mathbf{A}$ is given in the Appendix). The second-derivative matrix operator is obtained by multiplying the first-derivative matrices together. For a cosine series we use $\mathbf{BA}$, and for a sine series, $\mathbf{AB}$.

Before we can proceed further we need to decide how to expand each component of the velocity vector (the expansions for the vorticity components follow directly from the velocity expansions). With the restrictions implied by the continuity equation, there are three choices:

(a) cosine expansions for u and w, sine expansion for v,

(b) sine expansions for u and w, cosine expansion for v, or

(c) periodic expansions for all three components.

Any of these may be used. We chose (a) (mostly for historical reasons), although (c) may be slightly more efficient numerically. One advantage of (a) is that it is compatible with a sine expansion for ω_1, which means its boundary conditions at infinity are satisfied automatically.

The other operation that needs to be performed is integration of (7). Since this is the step that guarantees continuity, we want to be able to do it exactly. Replacing the RHS of (7) with b and suppressing x, z, and t for brevity, we write

$$v(y) = \int_{-\infty}^{y} b(\xi)\, d\xi. \tag{22}$$

In order for v to be well represented by a sine series, (22) implies that b must satisfy three constraints. Finiteness of v requires $b(-\infty) = b(\infty) = 0$, and the boundary conditions $v(-\infty) = v(\infty) = 0$ require $\int_{-\infty}^{\infty} b(y)\, dy = 0$. If we assume

$$b(y) = \sum_{m=0}^{M} \hat{b}_m \cos(\pi m \zeta), \tag{23}$$

then the three constraints become

$$\sum_{\substack{m=1 \\ m \text{ odd}}}^{M-1} \hat{b}_m = 0, \qquad \sum_{\substack{m=0 \\ m \text{ even}}}^{M} \hat{b}_m = 0, \tag{24a}$$

$$\int_{-\infty}^{\infty} b(y)\, dy = -\pi\beta \sum_{\substack{m=2 \\ m \text{ even}}}^{M} m\hat{b}_m = 0. \tag{24b}$$

Since (23) and (24) together imply that there are only $M-2$ independent elements of $\hat{\mathbf{b}}$ and since $\mathbf{B}\hat{\mathbf{v}}$ gives the analytical derivative of v if $\hat{v}_m = 0$, $m > M-2$, the exact solution of (22) can be written as

$$v(y) = \sum_{m=1}^{M-2} \hat{v}_m \sin(\pi m \zeta), \tag{25}$$

where $\hat{\mathbf{v}}$ is found from $\mathbf{B}\hat{\mathbf{v}} = \hat{\mathbf{b}}$. In practice, one removes the first row and last two rows of this system to obtain a square nonsingular tridiagonal system of order $M-2$ which is easily solved. The three removed equations will be satisfied if (24) holds. As a check, one can verify that $\mathbf{B}$ contains three linear dependencies corresponding to the solvability conditions on $\hat{\mathbf{b}}$. Returning to (7), we note that (24) applies to both u and w.

3.3. *Approximation of Spanwise Derivatives*

The treatment of the spanwise direction is classical [6]. We use periodic functions of the form

$$f(z) = \sum_{n=0}^{N/2-1} \tilde{f}_n \exp(i2\pi n z / L_z) + \text{c.c.}, \tag{26}$$

where N is even, $\tilde{f}_n$ is complex, and c.c. denotes the complex conjugate. Derivatives are analytical:

$$\frac{df}{dz} = \frac{i2\pi}{L_z} \sum_{n=0}^{N/2-1} n\tilde{f}_n \exp(i2\pi nz/L_z) + \text{c.c.} \tag{27}$$

3.4. *Solution of the Poisson Equation*

The solution of the elliptic operators in (4) and (6) must satisfy (24), so it is important for them to be inverted properly. The result of the time advancement scheme (next section) applied to (4) is a Poisson equation:

$$\nabla^2 \Delta u(x, y, z) = r(x, y, z), \tag{28}$$

where Δu is the difference of u between two successive substeps, and r is a linear combination of the RHS of (4) evaluated at two time levels. In this section we suppress the explicit dependence on t, since all quantities are evaluated at the same time level. There are two major problems with using the approximations discussed above without modification on the LHS of (28). First, any finite difference method in the x direction combined with a mapped Fourier method in the y direction would yield a very large sparse matrix. There are many ways of solving such a matrix, but they are all much slower than methods that decouple the two directions and then invert one-dimensional operators. Second, even after inverting such a matrix, the result (after solving (6) and (7)) may not satisfy continuity or the boundary conditions.

Standard finite difference methods applied to a multi-dimensional Poisson equation share the first problem. Fortunately, it is possible in some cases to decouple one or more of the directions from the others through the use of a transform method. These methods are based on expanding the solution in terms of the eigen-

[redacted]

the equation to eigenfunction space yields a diagonal operator for these directions. The final solution is obtained by transforming the result of the Poisson inversion back to the original space. In the staggered-grid method of Kim and Moin [2] it is possible to do this exactly since the eigenfunctions and associated eigenvalues are known analytically. This is central to their method because the satisfaction of continuity requires the numerical approximation of the Laplacian to equal the divergence of the gradient, and this property must be preserved through the transformations. Here, we propose a transform method where both of these features are sacrificed in exchange for greater generality and accuracy. The use of approximate eigenfunctions and eigenvalues results in extra truncation error (consistent with the overall scheme) near the boundaries. This is not expected to be important in free shear flows where much more accuracy is required in the interior of the domain than near the inflow and outflow boundaries. The lack of the identity $\nabla \cdot \nabla = \nabla^2$ for the numerical operators is not significant since the continuity equation is inverted directly.

