CONF-890752--1

SAND--89-0046C

DE89 009333

Numerical Solution of the Navier-Stokes Equations for Three-Dimensional Incompressible Flows with Open Boundaries*

J. A. Schutt

Fluid Mechanics and Heat Transfer Division III
Sandia National Laboratories
Albuquerque, NM 87185

INTRODUCTION

Numerical modelling of the transient, nonisothermal, three-dimensional, incompressible Navier-Stokes equations imposes extreme demands on computational resources. For incompressible flow the Navier-Stokes equations form a coupled parabolic-elliptic set, with the elliptic nature caused by the incompressibility constraint. The elliptic nature of the incompressibility constraint forces some portion of the solution algorithm to be implicit, with the attendant computational costs. In addition, in regions of the computational domain where advective effects overwhelm viscous effects, the Navier-Stokes equations exhibit behavior which is similar to the hyperbolic nature of the Euler (inviscid) equations. In such regions many numerical methods which are suitable for parabolic equations will be dispersive, causing oscillations to appear in the solution. If accurate long-time solutions are required, dispersion must be avoided, as well as excessive numerical diffusion, which can result from attempts to control dispersion.

The numerical algorithm presented here attempts to address these issues as follows. Following work presented by Kim and Moin [1], an operator split is performed on the incompressible Navier-Stokes equations, separating them into hyperbolic (advective effects), parabolic (diffusive or viscous effects), and elliptic (incompressibility constraint) components. Each operator is treated with a numerical method which addresses the particular character of that operator. Thus, advective terms are temporally integrated using flux-corrected transport (FCT) [2-5], an explicit method. The diffusive terms are implicitly integrated, with the

*This work performed at Sandian National Laboratories supported by the U.S. Department of Energy under Contract DE-AC04-76DP00789

---

resulting equation set solved using a preconditioned conjugate gradient (PCG) technique. Imposition of the incompressibility constraint results in a Poisson equation, which is also solved using PCG. This operator split has the additional benefit that the equations are uncoupled and linearized at each time step, reducing memory and execution time requirements for the implicit solution of diffusive effects.

NUMERICAL METHOD

Consider the case of a nonisothermal, incompressible Newtonian fluid with constant properties. If the Boussinesq approximation [6] is invoked so that the flow may be driven by buoyancy forces, the Navier-Stokes and energy equations may be manipulated into the form

$$\frac{\partial u_i}{\partial t} + \frac{\partial}{\partial x_j}(u_j u_i) = -\frac{1}{\rho_o}\frac{\partial p'}{\partial x_i} - \beta (T - T_o) g_i + \frac{\mu}{\rho_o}\frac{\partial}{\partial x_j}\frac{\partial u_i}{\partial x_j} \quad , \tag{1}$$

$$\frac{\partial T}{\partial t} + \frac{\partial}{\partial x_j}(u_j T) = Q + \frac{\mu}{\rho_o c_p}\frac{\partial}{\partial x_j}\frac{\partial T}{\partial x_j} \quad , \tag{2}$$

where Q is a volumetric heat source, and $p' = p - p_o$, with the hydrostatic pressure, p_o, given by $\partial p_o / \partial x_i = \rho_o g_i$. If stratified flows are modelled, then the reference density $\rho_o = \rho_o(x_i)$ and temperature $T_o = T_o(x_i)$ can be functions of position only.

The numerical techniques utilized in this work can most easily be discussed by noting that each of the Eqs. (1) and (2) can be written in the form

$$\frac{\partial \phi}{\partial t} + \frac{\partial}{\partial x_j}(u_j \phi) = \gamma \frac{\partial}{\partial x_j}\frac{\partial \phi}{\partial x_j} + S_\phi \quad , \tag{3}$$

where ϕ is either u_i or T. Note that in the case of the momentum equations, the pressure gradient term $\frac{1}{\rho_o}\frac{\partial p'}{\partial x_i}$ is not included in the source term S_ϕ, but is treated in a manner to be described below.

