JOURNAL OF COMPUTATIONAL PHYSICS **88**, 232–242 (1990)

Note

An Analytic Green's Functions Method in Pseudo-Spectral Navier–Stokes Solvers for Boundary Layer and Channel Flows

In pseudo-spectral simulations of a flow between rigid parallel plates with periodic boundary conditions in the horizontal directions, a number of different numerical methods are currently in use. These methods are briefly reviewed by Gottlieb *et al.* [1]. The differences among the methods amount essentially to different ways of imposing the incompressibility condition on the flow and this in turn is directly related to their efficiency in terms of computer storage and time required to perform simulations with a given spatial and temporal resolution. All these methods [2–8] treat the nonlinear terms in the Navier–Stokes equations explicitly and the pressure and viscous terms implicitly. The implicit part must be solved by inverting matrices resulting from spatial discretization of the pressure and viscous terms.

The most efficient methods in terms of computer time and storage are based on expansions of the velocity field into divergence free basis functions [4, 5]. Their disadvantage is that the basis functions must be constructed individually for each flow geometry and the inversion of the resulting matrices may require new algorithms for each geometry. For that reason, methods that can be reduced to solving sequences of standard Poisson and Helmholtz equations are more popular. Among these methods the most efficient is the full time splitting method of Orszag and Kells [8] that requires inversion of four N by N matrices for each horizontal wavenumber (k_x, k_y), where N is the number of mesh points between the plates. The matrices result from a sequence of four Poisson and Helmholtz equations and a variety of numerical schemes exist to acomplish inversions efficiently. The disadvantage of this method is that it violates incompressibility in a numerical boundary layer of thickness $O((\nu\,\Delta t)^{1/2})$ at the plates [6] and for this reason it is not used very frequently.

Incompressibility may be enforced by using the capacitance matrix algorithm of Kleiser and Schumann [7] or the equivalent Green's functions method of Marcus [3]. The Green's functions method of Marcus [3] may be implemented also with four Poisson solvers but ensures incompressibility at the expense of increasing storage requirements by two auxiliary arrays of size N^3. However, such storage is usually not available in high resolution numerical simulations and in commonly used Green's functions methods storage requirements are reduced to those in the full time splitting method, but required computer time increases since six instead of four Poisson equations must then be solved. Orszag *et al.* [6] discuss

0021-9991/90 $3.00

such a Green's functions method for a channel flow that requires six Poisson solvers. The capacitance matrix algorithm of Kleiser and Schumann [7] also requires six Helmholtz solvers. In essence, the Green's functions methods use either more computer time or storage than the full time splitting method. This additional work or storage is a consequence of a need to enforce incompressibility violated by the full time splitting method.

The purpose of this note is to demonstrate that for channel and for flat plate boundary layer flow the Green's functions method may be implemented with only four instead of six Poisson solvers per time step. The savings come from the observation that two of six Poisson equations may be solved analytically in terms of elementary functions. The implementation of the method is straightforward and existing pseudo-spectral computer codes may be easily modified bringing savings in the computer time. The modifications are described for both a channel and a boundary layer code.

Fluid is contained between two rigid parallel plates (channel flow) or above one horizontal plate (boundary layer flow). The z axis of the frame of reference is perpendicular to the plates. The velocity field is decomposed into a prescribed time-independent mean velocity $\mathbf{V}(z)=(U(z), 0, 0)$ in the x direction and a perturbation velocity $\mathbf{v}(x, y, z)=(u, v, w)$. The mean velocity is chosen to satisfy the boundary conditions for the entire flow, e.g., $U(z)$ is the Blasius profile for the boundary layer flow or a parabolic profile for the channel flow. Therefore the perturbation velocity satisfies homogeneous boundary conditions at the horizontal boundaries. With this decomposition the Navier–Stokes equations are