Since Δu does not satisfy periodic, homogeneous Dirichlet or homogeneous Neumann boundary conditions, a transform method cannot be used directly. Instead we must first define a computational variable u^* that does satisfy one of these boundary conditions. We choose homogeneous Dirichlet conditions and set

$$\Delta u = u^* + \left(1 - \frac{x}{L_x}\right) f_0(y, z) + \frac{x}{L_x} f_J(y, z) + h_0(x)\, g_0(y, z) + h_J(x)\, g_J(y, z), \quad (29)$$

where

$$f_0(y, z) \equiv \Delta u(x = 0, y, z), \qquad f_J(y, z) \equiv \Delta u(x = L_x, y, z),$$

$$h_0(x) \equiv -\frac{x}{6L_x}(2L_x^2 - 3xL_x + x^2), \qquad h_J(x) \equiv -\frac{x}{6L_x}(L_x^2 - x^2),$$

$$g_0(y, z) \equiv \frac{\partial^2}{\partial x^2} \Delta u(x, y, z)|_{x=0} = e_0(y, z) - \nabla_\perp^2 f_0(y, z),$$

$$g_J(y, z) \equiv \frac{\partial^2}{\partial x^2} \Delta u(x, y, z)|_{x=L_x} = e_J(y, z) - \nabla_\perp^2 f_J(y, z),$$

$$e_0(y, z) \equiv r(x = 0, y, z), \qquad e_J(y, z) \equiv r(x = L_x, y, z).$$

Inserting (29) into (28) yields a Poisson equation that *can* be solved by a transform method:

$$\nabla^2 u^* = r^* \equiv r - \left(1 - \frac{x}{L_x}\right) e_0 - \frac{x}{L_x} e_J - h_0 \nabla_\perp^2 g_0 - h_J \nabla_\perp^2 g_J. \quad (30)$$

The function f_0 and f_J in (29) serve to force u^* to be zero at $x = 0$ and $x = L_x$, and g_0 and g_J force r^* as well as $\partial^2 u^*/\partial x^2$ to be zero at the two ends. It is now natural to expand u^* in a sine series:

$$u^*(x, y, z) \cong \sum_{j=1}^{J-1} \bar{u}_j^*(y, z) \sin(\pi j x / L_x). \quad (31)$$

In general, $\partial^4 u^*/\partial x^4$ is not zero at the boundaries and thus sine functions only approximate the eigenfunctions. This approximation is equivalent to using Fourier functions to represent a periodic function that has discontinuities in the fourth derivative. This results in fourth-order (in Δx) errors near the discontinuities (i.e., boundaries) and fifth-order errors globally (in the L_2 norm). After expanding r^* in a series similar to (31) (in practice, by using a fast sine transform, Cooley *et al.* [22]), and using the results of Section 3.3 and the orthogonality properties of sine and Fourier series, (30) reduces to a set of ordinary differential equations,

$$\left(\frac{d^2}{dy^2} - p_j'^2 - q_n^2\right) \tilde{\bar{u}}_{jn}^*(y) = \tilde{\bar{r}}_{jn}^*(y), \qquad 1 \leqslant j \leqslant J-1,\ 0 \leqslant n \leqslant N/2 - 1, \quad (32)$$

where $\tilde{\tilde{u}}^*_{jn}(y)$ and $\tilde{\tilde{r}}^*_{jn}(y)$ are complex, $q_n = 2\pi n/L_z$, and p'^2_j is the effective wave number for the approximation of $\partial^2/\partial x^2$. Here, we use (14) with $\gamma = 10$:

$$p'^2_j = \frac{12}{\Delta x^2}(1 - \cos(\pi j/J))/(5 + \cos(\pi j/J)). \tag{33}$$

An effective wavenumber corresponding to a more accurate formula (e.g., by using $\gamma = 4$) does not seem to be warranted. It is tempting to replace (33) with the exact wavenumber, $p'_j = p_j \equiv \pi j/L_x$; however, this always seems to be less accurate. In one numerical test the use of p_j instead of (33) doubled the numerical error. Given first derivative approximations of a certain accuracy, the relationship between the accuracy of the Poisson equation inversion and the accuray of the final solution is not well understood.

Before we proceed with the solution of (32), we reduce it to its simplest form:

$$\left(\frac{d^2}{dy^2} - a\right) u(y) = s(y), \qquad a > 0. \tag{34}$$

As shown previously, the solution of (34) must satisfy the solvability conditions (24) so that the continuity equation (7) can be integrated for v. One way to ensure the satisfaction of (24) is to perform a Galerkin projection. The solution u is expanded in a special set of basis functions,

$$u(y) \cong \sum_{m=1}^{M-2} \hat{u}'_m \phi_m(\zeta), \tag{35}$$

where

$$\phi_m(\zeta) = \begin{cases} \cos(\pi(m-2)\zeta) - 2\cos(\pi m\zeta) + \cos(\pi(m+2)\zeta), & m \text{ even}, \\ \cos(\pi m\zeta) - \cos(\pi(m+2)\zeta), & m \text{ odd}. \end{cases}$$

Note that the number of coefficients is three less than in the normal expansion $u(y) \cong \sum_{m=0}^{M} \hat{u}_m \cos(\pi m\zeta)$, and that $\mathbf{u}$ can be reconstructed from *any* $\hat{\mathbf{u}}'$ and it will satisfy (24). The Galerkin method consists of substituting the appropriate expansions into (34), multiplying through by $\phi_n(\zeta)$, and integrating:

$$\sum_{m=1}^{M-2} \hat{u}'_m \int_0^1 \phi_n(\zeta)\left(\frac{d^2}{dy^2} - a\right)\phi_m(\zeta)\, d\zeta$$

$$= \int_0^1 \phi_n(\zeta)\, s(y)\, d\zeta, \qquad 1 \leqslant n \leqslant M-2. \tag{36}$$