In order to implement the advective-diffusive operator split, FCT with leapfrog time differencing is applied to the advective operator, including the source term S_ϕ, in order to calculate intermediate fields (denoted by $\hat{}$). This step can be represented by

$$\frac{\hat{\phi} - \phi^{n-1}}{2\Delta t} = -\frac{\delta}{\delta x_j}(u_j \phi)^n + S_\phi^n \quad , \tag{4}$$

where the superscripts n and $n-1$ denote time levels. Discrete difference operators are represented by $\delta / \delta x_j$, which in the case of Eq. (4) represents the FCT algorithm. Next, the intermediate advected fields are

diffused using a trapezoidal (*i.e.*, Crank-Nicolson) time integration step, with centered, second-order differencing used for the spatial derivatives:

$$\frac{\phi^{n+1} - \hat{\phi}}{2\Delta t} = \frac{\gamma}{2}\left[\frac{\delta}{\delta x_j}\frac{\delta \phi^{n-1}}{\delta x_j} + \frac{\delta}{\delta x_j}\frac{\delta \phi^{n+1}}{\delta x_j} \right] . \tag{5}$$

Note that the combination of Eqs. (4) and (5) is second-order accurate in time. For the first time step, or after a change in the time step size, Eq. (4) is implemented as an explicit Euler step starting at time level n, rather than a leapfrog step starting at time level $n-1$, and Eq. (5) is implemented as a trapezoidal step from time level n to $n+1$. This provides the two time levels needed to start the second order, leapfrog-trapezoidal time integration.

When centered, second order spatial differencing is applied to Eq. (5), a set of equations linear in ϕ are produced, which can be written as $A\ \vec{\phi}^{n+1} = \vec{b}$. Assuming that the mesh is orthogonal, A is a seven-diagonal, symmetric positive definite matrix, and $\vec{b}$ contains terms from the time differencing, and spatial differencing at time level $n-1$. This linear system of equations can be effectively solved using a point-Jacobi preconditioned conjugate gradient (PJPCG) method [7-8]. PCG methods converge much faster that other iterative methods such as successive over-relaxation (SOR), and require much less memory than direct methods. PJPCG was also chosen because it can be completely vectorized.

An added advantage of the operator split in this algorithm is that the inviscid Euler equations can be solved merely by bypassing the second step of the algorithm, Eq. (5). In this case, $\hat{\phi}$ becomes ϕ^{n+1}, and second-order time accuracy is retained.

When Eqs. (4) and (5) are applied to the momentum equations, the implementation is modified, following the work of Kim and Moin [1]. In this case, Eqs. (4) and (5) are applied without regard for pressure effects, resulting in a velocity field at time level $n+1$ which is not divergence-free. Call this provisional velocity field u_i^*. In order to satisfy continuity, u_i^* must be corrected to yield u_i^{n+1} according to

$$\frac{u_i^{n+1} - u_i^*}{\Delta t} = -\frac{\delta \Phi^{n+1}}{\delta x_i} , \tag{6}$$

where Φ^{n+1} is a scalar potential field calculated from

$$\frac{\delta}{\delta x_i}\frac{\delta \Phi^{n+1}}{\delta x_i} = \frac{1}{\Delta t}\left(\frac{\delta u_i^*}{\delta x_i} - \frac{\delta u_i^{n+1}}{\delta x_i} \right) . \tag{7}$$

Because the flow is incompressible, u_i^{n+1} should satisfy

$$\frac{\delta u_i^{n+1}}{\delta x_i} \equiv 0 . \tag{8}$$

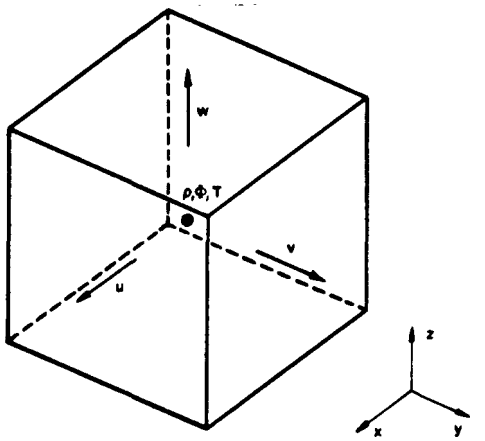

Figure 1: Representative computational cell for the three-dimensional staggered mesh.