$$\frac{\partial \mathbf{v}}{\partial t}=\left[\mathbf{v}\times\boldsymbol{\omega}-w\frac{\partial \mathbf{V}(z)}{\partial z}\right]-U(z)\frac{\partial \mathbf{v}}{\partial x}-\nabla\Pi+\nu\nabla^2(\mathbf{v}+\mathbf{V}(z)) \tag{1a}$$

$$\nabla\cdot\mathbf{v}=0, \tag{1b}$$

where $\boldsymbol{\omega}=\nabla\times\mathbf{v}$ is the vorticity and $\Pi=p/\rho+1/2\mathbf{v}^2$ is the pressure head where p is the pressure, ρ is the density, and ν is the kinematic viscosity.

Equations (1) are solved by the following pseudo-spectral time splitting method consisting of three separate fractional steps which advance flow velocities from time t_n to t_{n+1}. In Eq. (1a) the nonlinear term is separated into two components. In the first fractional step, the component in the square brackets is calculated pseudo-spectrally and advanced in time using the explicit Adams–Bashforth scheme. In the second fractional step, the advection part of the nonlinear term is diagonalized by Fourier expansion and is solved by the implicit Crank–Nicolson scheme to reduce the convective stability restrictions due to the large mean flow $U(z)$. The intermediate velocities resulting from application of these two fractional steps will be denoted by asterisks. In the third fractional step, pressure and viscous terms with incompressibility (1b) are solved by the Green's functions method as described by Marcus [3] for Couette flow and outlined by Orszag *et al.* [6] for channel flow leading to the final velocities at time t_{n+1}. Using Fourier expansions in the horizontal directions for the dependent variable

$$\mathbf{v}(x, y, z, t) = \sum_{|m|<M} \sum_{|n|<N} \hat{\mathbf{u}}(k_x, k_y, z, t) \exp(ik_x x) \exp(ik_y y), \tag{2}$$

where $k_x = 2\pi m/L_x$, $k_y = 2\pi n/L_y$ are the horizontal wavenumbers and L_x and L_y are the periodicity lengths, the following equations in spectral space are obtained from the pressure and the viscous terms at the last fractional step

$$\frac{\partial \hat{u}}{\partial t} = -ik_x \hat{\Pi} + \nu \left(\frac{\partial^2}{\partial z^2} - k_x^2 - k_y^2 \right) (\hat{u} + \hat{U}(z)) \tag{3a}$$

$$\frac{\partial \hat{v}}{\partial t} = -ik_y \hat{\Pi} + \nu \left(\frac{\partial^2}{\partial z^2} - k_x^2 - k_y^2 \right) \hat{v} \tag{3b}$$

$$\frac{\partial \hat{w}}{\partial t} = -\frac{\partial}{\partial z} \hat{\Pi} + \nu \left(\frac{\partial^2}{\partial z^2} - k_x^2 - k_y^2 \right) \hat{w}, \tag{3c}$$

where $\hat{U}(z)$ is the horizontal Fourier transform of the mean velocity $U(z)$, which is nonzero only for $(k_x, k_y) = 0$. The continuity equation (1b) becomes

$$ik_x \hat{u} + ik_y \hat{v} + \frac{\partial \hat{w}}{\partial z} = 0. \tag{4}$$

Equations (3) are discretized in time using the Crank–Nicolson method for the viscous term and the full implicit method for the pressure term. Eliminating $\hat{\Pi}$ from (3a) and (3b), and using the incompressibility condition (4) we get an equation for $\hat{w}^{n+1}$, the vertical component of the velocity at time t_{n+1},

$$(D^2 - k^2)\left(D^2 - k^2 - \frac{2}{\nu \Delta t} \right) \hat{w}^{n+1}(k_x, k_y, z) = g(k_x, k_y, z), \tag{5}$$

where

$$g(k_x, k_y, z) = \left(D^2 - k^2 + \frac{2}{\nu \Delta t} \right) [D(ik_x \hat{u}^* + ik_y \hat{v}^*) + k^2 \hat{w}^*]. \tag{6}$$

