Journal of Computational Physics **180,** 200–228 (2002)
doi:10.1006/jcph.2002.7088

Direct Numerical Simulation of Spatial Transition to Turbulence Using Fourth-Order Vertical Velocity Second-Order Vertical Vorticity Formulation

Kiran Bhaganagar, Dietmar Rempfer,[1] and John Lumley

Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, New York 14853
E-mail: kiran@mae.cornell.edu, dr36@cornell.edu, jll4@cornell.edu

Received August 24, 2000; revised January 25, 2002

A highly accurate algorithm has been developed to study the process of spatial transition to turbulence. The algorithmic details of the direct numerical simulation (DNS) of transition to turbulence in a boundary layer based on a formulation in terms of vertical velocity and vertical vorticity are presented. Issues concerning the boundary conditions are discussed. The linear viscous terms are discretized using an implicit Crank–Nicholson scheme, and a low-storage Runge–Kutta method is used for the nonlinear terms. For the spatial discretization, fourth-order compact finite differences have been used, as these have been found to have better resolution compared to explicit differencing schemes of comparable order. The number of grid points that are needed per wavelength is close to the theoretical optimum for any numerical scheme. The resulting time-discretized fourth-order equations are split up into two second-order equations, resulting in Helmholtz- and Poisson-type equations. The boundary conditions for the Laplacian of the vertical velocity are determined using an influence matrix method. A robust multigrid algorithm has been developed to solve the resulting anisotropic elliptical equations. For the outflow boundary, a buffer domain method, which smoothly reduces the disturbances to zero, in conjunction with parabolization of the Navier–Stokes equations has been used. The validation of the results for the DNS solver is made both for linear and weakly nonlinear cases.

Key Words: DNS; spatial transition; boundary conditions; 3D simulation; NS equations; code validation.

[1] New address: Illinois Institute of Technology, Chicago, Illinois 60616. E-mail: Rempfer@iit.edu.

0021-9991/02 $35.00

1. INTRODUCTION

The dynamics of transition to turbulence is still an unsolved fundamental problem of physics [18]. A good understanding of the pertinent mechanisms of transition is necessary to reliably predict and possibly control turbulence. As a step toward this goal, a simple configuration, the spatially evolving flat-plate boundary layer will be investigated. A particularly interesting feature of this flow is constituted by the fact that this is a system that shows a continuous evolution from a simple laminar and periodic state into a complex, chaotic, and ultimately turbulent condition. It is also interesting to note that in contrast to this continuous succession of flow states, our current knowledge of such flows shows a fairly distinct schism into two bodies, one dealing with the early and typically at most weakly nonlinear stages of transition, the subject of transition research [17, 22, 23, 30], and the other one concerned with the flow only after it has reached a fully developed (and preferably high-Reynolds-number) turbulent state [24]. Robust and highly accurate direct numerical methods, without approximation, appear to be a promising approach to shed more light on the pertinent mechanisms during the late stages of transition and thus bridge the gap of our understanding between transition and turbulence.

We have developed a parallel transition code that can accurately describe the flow dynamics from the early stages of transition to a fully developed turbulent state. The stages involved are receptivity, the stage where the boundary-layer instabilities known as Tollmien–Schlichting (TS) waves are generated, followed by their amplification first in the linear regime, then as amplitudes reach considerable values, a phase of nonlinear breakdown followed by the final transition to the turbulent state. In this paper, the numerical aspects of the code development are discussed along with simulation results for the linear and weakly nonlinear regime. The spatial transition to turbulence for the flat-plate boundary layer, with no pressure gradient, has been simulated for an inflow Reynolds number of 650 with respect to the displacement thickness. The size of the integration domain is on the order of 20 TS wavelengths in the streamwise direction and 14 displacement thicknesses in the wall–normal direction. Streamwise and normal directions are inhomogeneous; the spanwise coordinate is periodic.

The code has been thoroughly validated to test both that the correct equations are solved and that the equations are solved correctly. For the former, the growth rates and mode profiles for small disturbances are compared with the solutions of the Orr–Sommerfeld equation; for the latter, tests of Poisson and Helmholtz solvers using analytical solutions have been performed.

As a first step, using a separate code a steady Navier–Stokes (NS) solution is generated, which is used as a base flow to simulate the spatial transition to turbulence. This base flow is initially perturbed with small disturbances, which gradually evolve through the stages of linear regime, nonlinear regime, and finally breakdown to turbulence. The disturbances added are of the form of blowing and suction at the wall [19]. Thus our actual unsteady Navier–Stokes code simulates the behavior of disturbances to the steady Navier–Stokes solution. This approach of splitting the flow into a steady Navier–Stokes solution plus disturbances where the evolution in time is subsequently simulated is known as a "disturbance-flow formulation." A vertical velocity–vertical vorticity formulation [25] has been used. This formulation obviates the determination of the pressure boundary condition. We obtain a vertical vorticity equation by taking the curl of the Navier–Stokes equations, and we take the curl of the curl of the Navier–Stokes equation resulting in a fourth-order vertical

velocity equation. Thus, two equations, one fourth-order equation for vertical velocity v, and one second-order equation for vertical vorticity ω_2 are solved, and the other flow variables, streamwise velocity u, transverse velocity w, streamwise vorticity ω_1, and transverse vorticity ω_3 are obtained from the divergence-free conditions and vorticity definition. It is computationally efficient to split the fourth-order equation for v into two second-order equations and solve the resulting system. To avoid stringent time step restrictions, a semi-implicit time integration scheme has been implemented, using a fourth-order low-storage Runge–Kutta (RK) scheme for the nonlinear terms, and a Crank–Nicholson implicit scheme for the linear viscous terms. The resulting time-discretized equations are of elliptical nature and solved using a multigrid method. Due to the presence of strong anisotropy, a line Gauss–Seidel algorithm has been used as a smoother. The multigrid solvers that have been developed have been tested.

The waves in the inhomogeneous x- and y-directions need to be well resolved; hence high-resolution compact finite differences have been used to obtain the required spatial derivatives. For finite differences the low-wavenumber components are well resolved but the high-wavenumber parts are not well resolved, so to get higher resolution higher order finite differences need to be used [3]. Alternatively, without increasing the formal order of accuracy, spectral-like resolution can be achieved using compact finite differences (for details see [20]). These result in sparse banded matrix systems the solution of which needs $O(N + Np^2)$ operations [12], where N is the size of the matrix and p is the bandwidth. To obtain a spatial derivative of fourth-order accuracy, the bandwidth of the systems to be solved to obtain the derivatives is generally one. In the x- and y-directions, a fourth-order compact finite-difference scheme has been used along with a Fourier representation in the spanwise direction due to the homogeneity of the flow.

The organization of the paper is as follows. Section 2 discusses the mathematical model for the transition problem, Section 3 presents the multigrid methods which have been used to solve the resulting system of equations, Section 4 shows the implementation of the compact difference schemes for describing the resulting Poisson and Helmholtz equations, and Section 5 describes the spatial linear stability theory used as a validation tool for the first regime of the transition process, where the growth rate of the disturbances exhibits a linear behavior. Section 6 presents the semi-implicit time integration scheme. As the fourth-order vertical velocity equation is split into two equations, boundary conditions for the Laplacian of velocity are required, which are obtained by an influence matrix method discussed in Section 7. Transition is initiated by perturbing a base flow, so a steady Navier–Stokes solution is required for the initial condition, which is discussed in Section 8. Outflow boundary conditions have been selected carefully as described in Section 9. Section 10 presents the results and discussion.

2. GOVERNING EQUATIONS

We nondimensionalize our variables according to

$$x = \tilde{x}/L, \qquad y = \tilde{y}\sqrt{\mathrm{Re}}/L, \qquad z = \tilde{z}/L, \quad t = \tilde{t}U_\infty/L \tag{1}$$

$$u = \tilde{u}/U_\infty, \qquad v = \tilde{v}\sqrt{\mathrm{Re}}/U_\infty, \qquad w = \tilde{w}/U_\infty \tag{2}$$

$$\omega_1 = \frac{\partial \tilde{v}}{\partial \tilde{z}} - \frac{\partial \tilde{w}}{\partial \tilde{y}}, \quad \omega_2 = \frac{1}{\mathrm{Re}}\frac{\partial \tilde{w}}{\partial \tilde{x}} - \frac{\partial \tilde{u}}{\partial \tilde{z}}, \quad \tilde{\omega}_3 = \frac{\partial \tilde{u}}{\partial \tilde{y}} - \frac{1}{\mathrm{Re}}\frac{\partial \tilde{v}}{\partial \tilde{x}}. \tag{3}$$

Here $(\tilde{})$ refers to the dimensional physical quantities. U_∞ is the free-stream velocity, and L is some arbitrary length scale.

The governing equations can be written in the vertical velocity–vorticity form [25]. The advantages of this formulation are that the pressure term is eliminated, and only two governing equations need to be solved, reducing the storage requirements compared to a primitive-variables formulation or a vorticity-transport formulation.

The vertical component of the curl of the Navier–Stokes equations and the curl of the curl of the NS equations give the vertical vorticity and vertical velocity equations, respectively,

$$\frac{\partial \omega_2}{\partial t} = H_\omega + \Delta\omega_2, \tag{4}$$

$$\frac{\partial \Delta v}{\partial t} = H_v + \Delta^2 v, \tag{5}$$

where

$$H_v = -\frac{\partial}{\partial y}\left[\frac{\partial H_1}{\partial x} + \frac{\partial H_3}{\partial z}\right] + \left[\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial z^2}\right] H_2, \tag{6}$$

$$H_\omega = \left(\frac{\partial H_3}{\partial x} + \frac{\partial H_1}{\partial z}\right), \tag{7}$$

$$H = (H_1, H_2, H_3) = u \times \omega, \tag{8}$$

$$\Delta = \left[\epsilon\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \epsilon\frac{\partial^2}{\partial z^2}\right], \tag{9}$$

with $\epsilon = \frac{1}{\mathrm{Re}}$. For numerical simplicity the fourth-order vertical velocity Eq. (5) is split into two second-order equations as

$$\frac{\partial \phi}{\partial t} = H_v + \Delta^2 v, \tag{10}$$

$$\Delta v = \phi. \tag{11}$$

Equation (10) is the evolution equation for the Laplacian of the vertical velocity ϕ and Eq. (11) is the Poisson equation for v.

Semi-implicit time integration is performed using a low-storage Runge–Kutta scheme [29] for the nonlinear terms and implicit Crank–Nicholson for the linear terms. The time discretized equations (shown in detail in Section 6) resulting from (4) at the Runge–Kutta time step $(n+1)$ are

$$\left(I - \frac{dt(A_n + B_n)}{2}\Delta\right)\omega_2^{n+1} = A_n\omega^n + B_n\omega^{n-1} + \left(I + \frac{dt(A_n + B_n)}{2}\Delta\right)\omega_2^n, \tag{12}$$

and from (5), which is split into (10) and (11) we obtain

$$\left(I - \frac{dt(A_n + B_n)}{2}\Delta\right)\phi^{n+1} = A_n H_v^n + B_n H_v^{n-1} + \left(I + \frac{dt(A_n + B_n)}{2}\Delta\right)\phi^n, \tag{13}$$

$$\Delta v^{n+1} = \phi^{n+1}, \tag{14}$$

where A_n and B_n are the Runge–Kutta coefficients. These time-discretized equations result in Helmholtz- and Poisson-type equations which are solved using a multigrid method, discussed in the next section. The boundary conditions that have been prescribed are as follows. At the wall, disturbances in the form of blowing and suction are introduced (see A.2). At the free-stream boundary exponentially decaying disturbances are assumed, resulting in Robins-type boundary conditions for the vertical velocity,

$$(v' + \alpha v)|_{y=N} = 0, \quad (v'' - \alpha^2 v)|_{y=N} = 0, \tag{15}$$

$$v|_{y=0} = 0, \qquad v'|_{y=0} = 0, \tag{16}$$

$$\omega_2|_{y=0} = 0 \qquad \omega_2|_{y=N} = 0, \tag{17}$$

where α is the streamwise wavenumber at the inlet obtained from linear stability theory.