In matrix form, this yields the system

$$\mathbf{D}(\mathbf{E} - a\mathbf{I})\,\mathbf{C}\hat{\mathbf{u}}' = \mathbf{D}\hat{\mathbf{s}}, \tag{37}$$

where $\mathbf{I}$ is the identity matrix of order $M+1$, $\mathbf{C}$ is the $(M+1)\times(M-2)$ matrix that converts a ϕ series into a cosine series, $\mathbf{E}=\mathbf{BA}$, and $\mathbf{D}$ is a $(M-2)\times(M+1)$ matrix of integrals:

$$D_{nm}=\int_0^1 \cos(\pi m\zeta)\,\phi_n(\zeta)\,d\zeta, \qquad 0\leqslant m\leqslant M,\ 1\leqslant n\leqslant M-2.$$

The columns of $\mathbf{C}$ are given by

$$(C_{m-2,m}, C_{m,m}, C_{m+2,m})=(1,-2,1), \qquad 2\leqslant m\leqslant M-2, \qquad m \text{ even},$$
$$(C_{m,m}, C_{m+2,m})=(1,-1), \qquad 1\leqslant m\leqslant M-3, \qquad m \text{ odd}.$$

Note that $\mathbf{D}$ is simply 0.5 times the transpose of $\mathbf{C}$ except for one element: $D_{10}=1$. Equation (37) can be inverted as is; however, this requires the solution of a nine-band matrix for the even modes and a seven-band matrix for the odd modes. It would be advantageous if the LHS of (37) could be factored into simpler matrices (since the amount of linear algebra required is proportional to the *square* of the bandwidth). The matrices $\mathbf{E}$ and $\mathbf{C}$ do not commute, but it turns out that it is possible to rewrite (37) as

$$\mathbf{DC}(\mathbf{E}'-a\mathbf{I})\,\hat{\mathbf{u}}'=\mathbf{D}\hat{\mathbf{s}}, \tag{38}$$

where $\mathbf{E}'$ has the same pentadiagonal structure as $\mathbf{E}$. The elements of $\mathbf{E}'$ are given in the Appendix. The solution of (34) is obtained in three steps:

$$(\mathbf{DC})\,\hat{\mathbf{s}}'=\mathbf{D}\hat{\mathbf{s}}, \tag{39a}$$
$$(\mathbf{E}'-a\mathbf{I})\,\hat{\mathbf{u}}'=\hat{\mathbf{s}}', \tag{39b}$$
$$\hat{\mathbf{u}}=\mathbf{C}\hat{\mathbf{u}}'. \tag{39c}$$

The cost of implementing (39a), (39b) is about 62% of the cost of (37). $\mathbf{DC}$ is a symmetric matrix of order $M-2$, which, from the above definitions, can also be defined by

$$(DC)_{nm}=\int_0^1 \phi_m(\zeta)\,\phi_n(\zeta)\,d\zeta, \qquad 1\leqslant m, n\leqslant M-2.$$

Thus (39a) is a projection of an arbitrary cosine expansion onto a ϕ expansion of the form (35). Since it is a Galerkin projection, the error is minimized in $L^2[0,1]$. Minimizing the error in another space would yield a different projection. The solution of (28) requires four additional steps: the construction of r^* (30) and its sine transform to wave space in x before (39) is carried out, and the transform of $\bar{u}^*$ back to physical space in x and the reconstruction of Δu (29) after (39).

The spanwise velocity w is subject to the same restraints as u so nearly the same procedure is used to solve (6). Since there are no x-derivatives on the LHS of (6),

this equation immediately reduces to an equation like (32) with $p'^2_j = 0$. Then (39) is used directly to find w, without transforming to wave space in the x direction.

3.5. *Time Advancement*

One member of a family of explicit third-order Runge–Kutta schemes discussed by Wray [23] is used for the advancement in time of (4) and (5). These schemes, applied to the model equation $\partial u/\partial t = f(u)$ are defined by

$$u^{(k+1)} = u^{(k)} + \Delta t(c_{k'} f(u^{(k)}) + d_{k'} f(u^{(k-1)})), \tag{40}$$

where k is the substep number, $k' = 1 + \mathrm{mod}(k, K)$ is the substep number within a time step (Δt) and K is the number of substeps in a time step. The particular set of parameters we use here are

$$\begin{aligned} &K = 3, \\ &c_1 = \tfrac{8}{15}, \qquad d_1 = 0, \\ &c_2 = \tfrac{5}{12}, \qquad d_2 = -\tfrac{17}{60}, \\ &c_3 = \tfrac{3}{4}, \qquad d_3 = -\tfrac{5}{12}. \end{aligned}$$

This scheme is compact in the sense that it requires the same storage and arithmetic as second-order Adams–Bashforth (defined by $K = 1$, $c_1 = 1.5$, $d_1 = -0.5$), but it is third-order accurate. Third-order Runge–Kutta schemes are stable for CFL numbers less than $\sqrt{3}$, while Adams–Bashforth schemes are unconditionally unstable for linear hyperbolic equations (although they can be stabilized by viscous terms). Also, the former is self-starting ($d_1 = 0$), while the latter requires a separate method for the first time step. This allows one to change Δt at the beginning of every time step (i.e., when $k' = 1$), and thus to keep the CFL number very chose to the maximum. For the Reynolds numbers we typically consider, the maximum time step due to the viscous stability restriction is about ten times larger than the CFL restriction. For the model equation $\partial u/\partial t = -\lambda u$, the former restriction is $\Delta t \leqslant 2.5/\lambda$.

