SINTEF A1686 - Unrestricted

REPORT

A SEGREGATED IMPLICIT PRESSURE PROJECTION METHOD FOR TURBULENT FLOWS

Torbjørn Utnes

SINTEF ICT
Applied Mathematics

June 2007

SINTEF REPORT

SINTEF

SINTEF ICT

Address: NO-7465 Trondheim,
NORWAY
Location: Alfred Getz vei 1, NTNU
NO-7491 Trondheim
Telephone: +47 73 59 30 48
Fax: +47 73 59 29 71

Enterprise No.: NO 948 007 029 MVA

TITLE

A SEGREGATED IMPLICIT PRESSURE PROJECTION METHOD FOR TURBULENT FLOWS

AUTHOR(S)

Torbjørn Utnes

CLIENT(S)

Avinor AS

REPORT NO. SINTEF A1686	CLASSIFICATION Unrestricted	CLIENTS REF. Erling Bergersen	
CLASS. THIS PAGE Unrestricted	ISBN 978-82-14-04061-6	PROJECT NO. 90A302	NO. OF PAGES/APPENDICES 29
ELECTRONIC FILE CODE		PROJECT MANAGER (NAME, SIGN.) Karl J. Eidsvik K.J.Eidsvik	CHECKED BY (NAME, SIGN.) Karstein Sørli Karstein Sørli
FILE CODE 90A302	DATE 2007-06-21	APPROVED BY (NAME, POSITION, SIGN.) Roger Bjørgan, Research Director for Roger Bjørgan	

ABSTRACT

In the present report an improved solution algorithm is described for the SIMRA program. The algorithm is a segregated, implicit projection method, which improves the stability property substantially. Although the computer demands increase per time step due to the new implicit formulation, the total gain is considerably because of a much better stability. An additional gain is the improved robustness, meaning that there are no needs for strong "safety" restrictions on the Courant number.

The algorithm is described and analysed formally for properties like consistency, stability and accuracy. In addition the method is demonstrated on some test cases, including two- and three-dimensional cases, neutral and stratified flows.

KEYWORDS	ENGLISH	NORWEGIAN
GROUP 1	Modelling	Modellering
GROUP 2	Numerics	Numerikk
SELECTED BY AUTHOR	Algorithm	Algoritme

Contents

1 Introduction

Efficient numerical algorithms for high-Reynolds number flows still represent a major challenge, although several successful methods have been developed. Some of the more efficient methods may be classified as segregated or de-coupled, iterative procedures. Well-known examples are the SIMPLE or SIMPLER-like class of algorithms, which are now in use for both finite difference, finite volume and finite element methods [1, 2, 3]. These algorithms can be used both for steady state and time-dependent problems. In the latter case projection methods and/or fractional step methods are also extensively used. Examples of such methods are the classical Chorin-Temam projection method [4, 5], and similar more advanced algorithms [6, 7, 8]. Another related method is the PISO algorithm, which may as well be classified as a predictor-corrector method, although pressure projection is an important part of the algorithm [9, 10]. In addition to the algorithmic concept, the choice of discretization method is also a discussed theme, typically arguments between FD, FVM, FEM or other approaches (see e.g. [11]).

Important goals for any numerical algorithm is to obtain a procedure that is stable, efficient and sufficient accurate. In practical use good stability is always regarded as important. On the other hand, a very stable method may give a feeling of reliability that may be an illusion if the method is inaccurate. Still, the opposite situation, i.e. an accurate explicit algorithm which may easily crash at difficult, advection dominated flows, will often be regarded as unreliable by the user. The computer/cost efficiency of the method is another important quality. Although parallelization techniques have improving the computational speed substantially, the core algorithm of the method is still very important. This is especially the case in forecast situations, where a short time window is given in advance for the computation to be completed [12].

These considerations seem to indicate that some sort of segregated, implicit (or semi-implicit) projection method may be a good choice. The present paper presents an algorithm of this kind, including analysis of consistency, accuracy and stability, and then shows some computational results which are compared with experimental data. The organization of the paper is as follows: Section 2 presents the actual governing equations to be solved, in our case the so-called anelastic formulation of the equations of motion. In order to solve the high-Reynolds number turbulent flow we use an unsteady Reynolds-averaging procedure together with a standard high-Reynolds (k, ϵ) turbulence model. These equations are then discretized (Section 3) using a mixed finite element formulation, and the segregated, implicit solution algorithm is presented in Section 4. Key properties of the algorithm are analysed in Section 5. Finally, in Section 6 the method is tested on some high-Reynolds number flows and compared with experimental data.