The instantaneous velocity or total velocity vector $\mathbf{V}(x, y, z, t) = [u, v, w]$ can be expressed in the form

$$\mathbf{V} = \mathbf{V}_{\text{base}} + \mathbf{V}', \tag{18}$$

where $\mathbf{V}_{\text{base}}(x, y) = [U_{\text{base}}, V_{\text{base}}, 0]$ is the base flow and $\mathbf{V}'(x, y, z, t) = [u', v', w']$ is the disturbance velocity vector. In our unsteady simulations, we are solving the Navier–Stokes equations for the disturbances, which are obtained by subtracting the steady Navier–Stokes equations with solution $\mathbf{V}_{\text{base}}$ from the unsteady Navier–Stokes equations for the total flow velocity $\mathbf{V}$. The main reason for using the resulting "disturbance formulation"—rather than a "total velocity formulation" that would solve directly for $\mathbf{V}$ in one step—is that it is easier to prescribe disturbance boundary conditions than to find good boundary conditions for the total velocity.

For the base flow $\mathbf{V}_{\text{base}}$, a steady Navier–Stokes solution is computed using a code which employs the same set of governing equations but in a total velocity formulation rather than the disturbance formulation discussed above. The outflow boundary conditions will be discussed in detail in Section 4.

3. MULTIGRID SOLVER

To solve the discretized elliptic equations, a multigrid solver has been selected as the method of choice. Multigrid methods [28] are based on the idea of eliminating the low-wavenumber errors at each grid level, while the unsmoothed high-wavenumber errors are transferred to coarser grids so that high-wavenumber errors on the smooth grid become low-wavenumber errors on the coarse grid and can be eliminated at this level. Thus a series of grids will successively eliminate the errors. In a typical multigrid V-cycle the residue is restricted from the finest to the coarsest grid, and then the error is interpolated back to the fine grid. Multigrid is efficient if all the frequencies which are not represented on the coarser grid can be smoothed effectively. This method is also fast and generally two to three V-cycles are sufficient to solve the system. Further, it is an optimal method as the convergence rate is independent of the number of unknowns. In contrast, for single-grid solvers, the convergence is initially very fast, but it slows eventually; the finer the grid, the slower the asymptotic convergence [1]. The multigrid terminology that will be used is as follows. The finest grid is the highest order grid and the coarsest grid is the first grid. The

interpolation from coarse to fine grid is referred to as the *prolongation operation* and from fine to coarse is the *restriction operation*. For second-order equations linear interpolation has proven to be sufficient, in general for an equation of order m an interpolation scheme of order at least $2m-2$ is required. The smoothing factor is the maximum damping of a given scheme for all modes which are not represented on coarser grids. The general algorithm for multigrid solvers is given by

ALGORITHM 1.
Fine grid level Solve $v^h \leftarrow S^v v^h$,
Restrict residue on to coarse grid $r^{2h} = I_h^{2h} r^h$,
Solve error equation $L^{2h} e^{2h} = r^{2h}$,
Prolongate e^{2h} onto h level & correct $v^h \leftarrow v^h + I_{2h}^h e^{2h}$,
Relax v times on the fine grid $v^h \leftarrow S^v v^h$.

Here S is the y-line Gauss–Seidel smoother, I is the restriction/prolongation operator (between the grids of mesh size h and $2h$), L is the discretized Poisson or Helmholtz operator, v refers to the number of presmoothers, r represents the residue of the equation, e is the error of the solution, and superscripts denote the multigrid level. For the present case of a boundary-layer-type problem, the mesh is strongly anisotropic and the extent of this anisotropy increases with Reynolds number. For such cases, when a Gauss–Seidel scheme is used as a smoother, the convergence rate becomes very slow as ϵ decreases, where ϵ is the measure of anisotropy. As shown in the previous section, our Poisson and Helmholtz equations (Eqs. (12)–(14)), using the definition of Laplacian as [9], are of the form

$$\epsilon \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \epsilon \frac{\partial^2}{\partial z^2} = f(x, y, z), \tag{19}$$

$$\left[I - \frac{1}{\mathrm{Re}} \frac{dt}{2} \left(\epsilon \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \epsilon \frac{\partial^2}{\partial z^2} \right) \right] \phi = g(x, y, z), \tag{20}$$

where $f(x, y, z)$ and $g(x, y, z)$ are the right-hand sides of the Poisson and Helmholtz equations, respectively, and I is the identity matrix. When a Gauss–Seidel scheme is used as a smoother, the smoothing rate is poor due to the fact that as $\epsilon \to 0$, it smooths only in the y-direction and an error which is smooth in y and highly oscillatory in x is present. A good remedy is to use a Gauss–Seidel y-line algorithm. Here as $\epsilon \to 0$ the equations are in fact solved exactly. The convergence does not deteriorate but improves with decreasing ϵ, and the smoothing rate improves. The derivatives for both the Poisson and Helmholtz solvers have been discretized using a fourth-order compact difference scheme [13], discussed in detail in the next section. First the Helmholtz equation is solved and ϕ obtained, then the Poisson equation is solved to obtain v. The Helmholtz equation for vertical vorticity is solved to obtain the normal component of vorticity. All the other flow variables are obtained from the definition of vorticity and the continuity equation.

4. IMPLEMENTATION OF COMPACT DIFFERENCES FOR POISSON AND HELMHOLTZ SOLVERS

In the compact finite-difference implementation, the values of the derivatives on a set of nodes are expressed as a linear combination of the function values [20]. For a uniformly spaced mesh in the xy-direction, let the index of a node where the derivative needs to be

evaluated be represented by (i, j) along the x- and y-directions, and let (di, dj) denote the corresponding mesh increments at node (i, j). At the node (i, j) the x-derivative is $D^x v_{i,j}$ and the y-derivative is $D^y v_{i,j}$. We use D^x and D^y to represent the second derivatives, $(\frac{\partial^2}{\partial x^2})$ and $(\frac{\partial^2}{\partial y^2})$, respectively. A compact finite-difference scheme for the second derivative is given as

$$x\text{-derivative:} \quad \sum_{di} \alpha_{di} D^x v_{i+di,j} = \sum_{di} a_{di} v_{i+di,j}, \tag{21}$$

$$y\text{-derivative:} \quad \sum_{dj} \beta_{dj} D^y v_{i,j+dj} = \sum_{di} b_{dj} v_{i,j+dj}, \tag{22}$$

where the coefficients $\alpha_{di}, \beta_{dj}, a_{di}, b_{dj}$ are obtained by matching the Taylor series coefficients of various orders.

If (21) and (22) are written for each node of the grid, a linear system of equations, $Pf^{(d)} = Qf$, can be obtained, where d is the order of the derivative, and P, Q are the corresponding banded matrices representing the coefficients of the derivative and function values.

Now, we use the above compact difference representation to express the derivatives in the Poisson equation (9, 14); i.e.,

$$\left(\epsilon \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \epsilon \frac{\partial^2}{\partial z^2}\right) v = \phi. \tag{23}$$

In the present simulation due to the presence of periodicity in the z-direction, a Fourier representation has been used leading to

$$\frac{\partial^2}{\partial z^2} \hat{v} = K^2 \hat{v}, \tag{24}$$

where K is the wavenumber and $\hat{v}$ is the Fourier transform in z of v. Thus (23) can be written as

$$(\epsilon D^x + D^y - \epsilon K^2)\hat{v} = \phi. \tag{25}$$

We premultiply (21) by $\epsilon\beta_{dj}$ and (22) by α_{di}, and sum with respect to dj and di, respectively, which gives

$$\epsilon \sum_{dj} \sum_{dj} \beta_{dj} \alpha_{di} D^x \hat{v}_{i+di,j+dj} = \epsilon \sum_{dj} \sum_{di} \beta_{dj} a_{di} \hat{v}_{i+di,j+dj}, \tag{26}$$

$$\sum_{di} \sum_{dj} \alpha_{di} \beta_{dj} D^y \hat{v}_{i+di,j+dj} = \sum_{di} \sum_{dj} \alpha_{di} b_{dj} \hat{v}_{i+di,j+dj}. \tag{27}$$

Adding the above two equations and adding $-\epsilon K^2 \sum_{di} \sum_{dj} \alpha_{di} \beta_{dj} \hat{v}_{i+di,j+dj}$ to both left-hand and right-hand sides gives

$$\sum_{di} \sum_{dj} \alpha_{di} \beta_{dj} (\epsilon D^x + D^y - \epsilon K^2)\hat{v}_{i+di,j+dj}$$

$$= \sum_{di} \sum_{dj} [\epsilon \beta_{dj} a_{di} + b_{dj} \alpha_{di} - \epsilon \alpha_{di} \beta_{dj}] \hat{v}_{i+di,j+dj}. \tag{28}$$

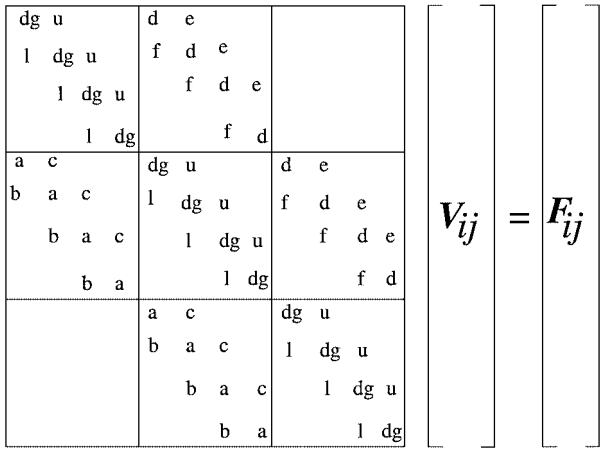

FIG. 1. Resultant block matrix system obtained to evaluate $\hat{v}$ using a compact finite-difference scheme.

Let $(\epsilon D^x + D^y - \epsilon K^2)\hat{v}_{i+di,j+dj} = \phi_{i+di,j+dj}$. Then the above equation can be written as

$$\sum_{di}\sum_{dj} \alpha_{di}\beta_{dj}\phi_{i+di,j+dj} = \sum_{di}\sum_{dj} [\beta_{dj}a_{di} + b_{dj}\alpha_{di}]\hat{v}_{i+di,j+dj}. \tag{29}$$

The system of equations given by (29) can be written for each node and a resultant linear system of equations is obtained. In the present case, to obtain the resultant system of fourth-order accuracy, di and dj take on the values $-1, 0, 1$, so we obtain a block-banded system as shown in Fig. 1. F_{ij} is the left-hand side of (29); as the values of ϕ are known, F_{ij} can be evaluated. Solving this system gives the unknown vector v. To solve this block-banded matrix system, a multigrid solver has been used (details are given in Section 3).

Using (29), we obtain the elements of the banded matrix in Fig. 1 as

$$\begin{bmatrix} c & a & b \\ l & dg & u \\ f & d & e \end{bmatrix} = \begin{bmatrix} \alpha_{-1}\beta_{-1} & \alpha_{-1}\beta_0 & \alpha_{-1}\beta_1 \\ \alpha_0\beta_{-1} & \alpha_0\beta_0 & \alpha_0\beta_1 \\ \alpha_1\beta_{-1} & \alpha_1\beta_0 & \alpha_1\beta_1 \end{bmatrix}. \tag{30}$$

Similarly, the Helmholtz equation (13), which is of the form (20), can be solved for ϕ. The elements of the *block-banded matrices* are given in the Appendix. Further, the numerical boundary conditions should be selected such that the resulting system is asymptotically stable (see [4] for details on the stability issues of the numerical boundary). For the present case third-order compact derivatives have been selected on the boundary, so that the resulting system is still of fourth-order [5].

5. LINEAR STABILITY THEORY: SPATIAL APPROACH

Linear stability theory is used to analyze the stability of a given base flow subjected to small disturbances. These disturbances can be described as a linear superposition of terms of the normal mode form

$$[u, v, w]^T = [\hat{u}(y), \hat{v}(y), \hat{w}(y)]^T \exp(i(\alpha x + \beta z - \omega t)), \tag{31}$$

where the real parts of α and β represent the x- and z-components of the wavenumber vector $\vec{k}$, ω is the frequency, and $\hat{u}(y)$, $\hat{v}(y)$, $\hat{w}(y)$ are the complex eigenfunctions representing the wall–normal structure of the mode. For the case of spatial theory [10], the frequency ω is real and the wavenumbers α and β are complex,

$$\alpha = \alpha_r + i\alpha_i, \quad \beta = \beta_r + i\beta_i. \tag{32}$$

Using the normal-mode ansatz and parallel flow assumptions the resultant linearized Navier–Stokes equations, known as the Orr–Sommerfeld equations [9], are obtained

$$([D^2 - (\alpha^2 + \beta^2)]^2 - iR(\alpha U + \beta W - \omega)[D^2 - (\alpha^2 + \beta^2)] - (\alpha D^2 U + \beta D^2 W))\hat{v} = 0, \tag{33}$$

$$\hat{v}(0) = 0, \quad D\hat{v}(0) = 0, \quad \hat{v}(y) \to 0, \quad D\hat{v}(y) \to 0, \quad y \to \infty. \tag{34}$$

In particular, the eigenvalue relation for this problem will be of the form

$$F(\alpha, \beta, \omega, R) = 0. \tag{35}$$

The above eigenvalue problem needs to be solved to obtain the growing modes for given frequency. As the eigenvalue α appears nonlinearly, a companion-matrix method has been used to solve the nonlinear eigenvalue problem as follows.