After $\boldsymbol{\omega}$ is obtained from the derivatives of $\mathbf{u}$, both $\boldsymbol{\omega}$ and $\mathbf{u}$ are transformed to physical space using fast sine and cosine transforms (as appropriate) in y and fast real/half-complex Fourier transforms in z [22]. The nonlinear terms, $\mathbf{H}$, are then evaluated in physical space on a grid with M_c collocation points in the y direction and N_c points in the z direction. The multiplications needed for this step produce aliasing errors unless 50% more points are used in the collocation grid than in the original expansions (i.e., the "3/2 rule," $M_c = 1.5M$, etc.). However, if the Fourier expansions converge well enough (which they will for well-resolved simulations) then a large part of the aliasing error will be eliminated by using just 10 or 20% more collocation points than modes, with smaller additional reductions in error by using 50% more points. We typically use between 25 and 50% more collocation points than Fourier modes, depending on the amount of conservatism desired and which transform lengths are the most efficient numerically. In some cases full

dealiasing is necessary to exactly preserve in the solution symmetries present in the inflow boundary conditions.

The algorithm can be summarized briefly as follows. At the beginning of a time step (first substep), only u, ω_1 and the boundary conditions on the three velocity components are known. The RHS of (6) is evaluated and the 2D Poisson equation [illegible] RHSs of (4) and (5) are evaluated, the 3D Poisson equation for u is formed, and ω_1 is advanced in time. Finally, the Poisson equation is solved to yield the solution at the next substep. For the second and third substeps, the RHSs of (4) and (5) at the previous substep are saved so that (40) can be evaluated.

4. Analytical Test Problems

The various parts of the code are tested by solving three problems with analytical or independently-known solutions. Diffusion equations are solved in the first subsection, the nonlinear Stuart solution in the next subsection, and a 3D linear stability problem in the last subsection.

4.1. *3D Stokes Flow*

The time advancement, viscous, and Poisson sections of the code are tested by solving (1) and (2) with $\mathbf{H} = 0$. One possible solution to the resulting diffusion equations is

$$u(x, y, z, t) = \cos(x)\cos(z)\frac{y - y_0}{(1 + 4tv)^{3/2}} \times \exp(-2tv)\exp\left(-\frac{(y - y_0)^2}{1 + 4tv}\right), \tag{41a}$$

$$\omega_1(x, y, z, t) = \sin(x)\cos(z)\frac{1}{(1 + 4tv)^{1/2}} \times \exp(-2tv)\exp\left(-\frac{(y - y_0)^2}{1 + 4tv}\right), \tag{41b}$$

where $v = 1/\mathrm{Re}$ and $y_0 \neq 0$ breaks a symmetry which would have forced the even modes in (41a) and the odd modes in (41b) to be zero. Since the even and odd modes are treated differently, it is important that both are present in both equations. Boundary and initial conditions are extracted from (41), and we set $L_x = L_z = 2\pi$, $\mathrm{Re} = 10$, $\beta = 4$, $y_0 = 1$, $N = 4$, and $\Delta t = 0.005$. Shown in Fig. 4 are the max–norm errors of u and ω_1 as a function of J and M at $t = 1$. We use a large value of M (64) in Fig. 4a to try to isolate the x-differencing errors, and a large value of J (96) in Fig. 4b to isolate the effect of truncating the trigonometric series in y. The expected results are obtained: fourth-order convergence with Δx and

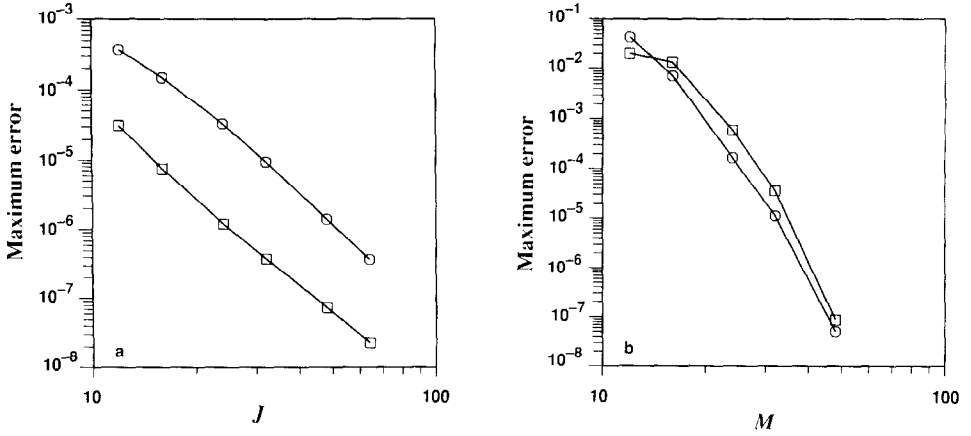

FIG. 4. Maximum error in u (○) and ω_1 (□) for the solution of a 3D Stokes problem as a function fo (a) number of grid points in x (with $M = 64$) and (b) number of Fourier modes in y (with $J = 96$).

(apparently) exponential convergence with M. Further tests show that time-differencing errors are negligible compared to the spatial approximations in all cases.