When centered differences on an orthogonal mesh are used to form Eq. (7), a set of linear equations $A\,\vec{\Phi}^{n+1} = \vec{b}$ is again obtained, where A has the same characteristics described earlier. Here also, the linear system is solved using PJPCG. Note that the degree to which Eq. (8) is satisfied depends directly on the accuracy of the solution to Eq. (7), which will not be exact because PCG is an iterative method. A measure of the RMS error in Φ is given by $\left|\Delta\vec{\Phi}\right|\left|\vec{\Phi}\right|^{-1}$; a value of 10^{-6} was used as the convergence criteria for the PCG solver.

This method is implemented using a staggered mesh, with a typical computational cell shown in Figure 1. In the staggered mesh, velocity components are defined on cell faces normal to their respective coordinate directions, and ρ, Φ, and T are defined at the grid points representing cell centers. If the mesh is nonuniform, grid points and cell faces are defined such that cell faces are located halfway between grid points. The computational domain is tiled with cells such that cell faces lie on the boundaries of the domain. This mesh definition implies that grid points, and hence quantities defined at grid points, never lie on the domain boundaries. Thus, "ghost points" are defined outside the domain boundary to simplify the application of boundary conditions.

To apply FCT to the advection of the velocity field, new computational cells are defined for each component, where these cells are offset half a cell width in their respective coordinate direction, according to the velocity component being advected. The advecting velocity field for these offset cells is calculated by averaging.

BOUNDARY CONDITIONS

Discussion of boundary conditions for this algorithm is complicated by the multiple operator splits. To simplify the discussion, the general

case will be treated first, with exceptions noted afterward.

As mentioned earlier, the advective step of the algorithm, Eq. (4), along with the velocity correction given by Eqs. (6) and (7), forms an acceptable algorithm for solution of the incompressible Euler equations. Thus, boundary conditions for the advective step are imposed as though the incompressible Euler equations were, in fact, being solved. For the incompressible Euler equations, information is passed along only one characteristic, defined by the local flow velocity. Thus, for all variables, Dirichlet boundary conditions may be specified only at inflow boundaries during the advective step of the algorithm. At all other boundaries, zero gradients are imposed for all variables. (The component of velocity normal to a boundary represents a special case, to be discussed later.)

All other Dirichlet boundary conditions, as well as those imposed at inflow boundaries, are imposed during the diffusive step of the algorithm. Boundary conditions that involve specifying a nonzero gradient (*e.g.* heat flux) are also applied only during the diffusive step.

The choice of appropriate boundary conditions for the provisional velocity fields is complicated by the fact that they are not required to satisfy continuity. This is an important point because it is through the incompressibility constraint that the entire flow field adjusts to the presence of impermeable and/or no-slip boundaries.

What is needed, then, is some way of estimating the error likely to arise in the provisional velocity fields as a result of ignoring the incompressibility constraint. Such an estimate exists in the form of gradients of Φ^n, which, according to Eq. (6), gives the difference between the provisional velocity field and the continuity-satisfying velocity field at time level n. Thus, boundary conditions for the provisional velocity fields are set from

$$\hat{u}_i = u_i^{n+1} + \Delta t \, \frac{\delta \Phi^n}{\delta x_i} \quad , \tag{9}$$

where u_i^{n+1} represents the desired velocity boundary condition at time level $n+1$. In keeping with the previous discussion, during the advective step Eq. (9) is applied only to the velocity component normal to boundaries, and zero gradient boundary conditions are applied to velocity components parallel to boundaries. During the diffusive step all fixed-value velocity conditions are applied by setting values of the provisional velocity u_i^* according to Eq. (9).