Equation (33), in which the parameter α appears to the fourth power, is cast as the eigenvalue problem $M_4\hat{v} = 0$, with M_4 a matrix containing polynomials of α of highest order 4. This can be expressed as a scalar polynomial with matrix coefficients C_0, C_1, C_2, C_3 as

$$M_4 = C_0\alpha^4 + C_1\alpha^3 + C_2\alpha^2 + C_3\alpha + C_4. \tag{36}$$

A companion matrix is formed for (36) (as shown in [2]), where

$$\left[\begin{pmatrix} -C_3 & -C_2 & -C_1 & C_0 \\ I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \end{pmatrix} - \alpha \begin{pmatrix} -C_4 & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{pmatrix}\right] \begin{pmatrix} \alpha^3\hat{v} \\ \alpha^2\hat{v} \\ \alpha\hat{v} \\ \hat{v} \end{pmatrix} = 0. \tag{37}$$

Here $\hat{v}$ is the eigenvector of $M_4(\alpha)$. The eigenvalues of (37) can be obtained using a QR algorithm. The main disadvantage of this method is large memory requirements, which is four times the size of the discretized coefficient matrix M_4. To compute the derivatives in the Orr–Sommerfeld equation, fourth-order compact differences have been used.

From the normal-mode ansatz, it can be seen that the growth rate of the disturbances in the x- and z-directions can be obtained from the direct numerical simulation (DNS) solver using

$$(1/A)\, dA/dx = -\alpha_i, \quad (1/A)\, dA/dz = -\beta_i, \tag{38}$$

where A is the magnitude of the flow variables u, v, or w velocity, and the spatial amplification is

$$|\sigma| = \left(\alpha_i^2 + \beta_i^2\right)^{1/2}. \tag{39}$$

These growth rates and the mode profiles of the disturbances from the DNS solver can be compared with the eigenvalues and eigenfunctions obtained from the Orr–Sommerfeld equations.

6. TIME INTEGRATION SCHEMES

Equations (4) and (10) can be represented as

$$\frac{dU}{dt} = N + LU, \tag{40}$$

where U represents ω_2 or ϕ, N represents the nonlinear terms H_ω or H_v, and L denotes the diffusion term Δ or Δ^2.

Semi-implicit time integration has been used to integrate the discretized equations in time. For the nonlinear terms an explicit low-storage fourth-order Runge–Kutta scheme has been used, and for the linear viscous terms, an implicit Crank–Nicholson scheme has been used. The general idea of low-storage RK schemes [6] is to leave the useful information in the storage register at each successive stage. The RK4 scheme can be cast as a $2N$-storage scheme [29] using cumulative storage, where N is the dimension of the system of ODEs. Implementation of this scheme is very simple, and there is a significant reduction in the memory requirements for the storage of the variables compared to classical RK schemes as only two variables need to be stored. To avoid stringent time step restrictions the linear viscous terms have been integrated using an implicit Crank–Nicholson scheme, which does not have time step restrictions due to the CFL criterion. Appendix A. 1 discusses the stability criterion for the present time integration scheme. The solution at the end of each time step is the sum of the solutions of the explicit and implicit parts. This yields

$$U^{n+1} = U^n + A_n dt N^n + B_n dt N^{n-1} + (A_n + B_n)\, dt \frac{LU^{n+1} + LU^n}{2}, \tag{41}$$

where A_n and B_n are the RK coefficients and n represents the Runge–Kutta substep.

7. INFLUENCE MATRIX METHOD

In the present case, when the fourth-order vertical velocity equation is split up into two second-order equations, one obtains an evolution equation for the Laplacian of vertical velocity ϕ (10) and a Poisson equation (11) for vertical velocity. For solving the ϕ equation, boundary conditions for ϕ are required, which are obtained using an influence matrix method [7, 8, 26]. The underlying idea is the principle of superposition of solutions to linear problems. The time-discretized problem that needs to be solved is described by (10) and (11) subject to the set of boundary conditions (15), (16), which gives a complete set for v and ϕ. The value of ϕ at the boundary is not known and needs to be determined such that it is consistent with the velocity boundary conditions.

This original problem can be split up into a pair of linear problems. Let $(\tilde{\phi}, \tilde{v})$ satisfy the set of equations for the *A Problem*

$$\left(I - \frac{1}{\mathrm{Re}} \frac{(A_n + B_n)\, dt}{2} \Delta\right) \tilde{\phi}^{n+1} = \left(I + \frac{1}{\mathrm{Re}} \frac{(A_n + B_n)\, dt}{2} \Delta\right) \phi^n + \left(A_n H_v^n + B_n H_v^{n-1}\right) dt, \tag{42}$$

$$\Delta v^{n+1} = \tilde{\phi}^{n+1}, \tag{43}$$

$$\phi|_{y=0,N} = \tilde{\phi}_{guess}, \tag{44}$$

along with one of the two pairs of boundary conditions (15), (16). Then let $(\hat{\phi}, \hat{v})$ be the difference between the true solutions ϕ, v and the intermediate solutions $\tilde{\phi}, \tilde{v}$ of the A problem, $(\tilde{\phi} - \phi, \tilde{v} - v)$. Due to the linearity of the problem this difference is a solution of the homogenous system, which we refer to as our *B Problem*, described by the homogeneous equations

$$\left(I - \frac{1}{\mathrm{Re}} \frac{(A_n + B_n)\, dt}{2} \Delta\right) \hat{\phi}^{n+1} = 0, \tag{45}$$

$$\Delta \hat{v}^{n+1} = \hat{\phi}^{n+1}. \tag{46}$$

For the v boundary condition one pair of (15), (16) is used, the boundary conditions for ϕ are

$$\hat{\phi}_k[\gamma(j)] = \delta_{kj}, \quad \forall \gamma(j) \in \Gamma, \tag{47}$$

where Γ is the lower and upper boundary, and $\gamma(j)$ corresponds to the grid points on the boundary where ϕ needs to be determined. Suppose the velocity and the ϕ fields are expressed as a sum of the particular and homogeneous solutions. Then by virtue of superposition the solutions (ϕ, v) can be expressed as

$$v = \tilde{v} + \sum_{k=1}^{N_T} \lambda_k \hat{v}_k, \tag{48}$$

$$\phi = \tilde{\phi} + \sum_{k=1}^{N_T} \lambda_k \hat{\phi}_k. \tag{49}$$

Here the summation $k = 1$ to N_T is over all the boundary points.

All the coefficients λ_k are obtained by requiring (16) on the lower and the upper boundary. Using (48) and (49) gives

$$(\tilde{\zeta} - \tilde{\phi})(M_i) + \sum_{k=1, N_T} \lambda_k (\hat{\zeta}_k - \hat{\phi}_k)(M_i) = 0, \tag{50}$$

where we assign $\tilde{\zeta} = \Delta \tilde{v}$ for discretization points $M_i = 1 \ldots N_T$. So $(\hat{\zeta}_k - \hat{\phi}_k)(M_i) \equiv A_{ik}$ are the elements of the matrix A. The unknown coefficients λ_k are determined using

$$\lambda_k = -\left(A_{ki}^{-1}\right) \sum_{i=1}^{N_T} ((\tilde{\zeta} - \tilde{\phi})(M_i), \tag{51}$$

where the inverse of A is referred to as the influence matrix. The influence matrix needs to be computed only once in a preprocessing step, and it need not be computed for each time step.

8. STEADY-STATE NAVIER–STOKES SOLUTION

To start the simulations, we require an initial steady-state solution for the Navier–Stokes equations. For this purpose, a steady Navier–Stokes solver has been developed which uses

for its boundary conditions

- at the inflow: Given Blasius profile,
- at the outflow: First-order convective boundary conditions of the form

$$\frac{\partial v}{\partial t} + U_\infty \frac{\partial v}{\partial x} = 0, \tag{52}$$

- at the wall: No-slip and no through-flow conditions,
- at the free stream:

$$\frac{\partial v}{\partial y} = 0, \tag{53}$$

$$\frac{\partial^2 v}{\partial^2 y} = 0. \tag{54}$$

The above free-stream boundary conditions work well for the steady Navier–Stokes solver but for the case where perturbations are added, exponentially decaying velocity at the boundary needs to be used. Figures 2 and 3 show the steady Navier–Stokes solution. It takes roughly 20 TS time periods to reach a steady solution starting from the Blasius profile. It should be noted that the difference between the steady Navier–Stokes solution and Blasius solution is only of order 10^{-4}.

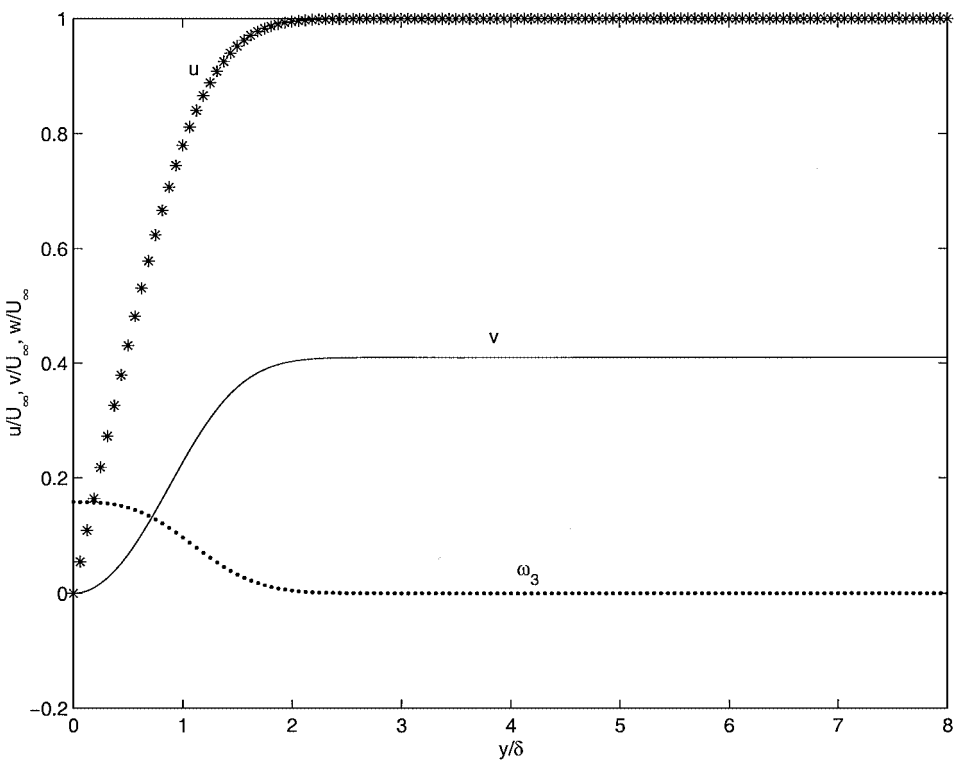

FIG. 2. Steady Navier–Stokes solution obtained as inflow condition for transition simulations.