4.2. *2D Stuart Vortices*

A class of solutions to the 2D inviscid Navier–Stokes equations was given by Stuart [24]. One member of this class is defined by the streamfunction

$$\psi(x, y, t) = cy + \ln(a \cosh(y - y_0) + b \cos(x - ct)), \tag{42}$$

where $a = \sqrt{b^2 - 1}$, $u = \partial\psi/\partial y$, and $v = -\partial\psi/\partial x$. This solution has the appearance

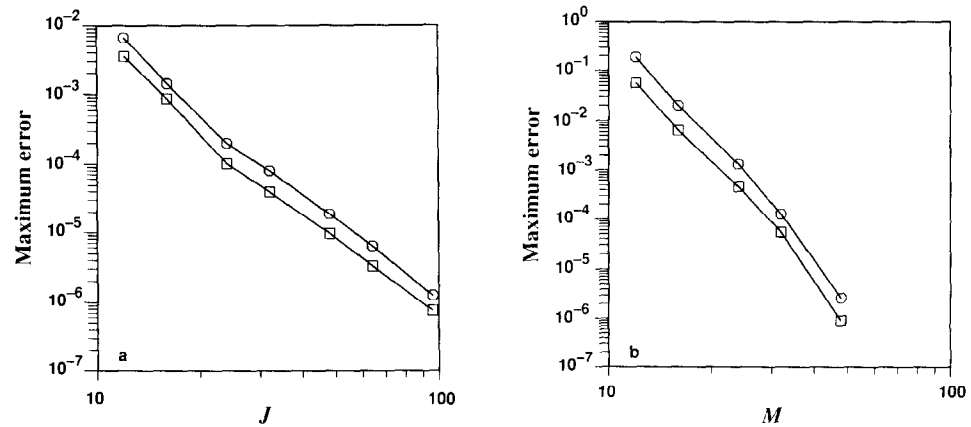

FIG. 5. Maximum error in u (○) and v (□) for the solution of a 2D Stuart vortices flow as a function of (a) number of x grid points (with $M = 64$) and (b) number of Fourier modes in y (with $J = 128$). Full dealiasing is used in all cases.

of a 2D mixing layer with vortices being advected downstream at a speed c. It is a good test of the formation of the nonlinear terms, the solution of the Poisson equation, and the time advancement. It is also a very good test of the overall stability of the algorithm, since neither physical nor numerical diffusion is present. As in the previous subsection, we are concerned mostly with the errors due to the x and y approximations. As before, time differencing errors are much smaller than the spatial error on the finest grid.

The parameters used for this test are $c = 1$, $a = \frac{1}{2}$, $L_x = 2\pi$, $\beta = 3$, $y_0 = 1$, and $\Delta t = 0.01$. In addition, $\mathrm{Re} = 10^9$ was used to make the simulation effectively inviscid. Shown in Fig. 5 are the max–norm errors of u and v as a function of J and M at $t = 1$. These calculations wre fully dealiased by using $M_c = 1.5M$. The two main sources of truncation error were isolated by using $M = 64$ in Fig. 5a and $J = 128$ in Fig. 5b. One sees fourth-order average convergence rates for the first case, and apparently eighth order (rather than exponential) convergence for the second case. The reason for the latter behavior is not known, but in any case it is certainly satisfactory. In Fig. 6, the effects of dealiasing are tested by comparing the results from Fig. 5b for u to those of calculations using $M_c = M$ (no dealiasing) and $M_c = 1.25M$ (partial dealiasing). For this somewhat unrealistic problem, the partial and full dealiasing simulations are almost identical, with the errors in both up to 10 times smaller than the non-dealiased simulations. As M increases all three tend towards each other faster than towards the exact solution, indicating that aliasing errors become less important for extremely well-resolved simulations. Of course, one is rarely in such a fortunate position.

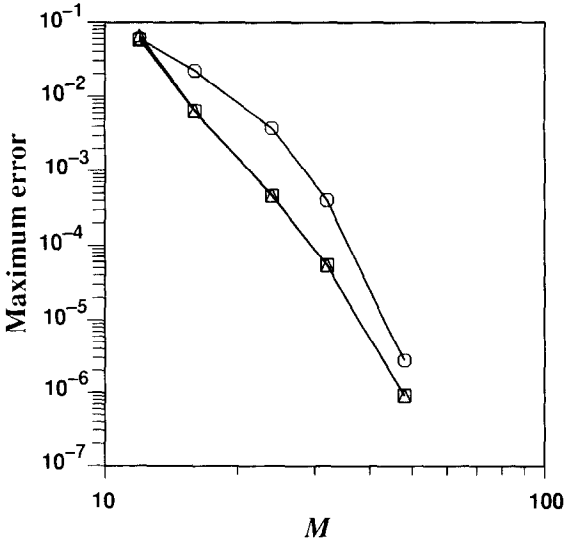

FIG. 6. Effect of the number of collocation points on the aliasing error for the u-component of the Stuart vortices flow: □, full dealiasing ($M_c = 1.5M$); △, partial dealiasing ($M_c = 1.25M$); and ○, no dealiasing ($M_c = M$).

4.3. *3D Linear Stability*

The final test consists of forcing the inflow with a small-amplitude oblique wave and comparing the growth rate with linear stability theory. Spatially-developing (i.e., the frequency is real) eigenfunctions and eigenvalues of the Rayleigh equation are obtained from Chen and Sandham [25] for a frequency of 0.58, an oblique angle of 30°, and a velocity ratio $\Lambda = 0.2$. The resulting growth rate and x-direction wavelength of the instability are 0.23351 and 8.31586, respectively. An amplitude of 10^{-6} at the inflow ensures linear growth for at least two wavelengths downstream. The numerical parameters used were $J = 192$, $M = 64$, $N = 4$, $\beta = 3$, $\Delta t = 0.18$, $\mathrm{Re} = 10^{10}$, $L_x = 60$, and $L_z = 14.403494$. The calculation was run until the instability wave approached the outflow boundary (one should avoid any feedback effect in this test). The average growth rate over $0 \leqslant x \leqslant 15$ and z of the fluctuations in u was found to be 0.23347. Convergence tests were inconclusive since the difference from the eigenvalue calculation, 4×10^{-5}, appears to be only partially due to the numerical approximations discussed earlier. Other contributions to this difference come from the residual of the transient (an error not present in temporal calculations), and errors in the eigenfunction calculation.