Boundary conditions for the scalar potential field, Φ^{n+1}, must be imposed so that global continuity is enforced. The appropriate boundary condition for boundaries with specified velocities can be derived by applying Gauss' Divergence Theorem to the integral of Eq. (7) over the

computational domain, which results in the condition that

$$\int_S n_i \frac{\partial \Phi^{n+1}}{\partial x_i} ds = \frac{1}{\Delta t} \int_S n_i \left(u_i^* - u_i^{n+1} \right) ds \quad , \tag{10}$$

where S represents the domain boundary and n_i represent direction cosines of the outward normal. One way to guarantee that Eq. (9) is satisfied is to require that the integrands be equal at every point on the boundary, which gives the boundary condition on Φ^{n+1}:

$$\frac{\partial \Phi^{n+1}}{\partial x_i} = \frac{1}{\Delta t} \left(u_i^* - u_i^{n+1} \right) \quad . \tag{11}$$

Together Eqs. (7) and (11) specify Φ^{n+1} to an arbitrary, additive constant. In order to fix the value of that constant, a Dirichlet boundary condition with value zero is set at one point on the boundary.

Specification of boundary conditions for domains with open boundaries is complicated by the fact that the velocity through such boundaries is not known *a priori*. However, a treatment for open boundaries may be derived by recognizing two basic facts.

First, the purpose of such a boundary is, by definition, to decouple the computational domain from the world at large. Nothing interesting should be happening at such a boundary; otherwise, the boundary is in the wrong place. Thus, at such a boundary it should suffice to specify zero gradients for all fields for which some other physical principle does not require an alternative boundary condition. This is the case for all fields except the scalar potential field.

Second, any treatment of an open boundary must ensure global mass conservation. Since Eqs. (6) and (7) together ensure that mass will be conserved on a local scale, this second fact suggests that Eqs. (10) may be used to derive boundary conditions which will ensure that mass will be conserved globally when open boundaries are used.

Thus, the following treatment can be used for the velocity and scalar potential fields at open boundaries. During both the advection and diffusion steps of the algorithm, zero gradient velocity boundary conditions are applied to the provisional velocity fields $\hat{u}_i$ and u_i^*. Now, the global divergence error in u_i^* is given by

$$\mathcal{E} = \int_S n_i u_i^* ds \quad . \tag{12}$$

Partition the boundary S into two parts, one with fixed velocity and the other open, called S_f and S_o, respectively. Then, $\partial \Phi^{n+1} / \partial x_i$ (denoted by $\Phi_{x_i}^{n+1}$) is known on S_f, according to Eq. (11). On S_o, we wish to find

some average $\Phi_{x_i}^{n+1}$ which will enforce global mass conservation. Thus,

$$\begin{aligned}\frac{\mathcal{E}}{\Delta t} &= \int_S n_i \Phi_{x_i}^{n+1} ds \\ &= \frac{1}{\Delta t}\int_{S_f} n_i \left(u_i^* - u_i^{n+1}\right) ds + \int_{S_o} n_i \left(\Phi_{x_i}^{n+1}\right)_{S_o,ave} ds \quad ,\end{aligned}$$

or

$$\left|\left(\Phi_{x_i}^{n+1}\right)_{S_o,ave}\right| \int_{S_o} n_i \eta_i \, ds = \frac{1}{\Delta t}\left(\mathcal{E} - \int_{S_f} n_i \left(u_i^* - u_i^{n+1}\right) ds\right) \quad ,$$

where η_i is defined for any point on S by:

$$\eta_i = \Phi_{x_i}^{n+1} \left|\Phi_{x_i}^{n+1}\right|^{-1} \quad .$$

Finally,

$$\left|\left(\Phi_{x_i}^{n+1}\right)_{S_o,ave}\right| = \frac{1}{A_o \Delta t}\left\{\mathcal{E} - \int_{S_f} n_i \left(u_i^* - u^{n+1}\right) ds\right\} \quad . \tag{13}$$

Note that $A_o = \int_{S_o} n_i \eta_i ds$ can have either sign since $\left|\left(\Phi_{x_i}^{n+1}\right)_{S_o,ave}\right|$ must be positive. In fact, if $A_o < 0$ then the provisional velocity field u_i^* has excess inflow; if $A_o > 0$, then u_i^* has excess outflow. Finally, the value of $\left|\left(\Phi_{x_i}^{n+1}\right)_{S_o,ave}\right|$ given by Eq. (13) is used to set boundary conditions for Φ^{n+1} on $S_o$according to

$$n_i \Phi_{x_i}^{n+1} = \frac{A_o}{|A_o|}\left|\left(\Phi_{x_i}^{n+1}\right)_{S_o,ave}\right| \quad . \tag{14}$$