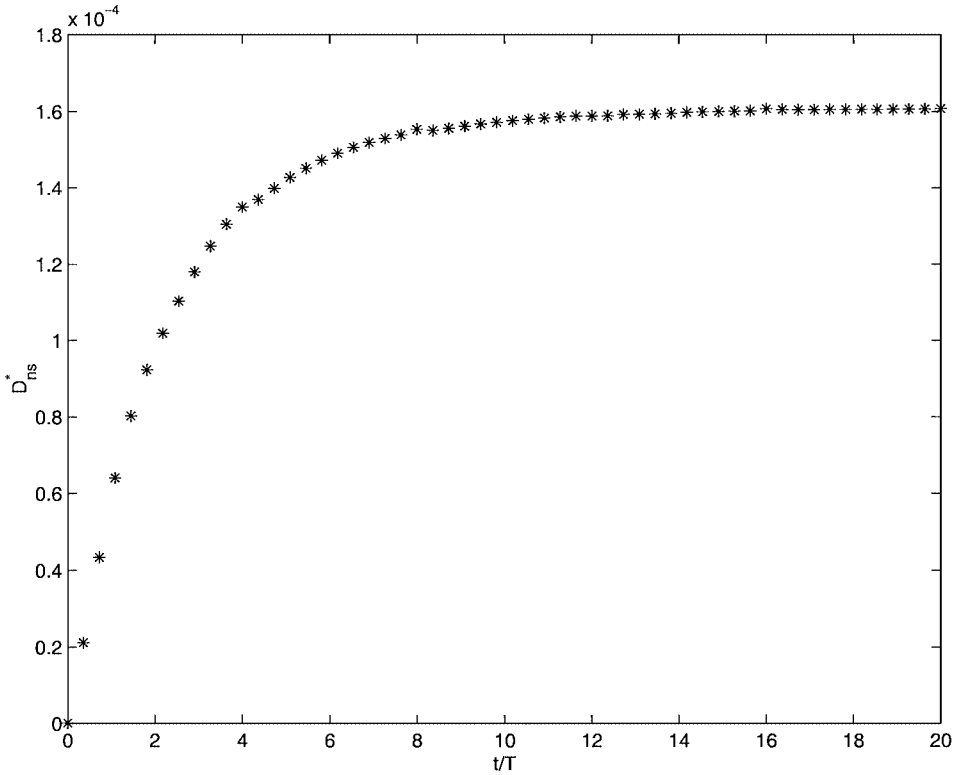

FIG. 3. Deviation of NS solution from Blasius solution, as a function of time during the solution of the steady problem.

9. OUTFLOW BOUNDARY CONDITIONS

Selection of good outflow boundary conditions is very important for a well-behaved numerical scheme. The general types of outflow that can be used for boundary-layer type flows are

- Convective boundary conditions: At the outflow, the basic assumption is that all the disturbances are convected out of the domain. The vertical velocity outflow condition is

$$\frac{dv}{dt} + U_\infty \frac{dv}{dx} = 0. \tag{55}$$

A first-order difference scheme for the first derivative is good enough. This method works well for problems which have parabolic behavior at the outflow. In the present study these boundary conditions gave good results for obtaining the steady Navier–Stokes solution.

- Parabolizing the Navier–Stokes equations at the outflow: When the equations for the perturbations were solved, convective boundary conditions were not yielding good results (numerical instability developed). For this reason, the present boundary method has been developed. The streamwise derivatives can be neglected as these are small compared to the normal derivative, which allows the outflow equations to be solved for the given boundary conditions. The essential idea is that in our case the Navier–Stokes equations represent an IVP that is approximately parabolic in nature. At the outflow, the physics of the problem

can be used to neglect the streamwise derivatives, as these are small compared to the normal derivatives, and the outflow equations can be solved for the given boundary conditions. The parabolized equations at the outflow are

$$\left.\frac{\partial \phi}{\partial t}\right|_{\text{outflow}} = \left.\left(H_v + \frac{1}{\text{Re}}\frac{\partial^2 v}{\partial^2 y}\right)\right|_{\text{outflow}}, \tag{56}$$

$$\phi|_{\text{outflow}} = \left.\left(\frac{\partial^2 v}{\partial y^2}\right)\right|_{\text{outflow}}. \tag{57}$$

• Buffer domain technique: This outflow method is based on the idea that, in a region close to the outflow, the disturbances are damped to zero [26]. This particular region does not give physically valid results, but the method prevents the reflection of the physical instabilities at the outflow boundary. Wave reflections tend to have disastrous results: When they reflect from the outflow and reach the inflow the waves interact and superpose and eventually the solution blows up. Buffer domain methods work well for elliptical problems. A damping function ζ is selected to satisfy the following boundary conditions.

1. Disturbances enter with the original level of amplitude; $\zeta(0) = 1$,
2. no discontinuity at the beginning of the domain; $\frac{\partial \zeta(0)}{\partial x_b} = 0$,
3. the damping function decreases monotonically; $\frac{\partial \zeta(0)}{\partial x_b} \leq 0$,
4. zero disturbance is guaranteed at the outflow of the domain; $\zeta(1) = 0$.

Figure 4 shows the damping function that has been used. For the range of values of

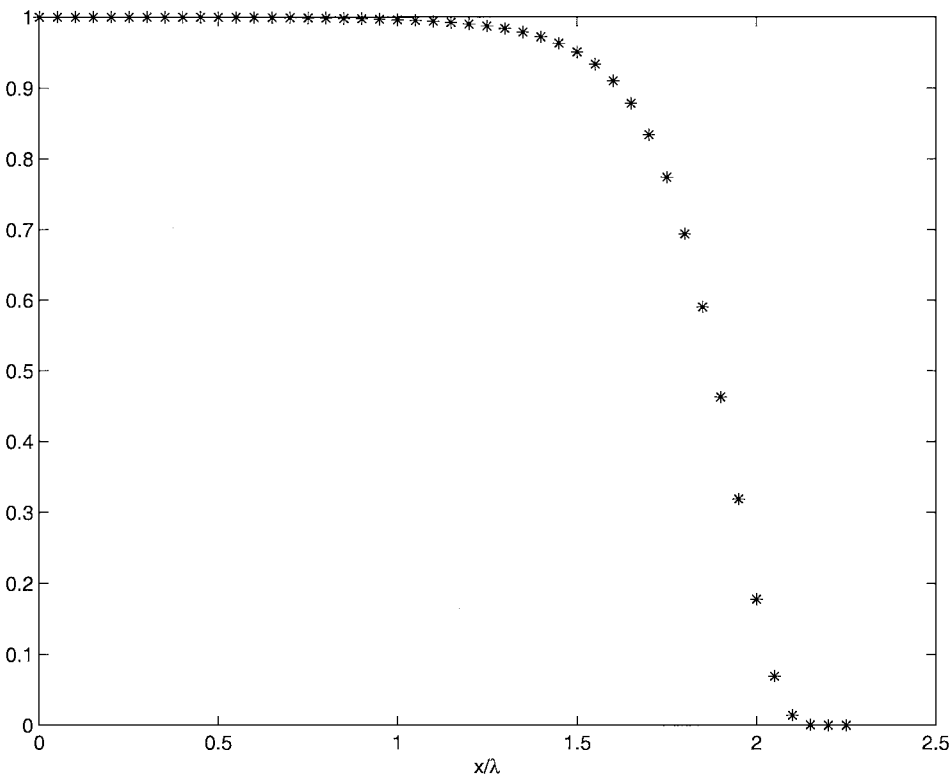

FIG. 4. Ramp function in the buffer domain.

$x_s \le \zeta \le x_e$, let

$$x_b = \frac{x - x_s}{x_e - x_s}, \tag{58}$$

where x_s is the start and x_e is the end of the buffer domain. We then use

$$\zeta = \left(1 - C_1 x_b^2\right)\left(1 - \frac{f_1}{f_2}\right), \tag{59}$$

$$C_1 = 0.1, \quad C_2 = 10, \quad C_3 = 330, \tag{60}$$

$$f_1 = 1 - \exp\left(C_2 x_b^2\right), \tag{61}$$

$$f_2 = 1 - \exp(C_2). \tag{62}$$

To make the damping function independent of the number of time steps/disturbances, a damping function $\tilde{\zeta}$ is used with tuning parameter C_3 [21].

$$\tilde{\zeta} = \zeta^{C_3 \omega \Delta t}, \tag{63}$$

where ω is the frequency of TS waves.

10. RESULTS

The Helmholtz and Poisson solvers have been tested using analytical functions. Results are shown for the following two cases:

Case 1. Equation (14) is solved with the right-hand side ϕ as

$$\phi = [(-16\epsilon - 1)\pi^2] \sin(4\pi x) \cos(\pi y). \tag{64}$$

Case 2. Equation (13) is solved with the right-hand side ψ as

$$\psi = [1 - (\epsilon + 1)]\pi^2 \sin(\pi x) \sin(\pi y). \tag{65}$$

We have tested for ϵ ranging from 10^{-2} to 10^{-5}. Figures 5 and 6 show solution and residue for Case 1; Figs. 7 and 8 show solution and residue for Case 2. Dirichlet boundary conditions have been prescribed for both cases. For these two cases, the computations have been performed on a mesh of size 513×513. Two V-cycles, three presmoothing steps, and one postsmoothing step have been used. Case 1 has been used to test the Poisson solver and Case 2 to test the Helmholtz solver. In both cases, the smoother used is the y-line Gauss–Seidel smoother. The residues are of order $O(10^{-9})$ and $O(10^{-11})$ for the Poisson and Helmholtz solvers respectively and the error in the solution was of order $O(10^{-7})$.

A reliable test for a transition code is to compare the early stages of the transition regime, where the disturbances are still small, with the Orr–Sommerfeld theoretical results, which are based on the parallel-flow assumption of the Navier–Stokes equations. The parameters that have been used are

$$L = 0.05 \text{ m}, \tag{66}$$

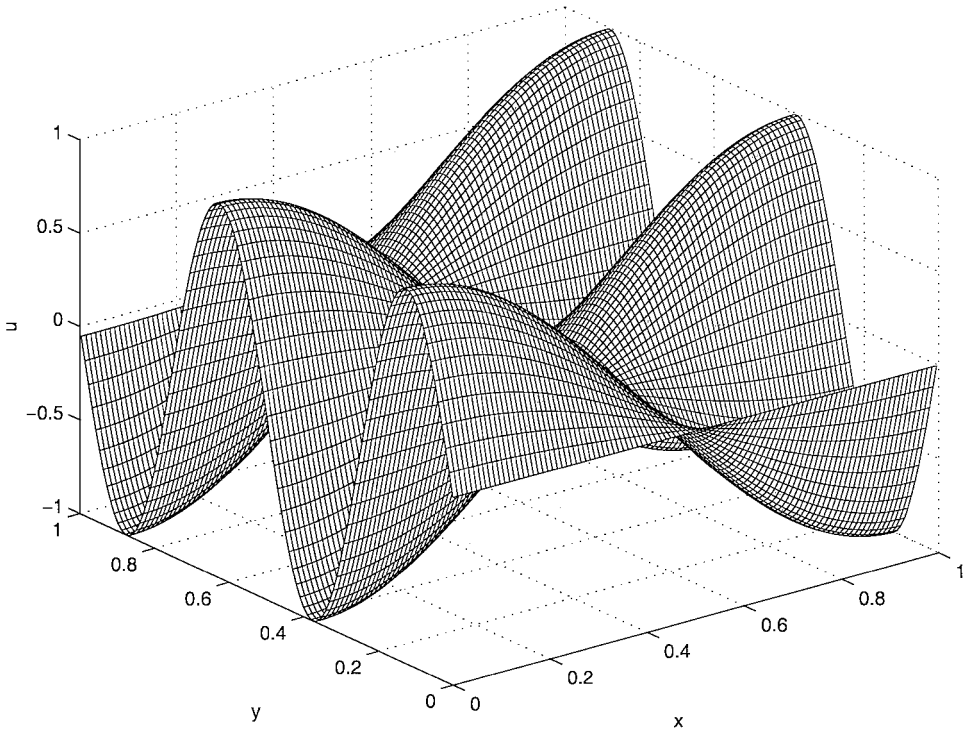

FIG. 5. Solution of Poisson solver for Case 1 obtained after two V-cycles, three presmoothers, and one postsmoother. The error in the solution is of $O(10^{-8})$.

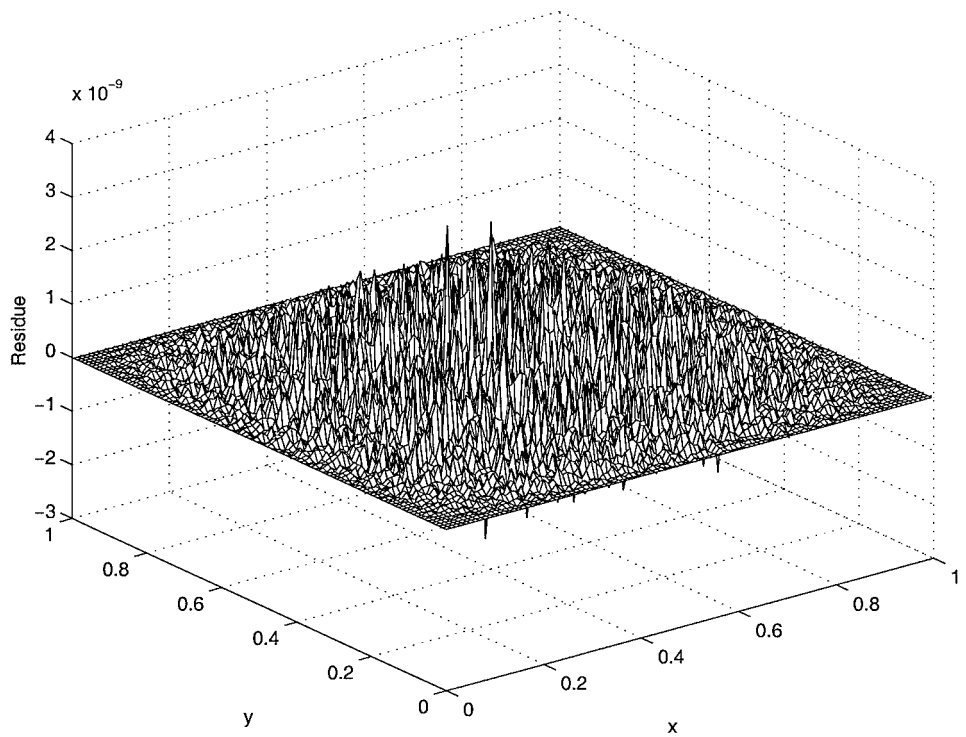

FIG. 6. Residue of Poisson solver for Case 1.