5. Mixing Layer Examples

For the present simulations, we used $V_e(y = -\infty) = 0.004$ and $V_e(y = \infty) = -0.01$. These entrainment velocities where chosen to minimize the streamwise pressure gradient outside of the shear layer (this is analogous to experiments where the walls of the wind tunnel are angled slightly). This was not done particularly accurately, but fortunately the flow does not appear to be very sensitive to the entrainment velocity at infinity. In the next two subsections we show results for 2D and 3D mixing layer flows.

5.1. *2D Flow*

Figure 7 shows contours of spanwise vorticity (ω_3) for a 2D mixing layer with $\mathrm{Re} = 200$ and $\Lambda = 0.2$. The numerical parameters used in this simulation were $J = 192$, $M = 128$, $M_c = 144$, $\beta = 8$, and $L_x = 70$. With these parameters the code requires 0.13 CPU seconds per substep on a Cray-XMP. The time step was adjusted every step to maintain the maximum CFL number in the domain at 1.7. Eigenfunctions of the Rayleigh equation are used to perturb the inflow profiles of u and v. The perturbation consists of two parts: a fundamental at a frequency of 0.5 and an amplitude of 0.01, and a subharmonic at half the frequency and amplitude. The former leads to the initial rollup and the latter to pairing of the "rollers" which starts at $x = 40$. Note that although there is no numerical diffusion, the contours are very smooth. This can be attributed to the accuracy of the first derivative approximations which produce dispersive errors small enough to be damped by the limited amount of physical diffusion available.

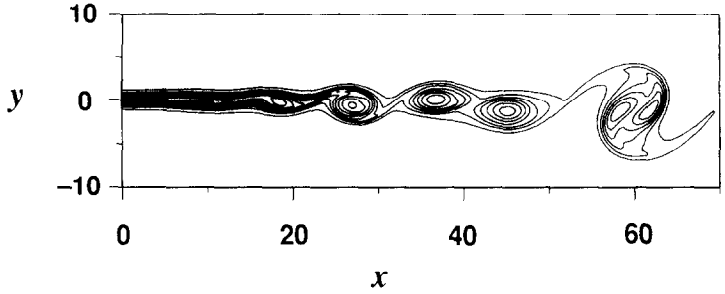

FIG. 7. Contours of $-\omega_3$ for a 2D mixing layer, Re = 200, $\Lambda = 0.2$. Contour interval is 0.1.

5.2. *3D Flow*

In this subsection we show what happens when counter-rotating streamwise vortices are added to the above 2D flow. Both theory [26, 27] and experiments [28] indicate that the "natural" or most amplified spanwise wavelength of the streamwise vortices is 60–70% of the 2D roller spacing (which is about 9.2 from Fig. 7). We chose a spanwise domain length of $L_z = 5.5$ and set the boundary conditions on v and w so that

$$\omega_1(x=0,\ y,\ z) = \frac{0.009}{k}(4y^2 - 2 - k^2)\, e^{-y^2} e^{ikz} + \text{c.c.},$$

where $k = 2\pi/L_z$ is the spanwise wavenumber. Because of the vortex stretching terms of the Navier–Stokes equations, smaller scales are created than in the 2D case, which lack such terms. Thus, a finer grid is needed. We found that the parameter set $J = 256$, $M = 144$, $M_c = 192$, $\beta = 8$, $N = 24$, and $N_c = 32$ works well. This simulation required about 7 CPU s per substep on a Cray-2.

Shown in Fig. 8 are contours of ω_3 in two of the x–y planes and contours of ω_1 in the plane midway between these two. The first two plots show the initial laminar layer, development of 2D rollers, and 3D cup-like structures. A comparison of these plots with Fig. 7 shows minimal differences up to $x = 25$ (ignoring the phase of the instability), but increasingly larger differences downstream. The third plot shows the increase of positive ω_1 in the braid regions between the main rollers and negative ω_1 within the rollers themselves. Note that the former (called "ribs") are of the same sign as the inflow forcing and the latter are of opposite sign. The ribs are intensified by the 2D strain field of the main rollers, but this mechanism cannot increase the circulation around a rib. However, since the lateral size of the ribs remains about constant, the circulation evidently increases significantly. The mechanism for this is the "conversion" of ω_3 into ω_1 by one of the "vortex stretching" terms of (2) (written in nonconservative form):

$$\frac{\partial \omega_1}{\partial t} = \cdots + \frac{\partial u}{\partial z}\omega_3 + \cdots.$$