EXAMPLE COMPUTATIONS

Flow over a backward-facing step was chosen as a test problem for several reasons. The position of the reattachment point depends directly on Reynolds number, and so can provide an assesment of the impact of numerical diffusion. The multiple separation bubbles which develop at moderate Reynolds numbers also provide a difficult test for outflow boundary conditions if such a boundary is placed near the separation regions. Also, experimental data for this case is present in the literature.

To provide a significant degree of verification of the present method at minimal cost, two-dimensional simulations of flow over the backward facing step at various Reynolds numbers were computed. These results were compared with the experimental data of Armaly *et. al.* [9], and the numerical results of Kim and Moin. Both investigators agree that for

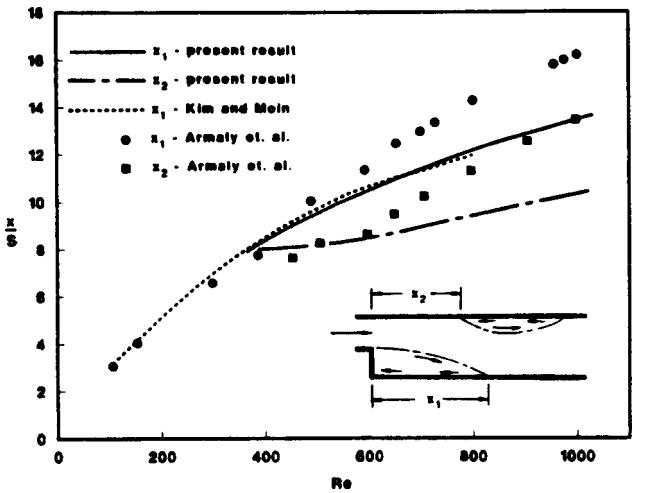

Figure 2: Comparison of separation and reattachment points for two-dimensional computations with data.

Reynolds numbers larger than ~600, three-dimensional effects become important. To verify this assertion, a three-dimensional simulation for Re = 800 was performed, and the results for reattachment length compared to data and two-dimensional simulations.

The two-dimensional cases were computed using a 100×35 mesh for a domain 20 step-heights in length. The mesh was uniform in the longitudinal (x) direction, and nonuniform in the vertical (z) direction, with grid spacing at the downstream duct centerline three times that near the walls. The ratio of step height to duct height was 0.485, corresponding to Armaly *et. al.*'s experiments. A parabolic velocity profile was used as the inlet boundary condition. For both the two- and three-dimensional simulations, Re was calculated based on two-thirds of the maximum velocity and twice the duct height (corresponding to mass-average velocity, U_o, and hydraulic diameter, D_h, respectively, for 2D flows.)

For 400<Re<1000, Armaly *et. al.*'s data show that the flow is characterized by a primary separation bubble behind the step, and a secondary separation bubble on the wall opposite the primary. Figure 2 shows a comparison of reattachment (primary) and separation (secondary) positions, normalized by the step height, S, for the results of this method, the data of Armaly *et. al.*, and the computations of Kim and Moin. Predictions of the reattachment length agree quite well with those of Kim and Moin, who used a 101×101 mesh on a domain $30S$ long. However, in both cases predictions begin to depart from data at about Re = 600. Predictions using the present method for the separation point of the secondary recirculation zone show good agreement with data, although the rate of increase of separation point with Re does not match exactly that shown by the data.