In (5) and (6) $k^2 = k_x^2 + k_y^2$, $D = \partial/\partial z$ and velocities denoted by asterisks are the results of the first two fractional steps. Note that these velocities do not satisfy incompressibility since this condition is imposed only at the last fractional step, so that the velocity is divergence free after the full time step. Equation (5) must be solved for each horizontal wavenumber (k_x, k_y) with the boundary conditions $\hat{w}^{n+1} = 0$ and $D\hat{w}^{n+1} = 0$. The latter condition follows from (4). In boundary layer flow this condition is used only at the lower boundary $z=0$. Once $\hat{w}^{n+1}$ is found then the expression for $\hat{\Pi}^{n+1}$ is

$$\hat{\Pi}^{n+1} = -\frac{\nu}{2k^2}\left[(D^2-k^2)(ik_x\hat{u}^* + ik_y\hat{v}^* - D\hat{w}^{n+1}) + \frac{2}{\nu\,\Delta t}(ik_x\hat{u}^* + ik_y\hat{v}^* + D\hat{w}^{n+1})\right]. \tag{7}$$

The horizontal velocity components are obtained from the following Helmholtz equations

$$\left(D^2-k^2-\frac{2}{\nu\,\Delta t}\right)\hat{u}^{n+1} = \frac{2}{\nu\,\Delta t}\left[ik_x\,\Delta t\hat{\Pi}^{n+1} - \hat{u}^*\right] - (D^2-k^2)(\hat{u}^* + 2\hat{U}(z)) \tag{8}$$

$$\left(D^2-k^2-\frac{2}{\nu\,\Delta t}\right)\hat{v}^{n+1} = \frac{2}{\nu\,\Delta t}\left[ik_y\,\Delta t\hat{\Pi}^{n+1} - \hat{v}^*\right] - (D^2-k^2)\,\hat{v}^*. \tag{9}$$

In what follows let us assume that we have efficient numerical solvers for the channel and the boundary layer flow geometry for the Poisson equation

$$(D^2-k^2)\,f(k_x, k_y, z) = g(k_x, k_y, z) \tag{10a}$$

and the Helmholtz equation

$$\left(D^2-k^2-\frac{2}{\nu\,\Delta t}\right)f(k_x, k_y, z) = g(k_x, k_y, z) \tag{10b}$$

with homogeneous Dirichlet boundary conditions. With these solvers it is a straightforward task to solve Eqs. (8) and (9) for $\hat{u}^{n+1}$ and $\hat{v}^{n+1}$. Equation (5) will be solved by a modified Green's functions method.

Boundary Layer Flow

Equation (5) must be solved for all pairs of horizontal wavenumbers (k_x, k_y) in the domain $z \in [0, \infty)$ with the boundary conditions

$$\hat{w}(0) = \hat{w}(\infty) = D\hat{w}(0) = 0. \tag{11}$$

Note that in (11) and in all subsequent formulae explicit dependence of various functions on k_x, k_y is often omitted if it does not lead to confusion. The solution of (5) is obtained as

$$\hat{w}(z) = b_-\,w_-(z) + w_0(z), \tag{12}$$

where w_0 is obtained by solving the sequence of equations

$$\left(D^2-k^2-\frac{2}{\nu\,\Delta t}\right)\zeta_0(z)=g(z); \qquad \zeta_0(0)=\zeta_0(\infty)=0 \tag{13a}$$

$$(D^2-k^2)\,w_0(z)=\zeta_0(z); \qquad w_0(0)=w_0(\infty)=0 \tag{13b}$$

and $w_-(z)$ is obtained by solving

$$\left(D^2-k^2-\frac{2}{\nu\,\Delta t}\right)\zeta_-(z)=0; \qquad \zeta_-(0)=1, \zeta_-(\infty)=0 \tag{14a}$$