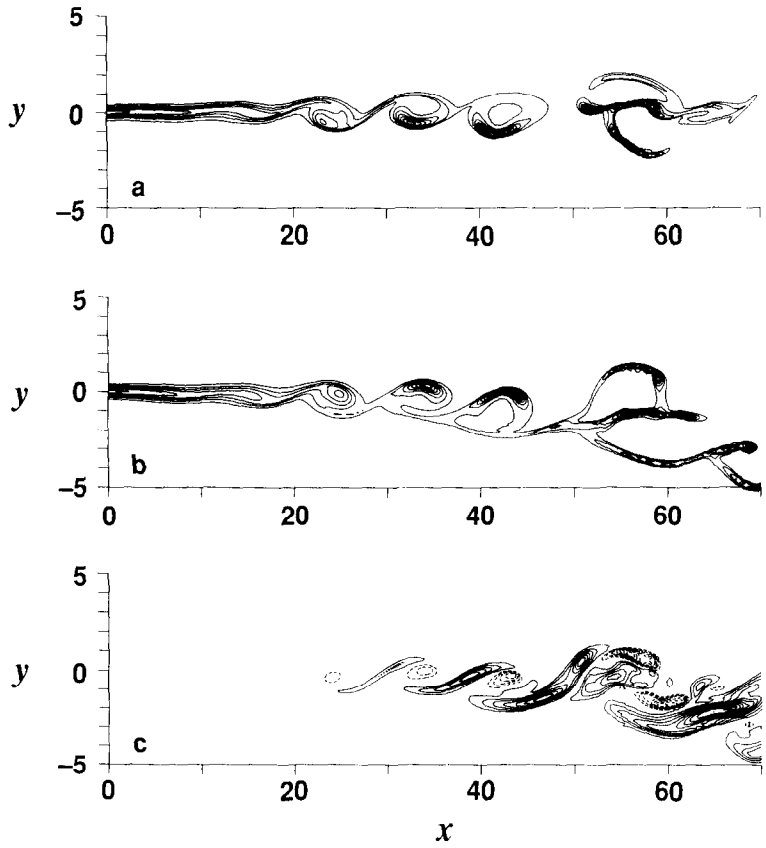

FIG. 8. Contours of vorticity for a 3D mixing layer, Re = 200, $\Lambda = 0.2$: (a) $-\omega_3$ at $z = 0$, (b) $-\omega_3$ at $z = 2.75$, (c) ω_1 at $z = 1.83$. Contour interval is 0.15.

The "cups" in the ω_3 plots are due to the local strain fields of the ribs and the distorted rollers. These strain fields are strong but very localized and cause extreme enhancement of ω_3 along the top or bottom (depending on z) of the rollers. In the present simulation the magnitude of the vorticity in the cups may reach 1.7 times that of the inflow. The opposite-sgned ω_1 in the middle of the rollers is also due to the 3D distortion of the rollers. Note that the cups associated with a roller are not at the same x location; the upper ones are displaced downstream from the lower ones. The vortex lines that connect the cups thus have an x component as well as y and z components. One can verify that the sign of ω_1 thus derived is consistent with Fig. 8c. The relative displacement of the cups and resulting ω_1 in the rollers are necessary to counteract the advection effects of the ribs and thus to prevent the cups from convecting away.

The four x–y cuts of ω_1 shown in Fig. 9 demonstrate the downstream development of the ribs. They start off as tilted elliptical vortices, where the aspect ratio

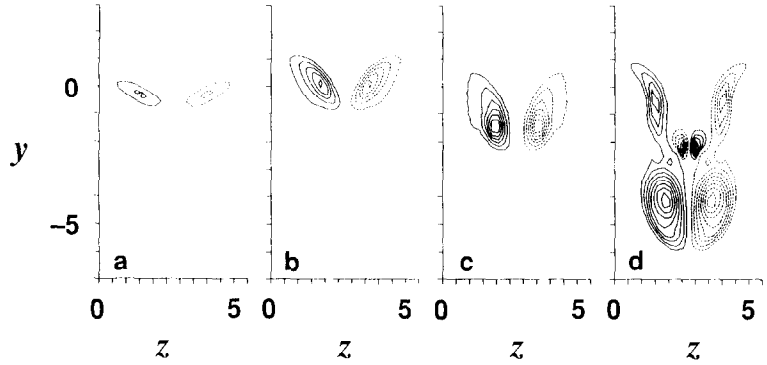

FIG. 9. Contours of ω_1 corresponding to Fig. 8 in y–z cuts at (a) $x = 28$, (b) $x = 38$, (c) $x = 48$, and (d) $x = 63$. Contour interval is 0.15.

and angle are determined by a balance between the effects of self-induction, 2D strain from the rollers, and viscosity. As the vortices gain strength through the mechanisms discussed above, the self-induction effect begins to dominate and the vortices "collapse" into nearly circular ribs [29]. The fourth plot is a cut downstream of a pairing of the rollers, but little direct effect of the pairing is seen on the ribs. However, there are many other possibilities for the inflow boundary conditions which may produce qualitatively different flows. These issues will be investigated in more detail in later papers.

6. CONCLUSION

We have presented here an accurate numerical algorithm for the solution of the 3D incompressible Navier–Stokes equations. The method is based on compact Runge–Kutta time differencing and three different methods for the three spatial directions. The Padé finite difference formulas of Lele [16, 17] with new modifications at the boundaries are used for derivative approximations in the streamwise direction, a new version of the spectral scheme of Cain *et al.* [5] is used in the (infinite) vertical direction, and a classical Fourier method is used in the homogeneous spanwise direction. For finite difference methods we emphasize the accuracy attained for moderate- to high-frequency functions, rather than the order of accuracy for asymptotically low-frequency functions. In addition, a Galerkin projection procedure is developed for the solution of the Poisson equation. This procedure ensures the satisfaction of the continuity and boundary condition constraints on the velocity field, but does not require the solution of a large sparse matrix. The results from 2D and 3D mixing layer simulations show the ability of the method to resolve strong gradients and complex solutions.