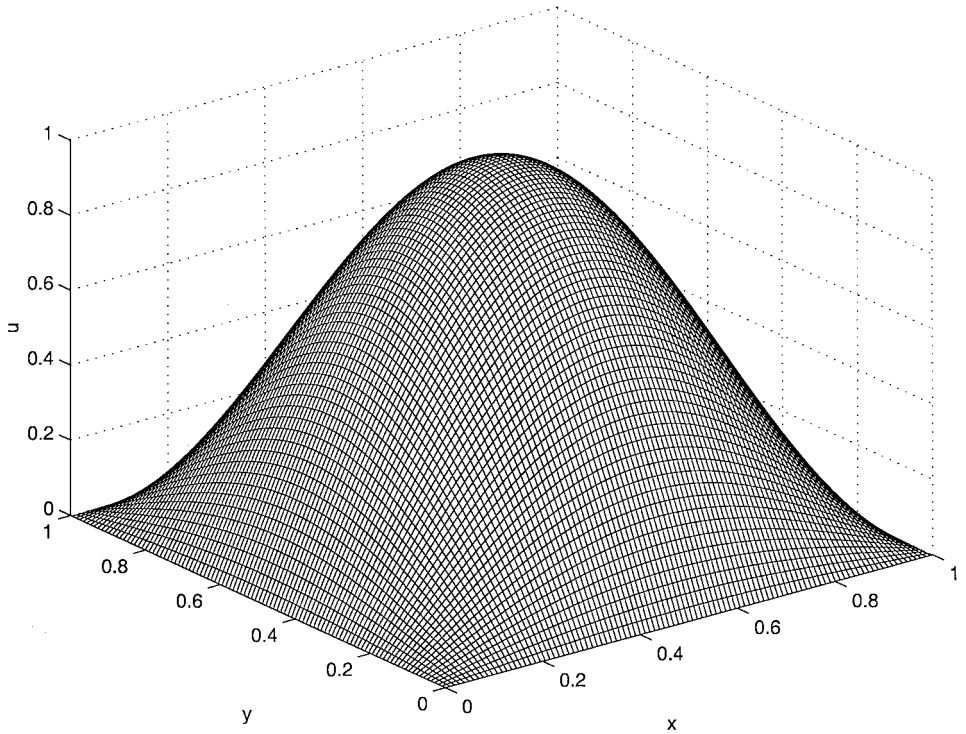

FIG. 7. Solution of Helmholtz solver for Case 2 obtained after two V-cycles, three presmoothers, and one postsmoother. The error in the solution is of $O(10^{-11})$.

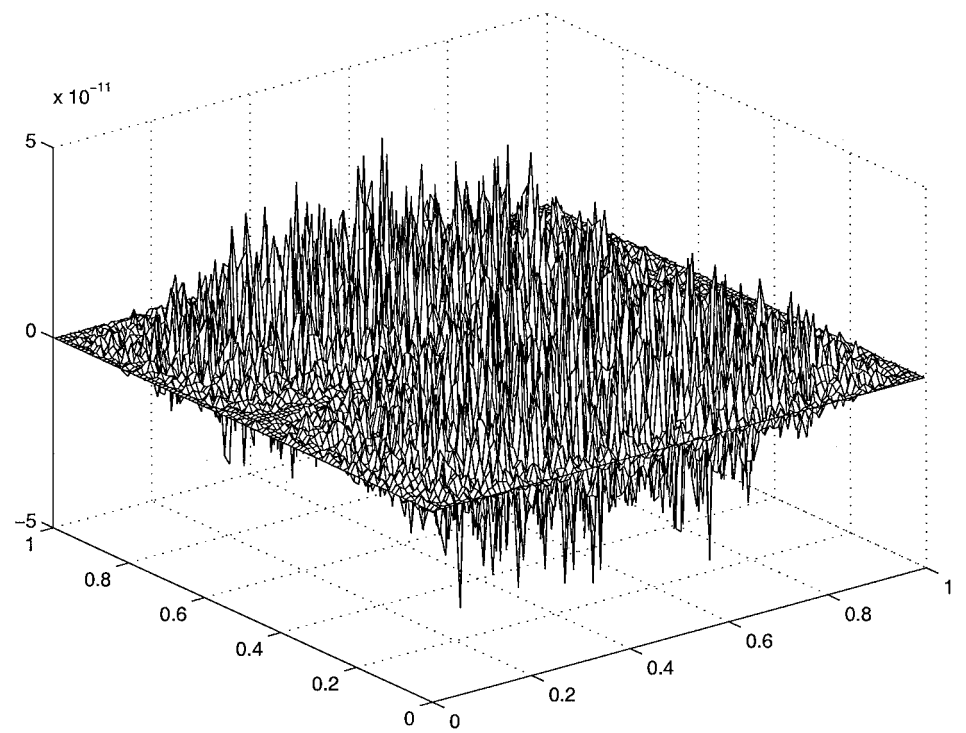

FIG. 8. Residue of Helmholtz solver for Case 2.

$$U_\infty = 30 \text{ m/s}^2, \tag{67}$$

$$x_0 = 70 \text{ mm}, \tag{68}$$

$$y_N/\delta^* = 10, \tag{69}$$

where L is a length scale, U_∞ is the free-stream velocity, x_0 is the beginning of the integration domain in the streamwise direction, measured from the leading edge of the half-infinite plate, and y_N is the height of the integration domain in the wall–normal direction.

Initially, tests have been performed for the case where the base flow is perturbed with a 2D disturbance of amplitude 0.01% of the free-stream velocity and $A_{3D} = 0$(A.5). The mode profiles and growth rates are compared with linear stability theory at different x- and y-locations; then 3D disturbances (A.5) have been introduced. Figure 9 shows the comparison with linear theory in the small-amplitude regime of the transition process at an x-location of 195 mm.

For the nonlinear computations the beginning of the integration domain x_0 has been moved to 190 mm. The initial disturbances are in the form of blowing and suction, which are similar to the vibrating ribbon experiments, where the ribbon induces a predominantly two-dimensional response, in which the least-stable linear mode rapidly dominates the flow downstream of the ribbon; this has been found to give pure Tollmien–Schlichting waves.

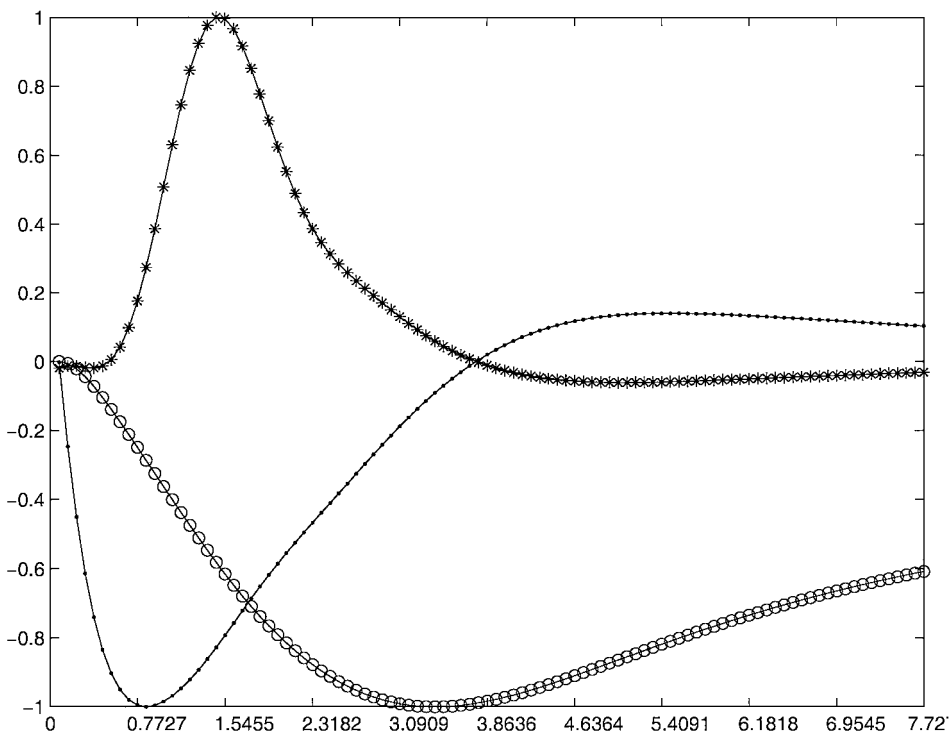

FIG. 9. Comparison of the mode profiles of scaled streamwise, normal, and spanwise velocity with linear theory for $\text{Re}_L = 10^5$, $\omega = 10$, $x = 195$ mm. The symbols represent the DNS solution and the lines are the Orr–Sommerfeld solutions.

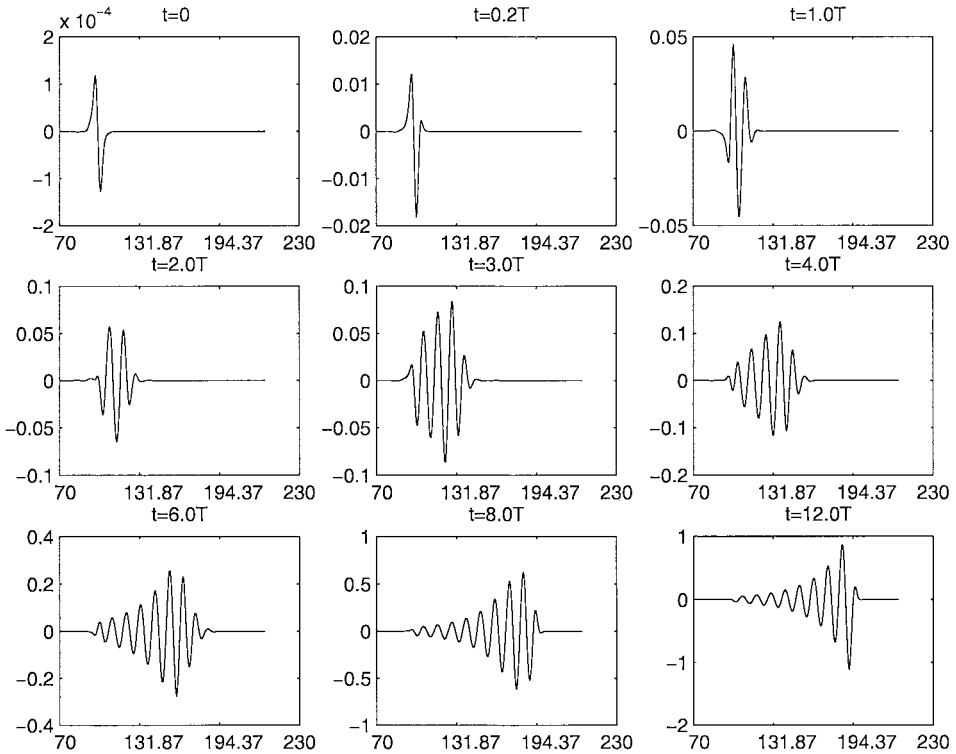

FIG. 10. The growth of the normal component of a TS wave in the streamwise direction from the initial conditions to $t = 12$ TS time periods, at $y/\delta = 0.5$.