To check the solution for mesh independence, the Re = 800 case was

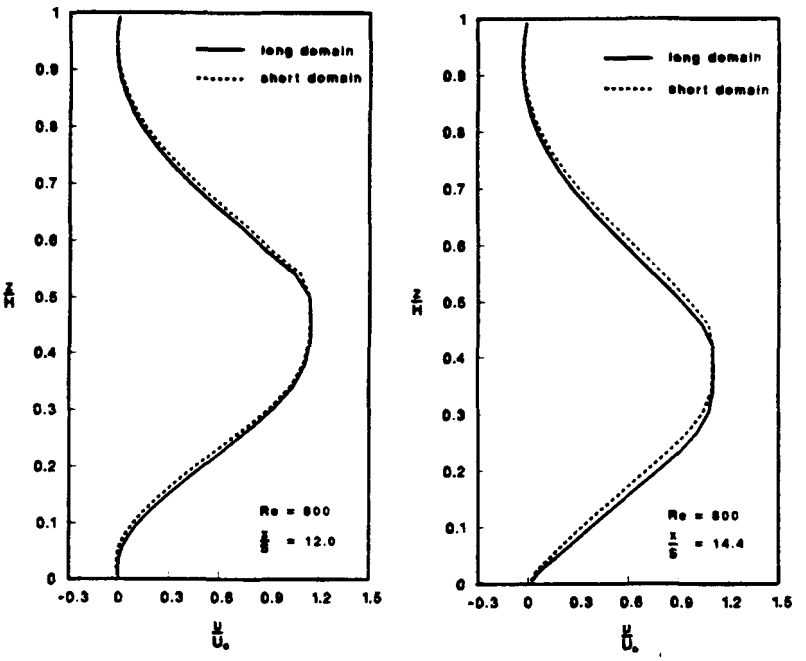

Figure 3: Effect of shortened domain and boundary conditions for open boundaries on streamwise velocity.

recomputed using a 150×70 mesh, again with $\Delta z_{max}/\Delta z_{min} = 3$. Results for the normalized reattachment and separation points were 12.25 and 9.75, a change of 1% and 3%, respectively.

To check for effects of the boundary conditions used at the open boundary, the Re = 800 case was recomputed again on a domain $14.4S$ long, using a 72×35 mesh. This gave the same mesh spacing as the base Re = 800 case, while truncating the secondary separation bubble at its midpoint. Results for the normalized reattachment and separation points were 12.35 and 9.5, for a change of 1.8% and 1.4%, respectively, from the base case. Profiles of the streamwise velocity for $x/S = 12.0$ and $x/S = 14.4$ are shown in Figure 3, which indicate that shortening the domain had little impact on predicted velocities.

The three-dimensional simulation employed a computational domain which duplicated the dimensions of Armaly *et. al.*'s experimental apparatus, under the assumption that the presence of side walls would induce three-dimensional effects for the Reynolds number range of concern. Thus, the computational domain was $20S$ long and $18.37S$ wide, and incorporated a vertical plane of symmetry along the duct centerline. A 100×70×35 mesh was used, giving the same longitudinal and vertical mesh spacing as the base two-dimensional cases. The mesh in the lateral (y) direction was nonuniform as well, with the grid spacing at the duct centerline twelve times that near the side wall. A Reynolds number of 800 was chosen for the simulation, since the disagreement of

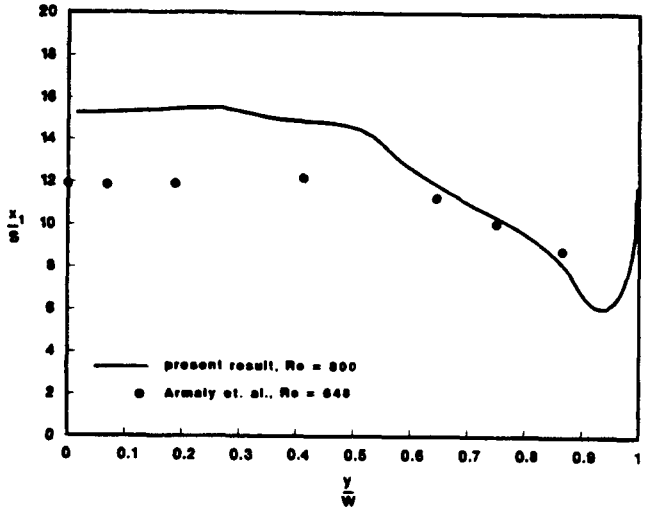

Figure 4: Spanwise variation of reattachment length.

two-dimensional predictions with data becomes significant at that value. The velocity profile used for the inlet boundary condition was computed from the series solution for fully developed laminar flow in a rectangular duct [10]. Due to computational cost, only one three-dimensional computation was attempted.