2 Governing equations

The equations of motion for incompressible flow may be generalized to atmospheric flows by use of the so-called anelastic approximation. This formulation is often applied in meteorological models, and may be written in the following energy conservative form ([13, 14, 15]):

$$\frac{D\mathbf{u}}{Dt} = -\nabla\left(\frac{p_d}{\rho_s}\right) + \mathbf{g}\frac{\theta_d}{\theta_s} + \frac{1}{\rho_s}\nabla\cdot\boldsymbol{\tau} + \mathbf{f}; \quad \nabla\cdot(\rho_s\mathbf{u}) = 0,$$
$$\frac{D\theta}{Dt} = \nabla\cdot(\gamma\nabla\theta) + q. \tag{1}$$

Here $(\mathbf{u}, p, \theta, \rho)$ represent velocity, pressure, potential temperature, and density, respectively. Furthermore, $\boldsymbol{\tau}$ is the stress tensor, $\mathbf{f}$ is a source term that may include rotational effects, $\mathbf{g}$ is the gravitational acceleration (with positive direction upwards), γ is the thermal diffusivity, and q is a temperature source term. Subscript 's' indicates hydrostatic values, and subscript 'd' the deviation between the actual value and its hydrostatic part, i.e.

$$p = p_s + p_d; \ \theta = \theta_s + \theta_d; \ \rho = \rho_s + \rho_d,$$

where the hydrostatic relation is given by

$$\partial p_s/\partial z = -g\rho_s.$$

In addition, the following expression for hydrostatic density may be derived from the state equation and the definition of potential temperature:

$$\rho_s = \frac{p_s}{R\theta_s}\left(\frac{p_o}{p_s}\right)^{R/C_p},$$

where R is the gas constant and C_p is the specific heat at constant pressure. Hence, once the hydrostatic (potential) temperature profile is given, the hydrostatic pressure and density may be calculated, and then substituted into equation system (1).

It may be noted that the Boussinesq approximation is obtained from the system (1) by assuming constant reference values (ρ_o, θ_o) instead of the hydrostatic values, and that formulation may as well be used for incompressible flow and ordinary temperature.

The aim of the present study is to solve these equations for high Reynolds-number flows. For this purpose we apply an unsteady Reynolds-averaged modelling of the equation system, together with a turbulence model. Presently a standard high-Reynolds (k, ϵ) turbulence model is used for this purpose. With these assumptions the model equations take the following form:

$$\frac{D\mathbf{u}}{Dt} = -\nabla\left(\frac{p_d}{\rho_s}\right) + \mathbf{g}\frac{\theta_d}{\theta_s} + \frac{1}{\rho_s}\nabla\cdot\boldsymbol{R} + \mathbf{f}; \quad \nabla\cdot(\rho_s\mathbf{u}) = 0,$$
$$\frac{D\theta}{Dt} = \nabla\cdot(\gamma_T\nabla\theta) + q,$$
$$\frac{Dk}{Dt} = \nabla\cdot(\nu_T\nabla k) + P_k + G_\theta - \epsilon, \tag{2}$$
$$\frac{D\epsilon}{Dt} = \nabla\cdot\left(\frac{\nu_T}{\sigma_e}\nabla\epsilon\right) + (C_1P_k + C_3G_\theta)\frac{\epsilon}{k} - C_2\frac{\epsilon^2}{k},$$
$$\nu_T = C_\mu\frac{k^2}{\epsilon}.$$

Here the variables are to be interpreted as Reynolds-averaged quantities. The stress tensor in the momentum equation has been replaced by the Reynolds stress tensor

$$R_{ij} = \nu_T \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} k \delta_{ij}, \tag{3}$$

and the potential temperature equation now contains an eddy diffusivity

$$\gamma_T = \nu_T / \sigma_T, \tag{4}$$

where σ_T is the turbulent Prandtl number. The production and stratification terms in the turbulence model are given by

$$P_k = \nu_T \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \frac{\partial u_i}{\partial x_j}, \quad G_\theta = -\frac{g}{\theta} \frac{\nu_T}{\sigma_T} \frac{\partial \theta}{\partial z}, \tag{5}$$

and conventional constants for the high-Reynolds (k, ϵ) model are [16]

$$(C_\mu, C_1, C_2, \sigma_e) = (0.09, 1.44, 1.92, 1.3).$$

The value for C_3 is more uncertain, and different practices have been used depending on the flow characteristics. In the present case we assume $C_3 G_\theta = \max(G_\theta, 0)$, i.e. $C_3 = 0$ in stably stratified flows, else $C_3 = 1$ (cf. Rodi [17]).