The resulting TS waves evolve linearly for several wavelengths downstream; eventually three-dimensionality creeps in, leading to laminar breakdown followed by turbulence. Figure 10 shows the evolution of the normal component of the TS modes in the x-direction from time $t = 0$ to $t = 10$ TS time periods. The initial disturbance ($t = 0$) is given in an otherwise undisturbed flow, and after 10 time periods TS waves develop and amplify in the spatial direction. Figures 11 and 12 show TS waves in the xy-plane at $z = 1$ and $z = 0$, respectively. Figure 13 shows the spanwise velocity component of the TS wave, when the TS wave becomes three-dimensional. The buffer domain starts around $x = 185$ mm; disturbances grow up to this location and then get damped. A buffer domain of approximately two TS wavelengths has been chosen for this particular case. The length of the buffer domain that was chosen represents a minimal value obtained from numerical experiments. After an initial transient period the TS wave grows in accordance with linear theory in the linear regime. All the results given here are after 10 TS time periods. It takes some time for the disturbances to reach a periodic state and the time history of the TS wave is shown in Fig. 14.

After thorough comparisons have been made in the linear regime for both 2D and 3D disturbances at different x- and y-locations, the size of the integration domain has been increased and the amplitude of the disturbances has been increased to 1% of the free-stream velocity to induce nonlinear disturbance development in the flow. As the large-amplitude disturbances travel downstream, they assume an increasingly nonlinear character. Finally around $x = 410$ mm as seen in Figs. 15 and 16, the first signs of randomness appear,

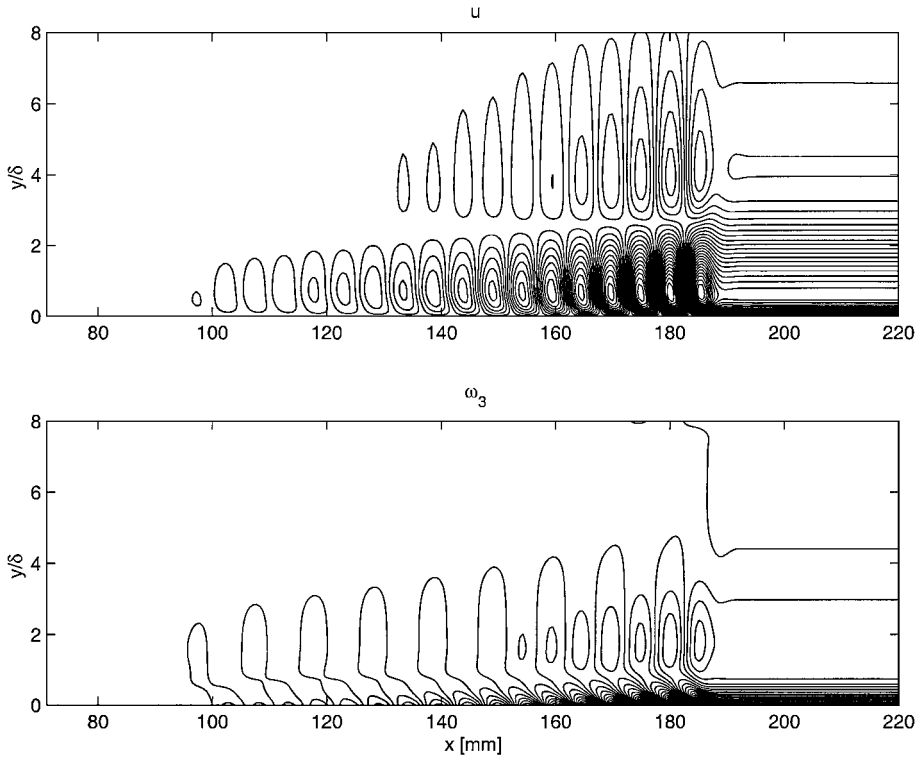

FIG. 11. Contour plot of streamwise velocity and spanwise vorticity in the xy-plane at $z = 1$.

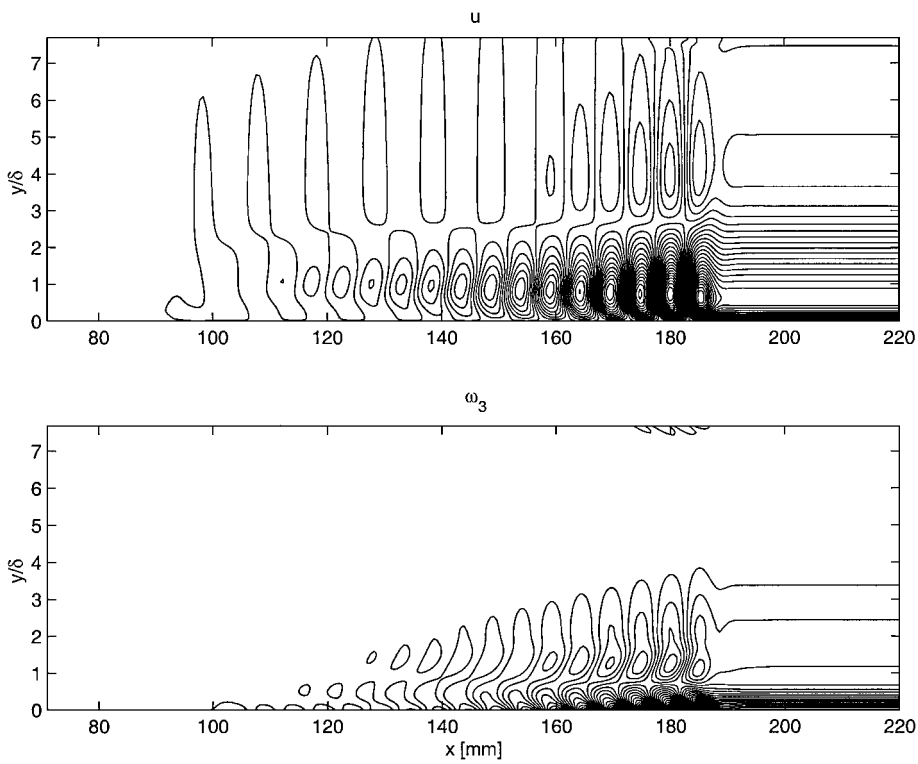

FIG. 12. Contour plot of streamwise velocity and spanwise vorticity in the xy-plane at $z = 0$.

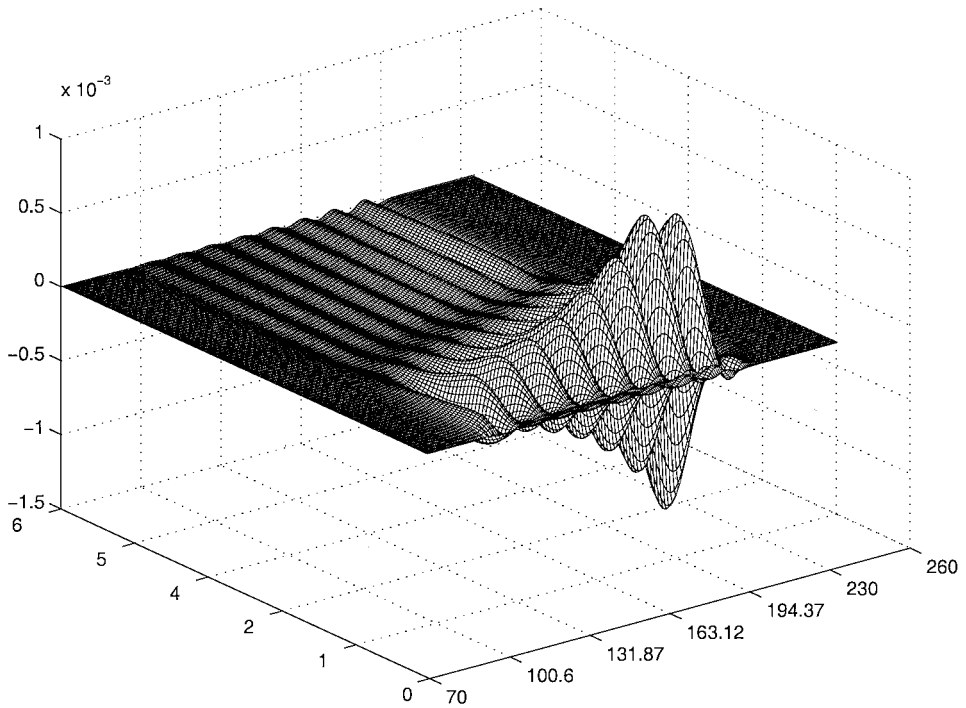

FIG. 13. Spanwise velocity in xy-plane at $z = 0$ with initial disturbances of amplitude 0.01% of free-stream velocity.

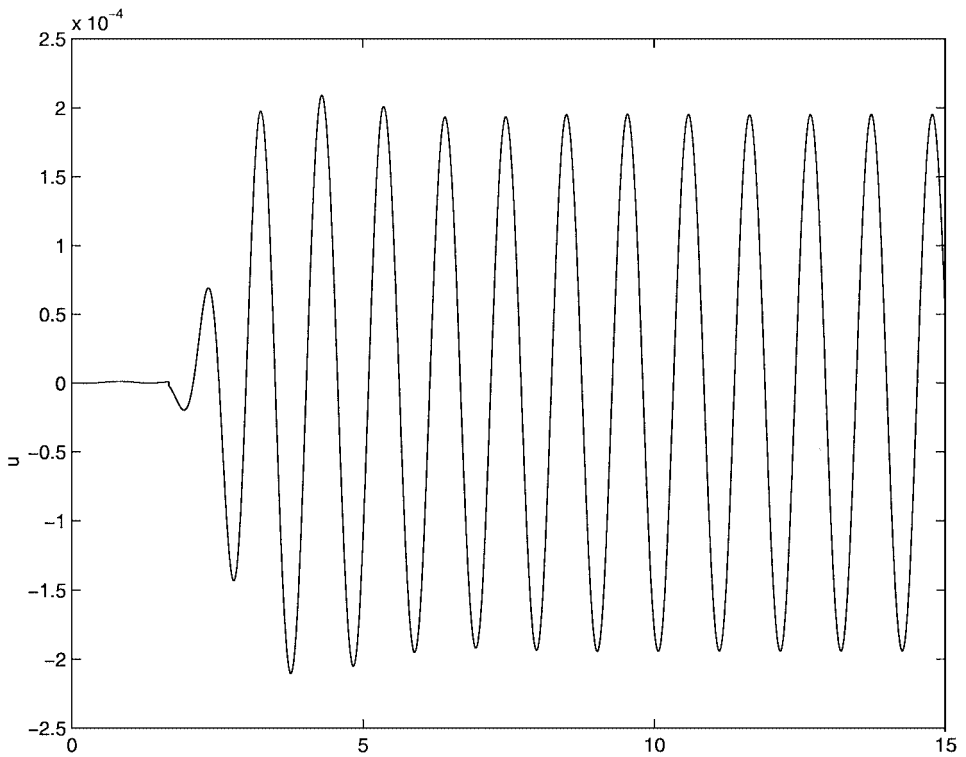

FIG. 14. Time periodic behavior of a TS wave starting from the initial time of $t = 0$ to $t = 15$ TS time periods at x-location of 195 mm, averaged in y-direction.

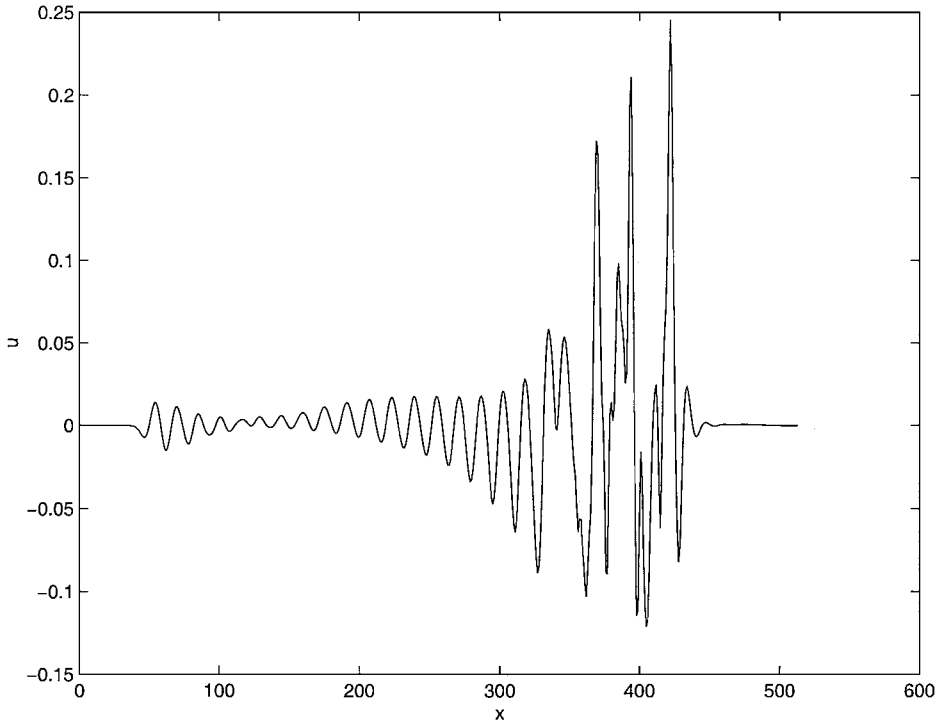

FIG. 15. Streamwise velocity at $y = 0.4$ mm from wall, $z = 0$; nonlinear effects can be seen beyond $x = 400$ mm.