$$(D^2-k^2)\,w_-(z)=\zeta_-(z); \qquad w_-(0)=w_-(\infty)=0. \tag{14b}$$

Constant b_- is determined in such a way that the boundary condition (11) for $D\hat{w}(z)$ is satisfied,

$$b_- = -Dw_0(0)/Dw_-(0). \tag{15}$$

In the Green's functions method described in [6] if the function w_- (the Green's function for this problem) is found in the preprocessing step and is stored, the solution of (5) requires solution of the two equations (13a) and (13b). If storage is limited only the boundary values $Dw_-(0)$ are stored. To get $\hat{w}$, after finding w_0 from (13) and b_- from (15) Eqs. (13) are solved once more with the following boundary conditions:

$$\hat{\zeta}(0)=b_-, \qquad \hat{\zeta}(\infty)=0; \qquad \hat{w}(0)=0, \qquad \hat{w}(\infty)=0. \tag{16}$$

Thus finding $\hat{w}$ is equivalent to solving numerically four Poisson equations.

A simplification of the Green's function method discussed here is based on the observation that Eqs. (14) for the Green's functions may be solved analytically. Indeed, the solution to Eq. (14a) is

$$\zeta_-(z)=\exp(-\kappa z), \tag{17}$$

where $\kappa=(k^2+2/(\nu\,\Delta t))^{1/2}$. A solution to Eq. (14b) is sought as

$$w_-(z)=Ae^{-\kappa z}+Be^{-kz}. \tag{18}$$

Equation (14b) and the boundary conditions are satisfied if

$$A=-B=\nu\,\Delta t/2. \tag{19}$$

The derivative of w_- at the boundary needed in (15) is

$$Dw_-(0)=0.5\nu\,\Delta t(k-\kappa). \tag{20}$$

After numerically solving the two equations (13a) and (13b), the complete solution $\hat{w}(z)$ is found from (12), (15), and (18)–(20). The work required is therefore equivalent to solving numerically only two Poisson equations.

Channel Flow

In the case of the channel flow, Eq. (5) must be solved in the domain $z \in [-1, +1]$ with the following boundary conditions

$$\hat{w}(-1) = \hat{w}(+1) = D\hat{w}(-1) = D\hat{w}(+1) = 0. \tag{21}$$

In the Green's functions approach [6] this is done by representing the solution $\hat{w}$ as

$$\hat{w}(z) = b_- w_-(z) + b_+ w_+(z) + w_0(z), \tag{22}$$

where w_0 is obtained from Eqs. (13) with zero boundary conditions at $z = \pm 1$, and the functions $w_\pm$ are solutions of the following sequence of equations

$$\begin{gathered} \left(D^2 - k^2 - \frac{2}{\nu\, \Delta t}\right) \zeta_\pm(z) = 0; \qquad \zeta_-(-1) = \zeta_+(+1) = 1; \\ \zeta_+(-1) = \zeta_-(+1) = 0; \end{gathered} \tag{23a}$$

$$(D^2 - k^2)\, w_\pm = \zeta_\pm(z); \qquad w_\pm(\pm 1) = 0. \tag{23b}$$

Constants $b_\pm$ will be determined from the boundary conditions for $D\hat{w}(\pm 1)$ (Eq. (21)).

As in the case of the boundary layer flow Eqs. (23) may be solved analytically. The solution of (23a) is

$$\zeta_\pm(z) = A_\pm e^{-\kappa z} + B_\pm e^{+\kappa z}, \tag{24}$$

where $\kappa = (k^2 + 2/(\nu\, \Delta t))^{1/2}$, and

$$A_- = \frac{1}{e^{\kappa} - e^{-3\kappa}}, \qquad B_- = \frac{1}{e^{-\kappa} - e^{3\kappa}}, \qquad A_+ = B_-, \qquad B_+ = A_-. \tag{25}$$