APPENDIX

The elements of the tridiagonal matrices **A** and **B** which define the numerical first derivatives of cosine and sines series, respectively, in the y direction are

$$\begin{aligned} A_{m,m-2} &= -B_{m,m-2} = (m-2)/(4\beta), \\ A_{m,m} &= -B_{m,m} \quad = -m/(2\beta) \\ A_{m,m+2} &= -B_{m,m+2} = (m+2)/(4\beta), \end{aligned}$$

where $0 \leqslant m \leqslant M$. Ignoring elements outside the tridiagonal matrix structure, the exceptions to the above are

$$\begin{aligned} A_{0,2} &= 0, \\ A_{1,1} &= -3/(4\beta), \\ B_{1,1} &= 1/(4\beta), \\ A_{M-1,M-1} = A_{M,M} &= A_{M-1,M-3} = A_{M,M-2} = 0, \\ B_{M-1,M-1} = B_{M,M} &= B_{M-3,M-1} = B_{M-2,M} = 0. \end{aligned}$$

Equation (35) defines a pentadiagonal matrix $\mathbf{E}'$ of order $M-2$. The elements are

$$\begin{aligned} E'_{m,m-4} &= -cm(m-2), \\ E'_{m,m-2} &= \begin{cases} 4cm^2, & m \text{ odd}, \\ 4cm(m-1), & m \text{ even}, \end{cases} \\ E'_{m,m} &= \begin{cases} -2c(3m^2+6m+4), & m \text{ odd}, \\ -6cm^2, & m \text{ even}, \end{cases} \\ E'_{m,m+2} &= \begin{cases} 4c(m+2)^2, & m \text{ odd}, \\ 4cm(m+1), & m \text{ even}, \end{cases} \\ E'_{m,m+4} &= \begin{cases} -c(m+2)(m+4), & m \text{ odd}, \\ -cm(m+2), & m \text{ even}, \end{cases} \end{aligned}$$

where $c = 1/(16\beta^2)$ and $1 \leqslant m \leqslant M-2$. Ignoring matrix elements outside this range, the exceptions to the above are

$$\begin{aligned} E'_{11} &= -27c, \\ E'_{M-3,M-5} &= c(M-3)(3M-11), \\ E'_{M-3,M-3} &= -c(M-3)(3M-7). \end{aligned}$$

ACKNOWLEDGMENTS

We gratefully acknowledge numerous discussions with Robert Moser, Michael Rogers, Sanjiva Lele, and Nagi Mansour. S. Lele kindly gave permission to adapt Figs. 2 and 3 from Ref. [17] before publication. Part of this research was carried out while the author held a National Research Council postdoctoral Associateship.

REFERENCES

1. P. COMTE, M. LESIEUR, H. LAROCHE, AND X. NORMAND, in *Turbulent Shear Flows VI* (Springer-Verlag, New York, 1987).
2. J. KIM AND P. MOIN, *J. Comput. Phys.* **59**, 308 (1985).
3. R. W. DAVIS AND E. F. MOORE, *Phys. Fluids* **28**, 1626 (1985).
4. P. S. LOWERY, Thesis, Stanford University, 1986 (unpublished).
5. A. B. CAIN, J. H. FERZIGER, AND W. C. REYNOLDS, *J. Comput. Phys.* **56**, 272 (1984).
6. D. GOTTLIEB AND S. A. ORSZAG, *Numerical Analysis of Spectral Methods: Theory and Applications*, CBMS-NSF Monograph, No. 26 (Soc. Indus. Appl. Math., Philadelphia, 1977).
7. J. C. BUELL, *J. Comput. Phys.* **75**, 54 (1988).
8. L. S. TUCKERMAN, *J. Comput. Phys.* **80**, 403 (1989).
9. J. W. MURDOCK, *AIAA Paper* 86-0434, 1986.
10. J. C. BUELL AND I. CATTON, *J. Heat Trans.* **105**, 255 (1983).
11. H. FRICK, F. H. BUSSE, AND R. M. CLEVER, *J. Fluid Mech.* **127**, 141 (1983).
12. J. KIM, P. MOIN, AND R. MOSER, *J. Fluid Mech.* **177**, 133 (1987).
13. M. GASTER, *J. Fluid Mech.* **14**, 222 (1962).
14. J. C. BUELL AND P. HUERRE, in *Proceedings of the* 1988 *Summer Program, Center for Turbulence Research*, 1988, p. 19.
15. P. HUERRE AND P. A. MONKEWITZ, *J. Fluid Mech.* **159**, 151 (1985).
16. S. K. LELE, *AIAA Paper* **89-0374** (1989).
17. S. K. LELE, *J. Comput. Phys.*, submitted.
18. C. E. GROSCH AND S. A. ORSZAG, *J. Comput. Phys.* **25**, 273 (1977).
19. R. W. METCALFE, S. A. ORSZAG, M. E. BRACHET, S. MENON, AND J. J. RILEY, *J. Fluid Mech.* **184**, 207 (1987).
20. P. R. SPALART, "Numerical Simulation of Boundary Layers: Part 1. Weak Formulation and Numerical Method," NASA TM-88222, 1986 (unpublished).
21. M. M. ROGERS AND R. D. MOSER, in *Proceedings, Seventh Symposium on Turbulent Shear Flows, Stanford, CA*, 1989, p.9.3.1.
22. J. W. COOLEY, P. A. W. LEWIS, AND P. D. WELCH, *J. Sound Vib.* **12**, 315 (1970).
23. A. A. WRAY, Minimal storage time-advancement schemes for spectral methods, *J. Comput. Phys.*, submitted.
24. J. T. STUART, *J. Fluid Mech.* **29**, 417 (1967).
25. J. H. CHEN AND N. SANDHAM, private communication (1988).
26. R. T. PIERREHUMBERT AND S. E. WIDNALL, *J. Fluid Mech.* **114**, 59 (1982).
27. C. M. HO, Y. ZOHAR, R. D. MOSER, M. M. ROGERS, S. K. LELE, AND J. C. BUELL, in *Proceedings*, 1988 *Summer Program, Center for Turbulence Research*, 1988, p. 29.
28. L. S. HUANG AND C. M. HO, Small-scale transition in plane mixing layers, *J. Fluid Mech.*, submitted.
29. S. J. LIN AND G. M. CORCOS, *J. Fluid Mech.* **141**, 139 (1984).