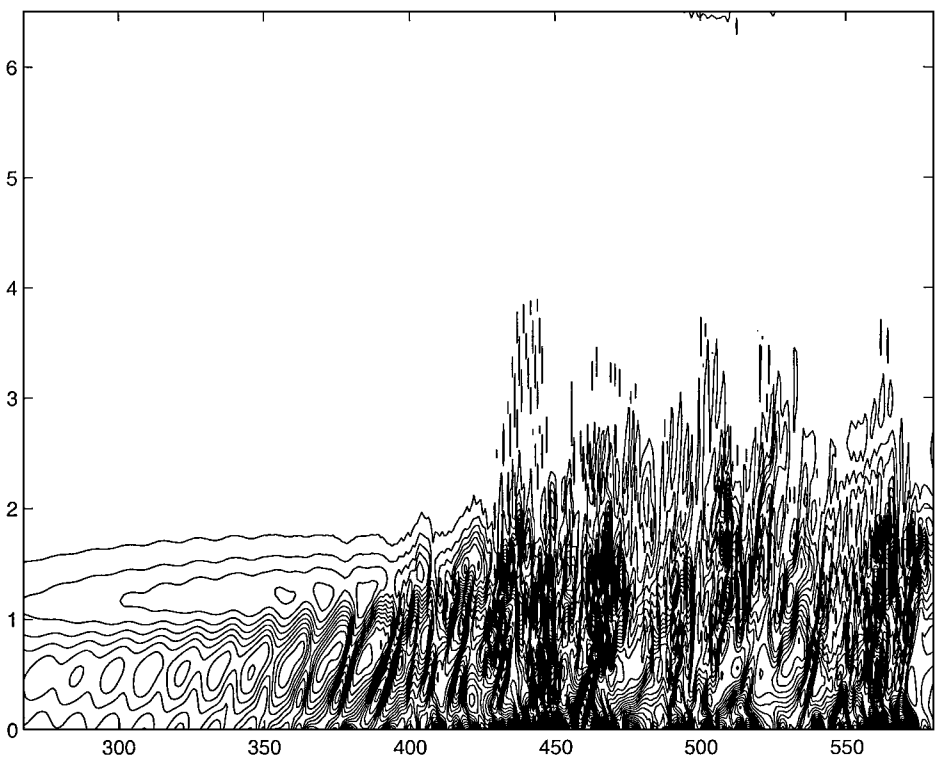

FIG. 16. Spanwise vorticity contours at $z = 0$, after 12 TS time periods in xy-plane.

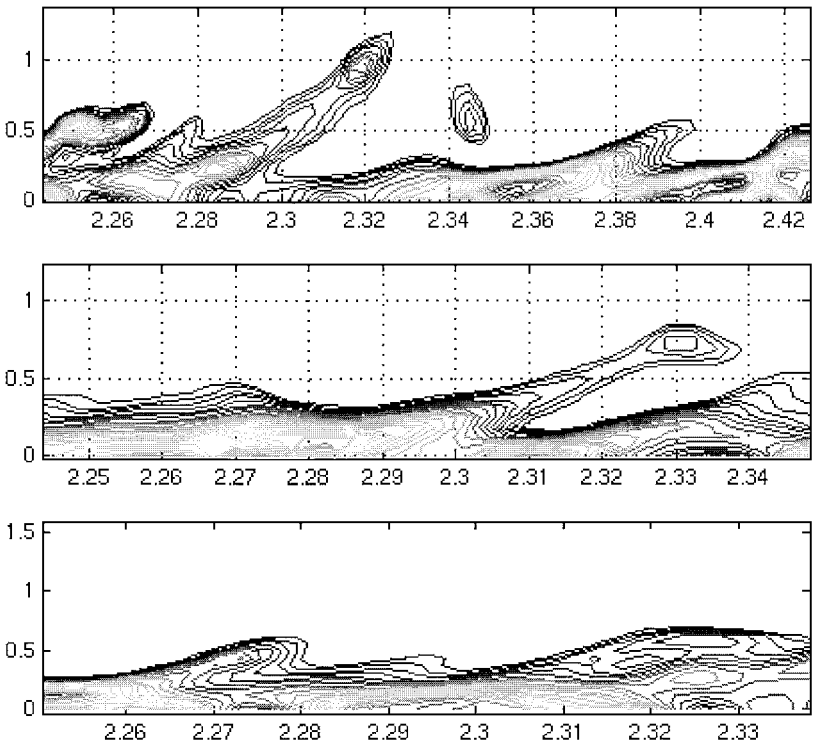

FIG. 17. Spanwise vorticity in the xy-plane at the $z = 0$ plane at time $t = 12.0T$, $12.4T$, and $12.6T$. The x-domain extends from nondimensional value from 2.3 to 3.3, which corresponds to $x = 350$ mm to $x = 410$ mm; y extends to a height of four displacement thicknesses (simulation performed with 32 Fourier modes to capture the shear layer).

which is at about the same location where spikes have been observed in experiments of Kachanov [15, 16] (the same domain parameters have been selected so that comparisons can be made). The domain in the streamwise direction extends from 190 to 500 mm; the 3D simulations have been computed with only five spanwise Fourier modes, but even with limited spatial resolution in the z-direction, good agreement has been obtained. The same simulations have been performed with increased resolution in the spanwise z-direction, using 32 Fourier modes. Figure 17 shows the spanwise vorticity in the xy-plane at $z = 0$ plane for the three different times of $t = 12.0T$, $12.4T$, and $12.6T$. At $t = 12.4T$ the lift-up of the shear layer can be clearly seen. This is a typical event of the transitional boundary layer. The x-domain extends from a nondimensional value of 2.3 to 3.3, which corresponds to $x = 350$ mm to $x = 410$ mm; y extends to a height of four displacement thicknesses.

11. CONCLUSIONS

As a step toward the goal of better understanding the mechanisms and classical structures of the laminar breakdown process, data of an evolving flat-plate boundary layer have been generated by direct numerical simulation of the Navier–Stokes equations without any

approximations. As it is crucial to avoid the contamination of the physical disturbances with numerical errors, a robust and an accurate transition code that can accurately describe the flow dynamics from the early stages of transition to a fully developed turbulent state has been developed. One of the aims of the present paper is to show the feasibility of developing a reliable code without many approximations for studying the physical problem of transition in open shear flows; we have discussed a number of numerical issues involved in the development of a code to study the spatial transition to turbulence and addressed the issues of concern (Sections 2–9). An important feature of the present algorithm is the improved resolution characteristics of the spatial derivatives compared to traditional finite-difference approximations, and thus it provides a better representation of the shorter length scales. Hence, it is highly suitable for the study of the laminar breakdown process, as all the harmonics of the fundamental modes generated need to be well resolved.

Further, one of the difficulties in solving the equations in the primitive variable formulation is obtaining a consistent boundary condition for the pressure, and an another fundamental difficulty arises from the saddle point nature of the discrete Stokes problem [3], unless special precautions are taken. Hence in the early 1990s the velocity–vorticity approach became popular (the terms are obtained by taking the curl of the NS equations in the primitive variable form) as the pressure term no longer appears in Eq. [11]. But this involves solving three vorticity transport equations for the three vorticity components and three Poisson-type equations for the velocity components—thus increasing the number of equations to be solved, as well as the number of variables to be stored in memory. We have used a formulation in terms of vertical velocity and vertical vorticity because this has the advantage of eliminating the pressure term from the Navier–Stokes equations, while requiring the solution of only two governing equations, namely an evolution equation for the Laplacian of the vertical velocity and an evolution equation for the normal vorticity. Thus, this formulation leads to reduced storage requirements compared to solutions using the Navier–Stokes equations in primitive variables or in a vorticity-transport formulation. Simulation of spatial transition is a computationally intensive process, minimizing the number of equations to be solved, and the number of variables to be stored is crucial. The boundary conditions for both v_2 and ω_2 are well defined at the inflow, wall, and free stream. The nonreflecting boundary condition at the outflow prevents nonphysical wave reflections at the outflow and a possible contamination of these reflected waves with the growing physical disturbances causing the numerical instability to trigger a nonphysical laminar breakdown.

The structure of the present algorithm is suitable for a straightforward and simple parallelization; solving the elliptical equations using the multigrid solver takes around 60% of the total computational time per time step. The multigrid algorithm has excellent algorithmic scalability features as well as implementation scalability features, thus making the multigrid routine the most suitable candidate for parallelization and resulting in a significant speedup.

For this simulation, we have assumed the flow to be periodic in time with transition occurring spatially, in the downstream direction. To simulate spatial transition to turbulence, we have developed a code solving the full three-dimensional nonlinear Navier–Stokes equations. The spatial domain extends 20 Tollmien–Schlichting wavelengths in the streamwise direction and 10 displacement thicknesses in the wall–normal direction, and Fourier

modes have been used in the spanwise z-coordinate, which is assumed periodic. For the results shown in the present paper, which mainly focuses on validation issues, we have used only four spanwise Fourier modes. Simulations with higher spanwise resolution are currently being performed.

A semi-implicit time integration scheme alleviates the problem of time step restriction. Analytical functions have been used to test the robustness and accuracy of the solvers. Forced transition is being simulated by adding disturbances in the domain, in the form of blowing and suction, which is similar to vibrating ribbon devices used in experimental transition studies. For the validation of the code for small disturbances, the DNS solutions were compared with Orr–Sommerfeld solutions and the solutions were found to match very well. The Tollmien–Schlichting waves developed in the linear regime are periodic in time, the flow is laminar and periodic at this stage, and the growth in the streamwise direction exhibits a linear behavior. The amplitude of the disturbances was increased to 1% so that nonlinear effects start developing. In that case, the simple periodicity observed in the linear regime is no longer present. The location where the randomness occurs in our simulation compares well with experiments. This serves as a validation in the weakly nonlinear regime. Our code seems to be well suited to perform simulations of the complete transition process up to the turbulent regime.

APPENDIX

A.1. CFL Criterion

When explicit schemes are used to integrate the equations, a stability criterion has to be met. The numerical stability criterion is usually defined in the form

$$\Delta t \leq \frac{1}{\tilde{\mathrm{CFL}}}, \tag{A.1}$$

where Δt is the time step. In general, the two terms contributing are the viscous and the nonlinear terms, so the resultant criterion is

$$\Delta t \leq \frac{1}{((\tilde{\mathrm{CFL}}_{nl})^2 + (\tilde{\mathrm{CFL}}_{vis})^2)^{1/2}}, \tag{A.2}$$

where CFL_{nl} is the CFL contribution due to the nonlinear terms and CFL_{vis} is the CFL contribution due to the viscous terms

$$\Delta t_{vis} \leq \frac{1}{\frac{\tilde{\mathrm{CFL}}_x}{(\Delta x)^2} + \frac{\tilde{\mathrm{CFL}}_y}{(\Delta y)^2} + \frac{\tilde{\mathrm{CFL}}_z}{(\Delta z)^2}}, \tag{A.3}$$

$$\Delta t_{nl} \leq \frac{1}{\frac{\tilde{\mathrm{CFL}}_x}{(\Delta x)} + \frac{\tilde{\mathrm{CFL}}_y}{(\Delta y)} + \frac{\tilde{\mathrm{CFL}}_z}{(\Delta z)}}, \tag{A.4}$$

where Δx, Δy, Δz represent the spatial discretization in the x-, y-, z-directions. So the resultant Δt is $\min(\Delta t_{nl}, \Delta t_{vis})$. In the present semi-implicit scheme there is no restriction due to the linear terms and the restriction due to the x-, y-, and z-terms is solely the nonlinear restriction.