A solution to Eq. (23b) is sought in the following form

$$w_\pm(z) = C_\pm e^{+\kappa z} + D_\pm e^{-\kappa z} + E_\pm e^{+kz} + F_\pm e^{-kz}. \tag{26}$$

The constants in (26) are determined from Eq. (23b) and its boundary conditions

$$C_- = \frac{\nu \Delta t/2}{e^{-\kappa} - e^{3\kappa}}, \qquad D_- = \frac{\nu \Delta t/2}{e^{\kappa} - e^{-3\kappa}} \qquad \text{(27a), (27b)}$$

$$E_- = -\frac{\nu \Delta t/2}{e^{-k} - e^{3k}}, \qquad F_- = -\frac{\nu \Delta t/2}{e^{k} - e^{-3k}} \qquad \text{(27c), (27d)}$$

$$C_+ = D_-, \qquad D_+ = C_-, \qquad E_+ = F_-, \qquad F_+ = E_-. \qquad \text{(27e)}$$

The boundary conditions (21) are used to get a system of linear equations for $b_\pm$,

$$b_+ Dw_+(\pm 1) + b_- Dw_-(\pm 1) = -Dw_0(\pm 1). \qquad (28)$$

Using symmetries (27e) we get

$$Dw_-(\pm 1) = -Dw_+(\mp 1) \qquad (29)$$

and the explicit expression for $b_\pm$ is obtained from (28)

$$b_\pm = \frac{(-Dw_+(\pm 1)\, Dw_0(+1) + Dw_+(\mp 1)\, Dw_0(-1))}{(Dw_+(+1)^2 - Dw_+(-1)^2)}. \qquad (30)$$

Expressions for $Dw_+(\pm 1)$ are gotten from (26),

$$Dw_+(-1) = \frac{1}{2} \nu \Delta t \left[\frac{2\kappa}{e^{2\kappa} - e^{-2\kappa}} - \frac{2k}{e^{2k} - e^{-2k}} \right] \qquad \text{(31a)}$$

$$Dw_+(+1) = \frac{1}{2} \nu \Delta t \left[\kappa \left(\frac{1}{1 - e^{-4\kappa}} - \frac{1}{1 - e^{4\kappa}} \right) - k \left(\frac{1}{1 - e^{-4k}} - \frac{1}{1 - e^{4k}} \right) \right]. \qquad \text{(31b)}$$

Note that the solution procedure must be modified for the wavenumber $k = 0$, since in this case a solution to (23b) is

$$w_\pm(z) = C_\pm e^{\kappa z} + D_\pm e^{-\kappa z} + E_\pm z + F_\pm. \qquad (32)$$

In (32) the constants $C_\pm$, $D_\pm$ are given by (27a), (27b), and (27e) and

$$E_- = -F_- = \tfrac{1}{4} \nu \Delta t, \qquad E_+ = F_+ = -\tfrac{1}{4} \nu \Delta t. \qquad (33)$$

The complete solution is determined by Eq. (22), where w_0 is obtained by numerically solving the two equations (13), $w_\pm$ are given by (26) (or (32) for $k = 0$), and constants $b_\pm$ are given by Eq. (30). Formula (30) involves derivatives of the exact solution w_+ and the solution w_0 which are calculated numerically with necessarily finite accuracy. For the vertical resolution $N < 32$ modes combining derivatives of the analytical and the numerical solutions led to slight errors in the constants $b_\pm$. In such a case it is advisable to use in (30) boundary derivatives of the function w_+ computed numerically in the pre-processing step. This results in cancellation of errors and improved accuracy for the constants $b_\pm$. For $N > 64$ this method does not improve accuracy any further and formulas (30) and (31) should be used.

TABLE I

Comparison between Growth Rates of the Tollmien–Schlichting Waves Obtained from the Orr–Sommerfeld Equation (ω_{OS}) and from the Navier–Stokes Solver (ω_{NS}).