Figure 4 shows the spanwise variation of the reattachment and separation line for Re = 800 at a dimensionless time of 372, where the reference time is D_h/U_o, and the velocity field was initially quiescent. Also included, for the sake of qualitative comparison, is the data of Armaly *et. al.* for the reattachment line at Re = 648. The spanwise coordinate has been normalized by the duct half-width, W. Both the data and predicted position for the reattachment line show three-dimensional effects over a significant portion of the duct, although it is more apparent in the numerical result. The predicted position of the reattachment point at the duct centerline, $x_1/S = 15.3$, is in good agreement with Armaly *et. al.*'s data (see Figure 2), over-predicting it by 7%. However, at this time the computed flow field has not reached steady state, and the position of the reattachment line changes with time over a range of about 10%. Also at this time, a well-formed secondary recirculation bubble has not yet stabilized on the upper surface of the duct. It is not yet clear why the computed flow does not appear to be approaching steady state; the most likely explanation is that although the domain was of sufficient length for the open boundary to have little impact on the two-dimensional computed flow, it must be longer in the three-dimensional case. This area is still under investigation. However, the results of Figure 4 clearly confirm that three-dimensional effects contribute to the under-prediction of the reattachment point by two-dimensional simulations.

SUMMARY

A computational technique for the solution of the transient, non-isothermal, three-dimensional, incompressible Navier-Stokes equations has been developed. This method, based on an operator split of the governing equations, employs flux-corrected transport (FCT) for the advective operator, and implicit integration of the diffusive operators. The Poisson equations which result from implicit integration of the diffusive terms, as well as the imposition of the incompressibility constraint, were solved iteratively using the point-Jacobi preconditioned conjugate gradient technique. A treatment of boundary conditions for open boundaries, which does not require that the domain be extended far past the region of interest, was developed. Two- and three-dimensional simulations of flow over a backward-facing step show good agreement both with data and other numerical predictions. In addition, the three-dimensional result indicates that the deviation of two-dimensional results from data at higher Reynolds numbers can be attributed to three-dimensional effects caused by the presence of a side wall.

REFERENCES

1. KIM, J. and MOIN, P. - Application of a Fractional Step Method to Incompressible Navier-Stokes Equations, J. Comput. Phys., 59, 308–323, (1985).
2. BORIS, J. P., and BOOK, D. L. - Flux-Corrected Transport, I. SHASTA, A Fluid Transport Algorithm that Works, J. Comput. Phys., 11, 38–69, (1973).
3. BOOK, D. L., BORIS, J. P., and HAIN, K. - Flux-Corrected Transport II: Generalizations of the Method, J. Comput. Phys., 18, 248–283, (1975).
4. BORIS, J. P., and BOOK, D. L. - Flux-Corrected Transport III. Minimal Error Algorithms, J. Comput. Phys., 20, 397–431, (1976).
5. BAER, M. R. and GROSS, R. J. - A Two-Dimensional Flux-Corrected Solver for Convectively Dominated Flows, Sandia National Laboratories Report, SAND85-0613, 1985.
6. GRAY, D. D. and GIORGINI, A. - The Validity of the Boussinesq Approximation for Liquids and Gases, Int. J. Heat Mass Transfer, 19, 545–551, (1976).
7. GOLUB, G. H. and VAN LOAN, C. V. - Matrix Computations, John Hopkins University Press, Baltimore, 1983.
8. HAGEMAN, L. A. and YOUNG, D. M. - Applied Iterative Methods, Academic Press, New York, 1981.
9. ARMALY, B. F., DURST, F. J., PEREIRA, C. F., and SCHÖNUNG, B. - Experimental and Theoretical Investigation of Backward-facing Step Flow, J. Fluid Mech., 127, 473–496, (1983).
10. WHITE, F. M. - Viscous Fluid Flow, McGraw-Hill, New York, 1974, p. 123.