A.2. Generation of Tollmien–Schlichting Waves

The blowing–suction velocity is prescribed at the wall

$$v(x, 0, z, t) = (A_{2d} f(x) + A_{3d} \cos(\gamma z)) \sin(\omega t), \tag{A.5}$$

where A_{2d}, A_{3d} are the two and three-dimensional disturbance amplitudes, $f(x)$ is the distribution function, ω is the frequency obtained from linear stability theory, and α, γ are the corresponding streamwise and spanwise wavenumbers. The distribution function is of the form

$$f(x_b) = \frac{4 \sin x_b (1 - \cos x_b)}{\sqrt{27}}, \tag{A.6}$$

where the nondimensionalized streamwise coordinate is given by

$$x_b = \frac{2\pi(x - x_s)}{(x_e - x_s)}. \tag{A.7}$$

Here x_e is the starting streamwise location of periodic function and x_s is the ending location of the periodic function.

A.3. Filter of the Equations

It is necessary to eliminate undesirable high-frequencey components of the numerical solution without affecting the low-frequency components. Compact finite-difference schemes have been constructed for the filtering operation. A symmetric filter has been selected as it satisfies the criterion of leaving the phase unchanged, affecting only the amplitude [27]. A three-parameter filter of fourth-order accuracy has been used. The filter F that has been used is such that the filtered velocity u^F is given as

$$\sum_{di=-1,1} \alpha_{di} u^F_{i+di,j} = \sum_{di=-3,3} a_{di} u_{i+di,j}. \tag{A.8}$$

The coefficients used for the interior points are

$$\alpha_0 = 0.475, \quad \alpha_1 = 1.0, \quad \alpha_{-1} = 1.0, \quad a_0 = 0.85625, \tag{A.9}$$

$$a_1 = \frac{1.0375}{2.0}, \quad a_{-1} = \frac{1.0375}{2.0}, \tag{A.10}$$

$$a_2 = \frac{0.11875}{2.0}, \quad a_{-2} = \frac{0.11875}{2.0}, \tag{A.11}$$

$$a_3 = \frac{-1.0}{32.0}, \quad a_{-3} = \frac{-1.0}{32.0}. \tag{A.12}$$

For the numerical boundary points, the filter coefficients are

$$u^F = \frac{15}{16} u_1 + \frac{1}{16}(4u_2 - 6u_3 + 4f_4 - u_5). \tag{A.13}$$

The filter has been applied at every fifth time step.

A.4. Poisson and Helmholtz Solver Coefficients

The matrix elements $P1_{block}$, $P2_{block}$, $Q1_{block}$, $Q2_{block}$ for block-banded-tridiagonal matrices P and Q for the Poisson and Helmholtz solvers are as follows: the elements of Fig. 1 given by

$$P = \begin{bmatrix} c & a & b \\ l & dg & u \\ f & d & e \end{bmatrix}. \tag{A.14}$$

The corresponding values for elements for Poisson and Helmholtz solvers are given by $Q1_{rhs}$, $P1_{lhs}$, $Q2_{rhs}$, $P2_{lhs}$, where $Q1_{rhs}$, $P1_{lhs}$ are the corresponding elements in the right-hand-side and left-hand-side matrices of the Poisson solver and $Q2_{rhs}$, $P2_{lhs}$ are the corresponding elements in the right-hand-side and left-hand-side matrices of the Helmholtz solver

$$Q1_{rhs} = \begin{bmatrix} \frac{6}{50}\left(\frac{1}{\mathrm{Re}}\frac{1}{dx^2} + \frac{1}{dy^2}\right) & -2.4\left(\frac{0.1}{\mathrm{Re}}\frac{1}{dx^2} - \frac{0.5}{dy^2}\right) & \frac{6}{50}\left(\frac{1}{\mathrm{Re}}\frac{1}{dx^2} + \frac{1}{dy^2}\right) \\ -2.4\left(\frac{0.1}{\mathrm{Re}}\frac{1}{dx^2} - \frac{0.5}{dy^2}\right) & -2.4\left(\frac{1}{\mathrm{Re}}\frac{1}{dx^2} + \frac{1}{dy^2}\right) & -2.4\left(\frac{0.1}{\mathrm{Re}}\frac{1}{dx^2} - \frac{0.5}{dy^2}\right) \\ \frac{6}{50}\left(\frac{1}{\mathrm{Re}}\frac{1}{dx^2} + \frac{1}{dy^2}\right) & -2.4\left(0.1\frac{1}{\mathrm{Re}}\frac{1}{dx^2} - \frac{0.5}{dy^2}\right) & \frac{6}{50}\left(\frac{1}{\mathrm{Re}}\frac{1}{dx^2} + \frac{1}{dy^2}\right) \end{bmatrix} \tag{A.15}$$

$$P1_{rhs} = \begin{bmatrix} 0.01 & 0.1 & 0.01 \\ 0.1 & 1.0 & 0.1 \\ 0.01 & 0.1 & 0.01 \end{bmatrix} \tag{A.16}$$

$$Q2_{rhs} = \begin{bmatrix} 1 + \frac{dt}{2}\left(-\frac{1}{\mathrm{Re}}\frac{12}{dx^2} - \frac{12}{dy^2}\right) & 10 + \frac{dt}{2}\left(-\frac{1}{\mathrm{Re}}\frac{120}{dx^2} + \frac{24}{dy^2}\right) & 1 + \frac{dt}{2}\left(-\frac{1}{\mathrm{Re}}\frac{12}{dx^2} - \frac{12}{dy^2}\right) \\ 10 + \frac{dt}{2}\left(\frac{1}{\mathrm{Re}}\frac{24}{dx^2} - \frac{120}{dy^2}\right) & 100 + \frac{dt}{2}\left(\frac{1}{\mathrm{Re}}\frac{240}{dx^2} + \frac{240}{dy^2}\right) & 10 + \frac{dt}{2}\left(\frac{1}{\mathrm{Re}}\frac{24}{dx^2} - \frac{120}{dy^2}\right) \\ 1 + \frac{dt}{2}\left(-\frac{1}{\mathrm{Re}}\frac{12}{dx^2} - \frac{12}{dy^2}\right) & 10 + \frac{dt}{2}\left(-\frac{1}{\mathrm{Re}}\frac{120}{dx^2} + \frac{24}{dy^2}\right) & 1 + \frac{dt}{2}\left(-\frac{1}{\mathrm{Re}}\frac{12}{dx^2} - \frac{12}{dy^2}\right) \end{bmatrix} \tag{A.17}$$

$$P2_{rhs} = \begin{bmatrix} 1.0 & 10.0 & 1.0 \\ 10.0 & 100.0 & 10.0 \\ 1.0 & 10.0 & 1.0 \end{bmatrix}, \tag{A.18}$$

and where dx, dy is the mesh size in the streamwise and normal directions, respectively, dt is the Runge–Kutta time step, and Re is the Reynolds number of the flow.

ACKNOWLEDGMENTS

This work was supported in part by Contracts F49620-92-J-0287 and F49620-97-1-0203 jointly funded by the U.S. Air Force Office of Scientific Research (Control and Aerospace Programs) and the U.S. Office of Naval Research, in part by Grants F49620-92-J-0038, F49620-99-1-0012, and F49620-96-1-0329, funded by the U.S. Air Force Office of Scientific Research (Aerospace Program), in part by Grant CTS-9613948 funded by the U.S. National Science Foundation and in part by the Physical Oceanography Programs of the U.S. National Science Foundation (Contract OCE-901 7882) and the U.S. Office of Naval Research (Grant N00014-92-J-1547), and in part by the German DFG, Grant 957/5-1.

REFERENCES

1. A. Brandt, Multi-level adaptive solutions to boundary-value problems, *Math. Comput*. **31**(138), 333 (1977).
2. T. J. Bridges and P. J. Morris, Differential eigenvalue problem in which the parameter appears nonlinearly, *J. Comput. Phys*. **55**, 437 (1984).
3. C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral Methods in Fluid Dynamics*, Springer Series in Computational Physics (Springer-Verlag, Berlin/New York, 1988).
4. M. H. Carpenter, D. Gottlieb, and S. Abarbanel, The stability of numerical boundary treatments for compact high-order finite-difference schemes, *J. Comput. Phys*. **108**, 272 (1993).
5. M. H. Carpenter, D. Goettlieb, and S. Abarbanel, Stable and accurate boundary treatments for compact, high-order finite difference schemes, *Appl. Numer. Math*. **12**, 55 (1993).
6. M. H. Carpenter and C. A. Kennedy, *A Fourth Order 2N-Storage Runge–Kutta Scheme*, NASA TM 109112 (June 1994).
7. H. J. H. Clerex, A spectral solver for the Navier–Stokes equations in the velocity–vorticity formulation for flow with two nonperiodic directions, *J. Comput. Phys*. **137**, 186 (1997).
8. O. Daube, Resolution of the 2D Navier–Stokes equations in velocity–vorticity form by means of influence matrix technique, *J. Comput. Phys*. **103**, 402 (1992).
9. P. G. Drazin and W. H. Reid, *Hydrodynamic Stability* (Cambridge Univ. Press, Cambridge, UK, 1981).
10. M. Gaster, On the generation of spatially growing waves in the theory of hydrodynamic stability, *J. Fluid Mech*. **22**, 433 (1965).
11. T. B. Gatski, Review of incompressible fluid flow computations using the velocity–vorticity formulation, *Appl. Numer. Math*. **7**, 227 (1991).
12. G. H. Golub and C. F. Vanloan, *Matrix Computations* (The John Hopkins Univ. Press, Baltimore, 1996).
13. M. M. Gupta, J. Kouatchou, and J. Zhang, Comparisons of second- and fourth-order discretizations for multigrid Poisson solvers, *J. Comput. Phys*. **132**, 226 (1997).
14. R. Jordinson, The flat plate boundary layer. Part 1. Numerical integration of the Orr–Sommerfeld equation, *J. Fluid Mech*. **43**, 801 (1970).
15. Y. S. Kachanov, On the resonant nature of the breakdown of a laminar boundary layer, *J. Fluid Mech*. **184**, 43 (1987).
16. Y. S. Kachanov, V. V. Kozlov, V. Y. Levchenko, and M. P. Ramazanov, On the nature of K-breakdown of a laminar boundary layer; new experimental data, *Laminar-Turb. Transit*. **61** (1985).
17. L. Kleiser and U. Schumann, Spectral simulation of the laminar-turbulent transition process in plane Poiseuille flow, in *Spectral Methods for Partial Differential Equations* (SIAM, Philadelphia, 1984), p. 141.
18. L. Kleiser and T. A. Zang, Numerical simulation of transition in wall bounded shear flows, *Ann. Rev. Fluid Mech*. **23**, 495 (1991).
19. V. V. Kozlov and V. Y. Levchenko, Laminar-turbulent transition controlled by localized disturbances, in *Turbulent Management and Relaminarization* (Springer-Verlag, Berlin/New York, 1987), p. 249.
20. S. K. Lele, Compact finite difference schemes with spectral like resolution, *J. Comput. Phys*. **103**, 16 (1992).
21. Z. Liu and C. Liu, Fourth order finite difference and multigrid methods for modeling instabilities in flat plate boundary layers—2-D and 3-D approaches, *Comput. Fluids* **23**(7), 955 (1994).
22. S. A. Orzag and L. C. Kells, Transition to turbulence in plane Poiseuille flow and plane Couette flow, *J. Fluid Mech*. **96**, 159 (1980).
23. U. Rist and H. Fasel, Direct numerical simulation of controlled transition in a flat-plate boundary layer, *J. Fluid Mech*. **298**, 211 (1995).
24. R. S. Rogallo and P. Moin, Numerical simulation of turbulent flows, *Ann. Rev. Fluid Mech*. **16**, 99 (1984).

25. P. J. Schmid and D. S. Henningson, *Stability and Transition in Shear Flows*, Appl. Math. Sci. Vol. 142 (2000).
26. C. L. Streett and M. G. Macaraeg, Spectral multidomain for large scale fluid dynamics simulations, *Int. J. Appl. Numer. Math*. **6**, 123 (1989).
27. R. Vichnevetsky and J. B. Bowles, *Fourier Analysis of Numerical Approximations of Hyperbolic Equations* (SIAM, Philadelphia, 1982).
28. P. Wesseling, *An Introduction to Multigrid Methods* (Wiley, New York, 1992).
29. J. H. Williamson, Low-storage Runge–Kutta schemes, *J. Comput. Phys*. **35**, 48 (1980).
30. T. A. Zang and M. Y. Hussaini, Numerical simulation of nonlinear interactions in channel and boundary-layer transition, in *Nonlinear Wave Interactions in Fluids*, AMD-1987 (ASME, New York, 1987). p. 137.