Re	ω_{OS}	ω_{NS}	ε
1500	−0.0010049	−0.0010048	10^{-4}
1800	−0.0000991	−0.0000990	10^{-3}
2100	+0.0005267	+0.0005269	3×10^{-4}

Note. The relative error $\varepsilon = (\omega_{NS} - \omega_{OS})/\omega_{OS}$.

Numerical Example

To test the above method a pseudo-spectral boundary layer code used previously by Domaradzki and Metcalfe [9] was modified according to these ideas. The code used in [9] was derived from the channel code of Orszag and Kells [8] and uses the full time splitting method which is known to violate the incompressibility condition. For this reason it generally produces results that are significantly less accurate than results obtained by numerical codes that satisfy incompressibility. A standard test of the accuracy of a boundary layer and a channel flow code is made by comparing growth rates of modes calculated from the Orr–Sommerfeld equation with growth rates of the same modes predicted by the Navier–Stokes code.

We have performed such tests for the modified code. The Navier–Stokes simulations were initialized with the velocity fields obtained from the most unstable mode of the Orr–Sommerfeld equation for a wavenumber $\alpha = 1.0$ and three different Reynolds numbers (based on the boundary layer thickness defined as $\delta = 6.02(\nu x/U_0)^{1/2}$). The amplitude of the wave was chosen as 10^{-5} of the free stream velocity U_0, so that the nonlinear effects were small. Simulations were two-dimensional with 65 mesh points in the vertical and 8 points in the horizontal direction. In Table I the comparison is presented between growth rates of the Tollmien–Schlichting waves obtained from the Orr–Sommerfeld solver and results of numerical solution of the Navier–Stokes equation after 50 time steps. For all three cases the growth rates predicted by the Navier–Stokes solver agree with the growth rates calculated from the Orr–Sommerfeld equation up to six significant figures. This accuracy matches the accuracy of results obtained by Marcus [3] in a similar test problem for his divergence free code for Couette flow (see Table I in [3]). Also the relative error ε is generally about two orders of magnitude less than the error observed in simulations performed with the full time splitting method used in [9] for the boundary layer and in [8] for the channel flow.

In Table II timings for the analytical and numerical Green's functions methods are presented for the boundary layer code run with a resolution of 64^3 modes on

TABLE II

CPU Timings (in Seconds) for Different Implementations of the Green's Functions Method

Method	Full step	3rd step	Poisson solver
NA	12.20	9.37	0.50
AA	11.03	8.25	0.50
NF	19.26	16.34	1.64
AF	15.40	12.58	1.58

Note. NA-numerical with an assembly Poisson solver; AA-analytical with an assembly Poisson solver; NF-numerical with a Fortran Poisson solver; AF-analytical with a Fortran Poisson solver. Timings are for the full time step, the third fractional step (the pressure and viscous step), and the Poisson solver.

the Cray X-MP. Using the analytical Green's functions reduces required CPU time by about 10% (time needed to solve two Poisson equations) as compared with the methods discussed in [6, 7] which solve all Poisson equations numerically. The above estimate should be considered as a lower bound since we used a highly optimized Poisson solver coded in the Assembly language on the Cray X-MP. For the same Poisson solver coded in Fortran savings in computer time are about 20% when using the analytical Green's functions method instead of the numerical one, since two Poisson solvers saved constitute larger portion of the full time step than in the previous case. Similar savings were also observed after applying the analytical Green's functions methods to modify the numerical code used by Domaradzki and Metcalfe [10] to simulate Rayleigh-Benard convection between two rigid plates. The numerical Navier–Stokes solver in [10] uses the Green's functions method of reference [6]. How much time will precisely be saved in any particular case depends on details of a numerical code since savings are equivalent to time needed to solve two Poisson equations and this time may vary among different codes. In high resolution numerical simulations an increase by 10–20% in an efficiency of a numerical code may translate into hours of supercomputer time saved per run.

The incompressibility was checked by comparing individual terms $\partial u/\partial x$, $\partial v/\partial y$, $\partial w/\partial z$ with their sum, which should be equal to zero. At the first mesh point away from the boundary $z=0.00018\delta$ the relative error is 2%, dropping to 0.03% at the second mesh point $z=0.0007\delta$, and to 0.001% at the third point $z=0.0017\delta$. This error in the divergence at the boundary is caused entirely by an inaccuracy of the order 10^{-10} in the numerical calculation of the z-derivative in that region. Close to the boundary derivatives of the velocity are all of the order 10^{-8} so that the absolute error of the order 10^{-10} has an appreciable effect on the accuracy of the divergence. The relative error in the divergence becomes uniformly less than 10^{-8} for $z>0.015\delta$.

Conclusions

We have shown that the Green's functions method [3, 6, 7] for the channel and the flat plate boundary layer flow may be modified by solving analytically several equations that are usually solved numerically. This modification reduces from six to four the number of Poisson and Helmholtz equations that must be solved numerically at each time step. For a typical pseudo-spectral Navier–Stokes solver this modifications saves about 10% of computer time as compared with the original method that solves all equations numerically. The amount of required computer time may be reduced even further as follows. If the vertical component of velocity w is known and one of the horizontal components of velocity (u) is determined from (8) then the other horizontal component may be obtained from the incompressibility condition instead of Poisson Eq. (9). This reduces to three the total number of Poisson solvers needed per time step. The implementation of the Green's functions method described in this note is especially attractive in modifying existing pseudo-spectral codes that use either the Green's functions method [6] or the equivalent capacitance matrix technique [7]. It may also be attractive in the development of new codes since an algorithm to solve efficiently Eqs. (1) may be constructed from standard numerical building blocks: Fast Fourier Transform subroutines to calculate the nonlinear terms and Poisson and Helmholtz solvers (with homogeneous Dirichlet boundary conditions) to treat viscous and pressure effects. The existence of the analytical Green's functions in Navier–Stokes solvers for simple flat plate and channel flow geometry also suggests that their existence in more complicated geometries should be investigated.

Acknowledgments

This research was supported by the Office of Naval Research Contract N00014-88-K-0157 and a Research Initiation Grant from the Powell Foundation. Computer time on Cray X-MP was provided by the San Diego Supercomputer Center.

References

1. D. Gottlieb, M. Y. Hussaini, and S. A. Orszag, in *Spectral Methods for Partial Differential Equations*, edited by R. G. Voigt *et al.* (SIAM, Philadelphia, 1984), p. 1.
2. P. Moin and J. Kim, *J. Comput. Phys.* **35**, 381 (1980).
3. P. S. Marcus, *J. Fluid Mech.* **146**, 45 (1984).
4. R. D. Moser, P. Moin, and A. Leonard, *J. Comput. Phys.* **52**, 524 (1983).
5. P. R. Spalart, NASA Technical Memorandum 88222 (1986).
6. S. A. Orszag, M. Israeli, and M. O. Deville, *J. Sci. Comput.* **1**, 75 (1986).
7. L. Kleiser and U. Schumann, in *Spectral Methods for Partial Differential Equations*, edited by R. G. Voigt *et al.* (SIAM, Philadelphia, 1984), p. 141.

8. S. A. Orszag and L. C. Kells, *J. Fluid Mech.* **96**, 159 (1980).
9. J. A. Domaradzki and R. W. Metcalfe, *Phys. Fluids* **30**, 695 (1987).
10. J. A. Domaradzki and R. W. Metcalfe, *J. Fluid Mech.* **193**, 499 (1988).

Received: June 6, 1988; revised: April 14, 1989

J. Andrzej Domaradzki

Department of Aerospace Engineering,
University of Southern California,
Los Angeles, California 90089-1191