3 Algebraic formulation

The governing equations (2) are discretized in space by use of a finite element method, and the time integration is performed using a semi-implicit two-niveau formulation. In compressed form the discretized equation system may be written in the following form:

$$\begin{aligned} \left[\frac{M}{\Delta t} + \alpha A_u^* \right] \Delta u^{n+1} &= -A_u^* u^n + s_u^n - C \left\{ \frac{p}{\rho_s} \right\}^{n+1}, & (6) \\ C^T \rho_s u^{n+1} &= 0, & (7) \\ \left[\frac{M}{\Delta t} + \alpha A_\phi^* \right] \Delta \phi^{n+1} &= -A_\phi^* \phi^n + s_\phi^n. & (8) \end{aligned}$$

Here M represents the mass matrix, A is the sum of diffusion and advection matrices (subscripts indicating the actual variable), C is the gradient matrix, and s (with subscripts) represents source terms. The implicit parameter α may be chosen in the interval (1/2, 1), and A^* indicates the advection velocity taken at $u^{n+\alpha}$. The variables (u, p) are redefined here as nodal vectors for velocity and pressure, and ϕ represents nodal vectors for each of the scalar variables (θ, k, ϵ). The increments are defined by $\Delta u^{n+1} = u^{n+1} - u^n$, $\Delta \phi^{n+1} = \phi^{n+1} - \phi^n$, where superscripts indicate time levels.

4 Segregated implicit projection algorithm

In practical applications a numerical method should have good numerical stability and cost/computer efficiency, in addition to an acceptable accuracy. There is no easy way to obtain an 'optimal' method, and compromises usually have to be made. However, some types of algorithms have proved to be fairly efficient, such as implicit or semi-implicit, segregated formulations. Examples of such algorithms are the well-known class of SIMPLE/SIMPLER-like iterative methods [3, 2], non-iterative pressure-correction procedures like the PISO method [9, 10], and other segregated projection formulations like the 'Projection 1' and 'Projection 2' formulations [6, 18].

In this study we propose a segregated, implicit projection (SIP) method that is non-iterative, with corrections within each time-step. This algorithm has several features in common with the SIMPLER-like pressure projection method described in [3], but instead of iterations it applies corrections similar to the PISO method.

4.1 SIP algorithm

From the considerations above the proposed algorithm is given by the follows steps:

Predict the pressure field via a pseudo-velocity prediction from the system

$$\begin{aligned} \tilde{M}\Delta\hat{u} &= -A_u u^n + s_u; \text{ where } \Delta\hat{u} = \hat{u} - u^n, && (9)\\ Lp^* &= C^T \rho_s \hat{u}, && (10) \end{aligned}$$

where L represents a discretized Lpalacian operator, see Remarks below.

Compute the velocity field from the (semi)implicit momentum equation

$$\left[\tilde{M} + \alpha A_u^*\right]\Delta u^* = -A_u^* u^n + s_u - C\left\{\frac{p}{\rho_s}\right\}^*; \text{ where } \Delta u^* = u^* - u^n. \qquad (11)$$

Compute (semi)implicit equations for other scalar quantities (k, ϵ, θ):

$$\left[\tilde{M} + \alpha A_\phi^*\right]\Delta\phi^{n+1} = -A_\phi^*\phi^n + s_\phi; \text{ where } \Delta\phi^{n+1} = \phi^{n+1} - \phi^n. \qquad (12)$$

Correct the velocity and pressure fields by use of the projection step

$$\begin{aligned} L\Delta p^{n+1} &= C^T \rho_s u^*; \text{ where } p^{n+1} = p^* + \Delta p^{n+1}, && (13)\\ \tilde{M}\left(\rho_s u^{n+1} - \rho_s u^*\right) &= -C\Delta p^{n+1}. && (14) \end{aligned}$$

The advection matrices A^* indicate use of a time-centered advection velocity, which may be calculated as $u^{n+\alpha} = (1+\alpha)u^n - \alpha u^{n-1}$, but see Remarks below. Further, $\tilde{M} = M/\Delta t$, and the mass matrix may be lumped (see Remarks). The present algorithm is comparable to that of Nonino and Comino [19], and it resembles several other projection methods as well. Apart from the pressure prediction, the method is similar to other incremental projection methods. However, like pressure-free projection methods, the pressure is predicted anew at each time-step via the prediction step, and therefore has no memory that could accumulate errors. This may be concidered appealing provided the accuracy is comparable to standard second-order projection methods.

Remarks:

1. Regarding the implicit parameter (α) two cases are of interest: The fully implicit case, $\alpha = 1$, especially suited for steady state computations, and the time-centered case, $\alpha = 1/2$, for time-accurate computations. In the first case the advection matrix may be taken as A^n to simplify calculations.

2. The advection velocity $u^{n+\alpha}$ in the algorithm above is calculated via extrapolation. Alternatively we may use the correction step in a complete projection 1: $u^{n+\alpha} = u^n + \alpha\tilde{M}^{-1}CP_*$.

3. This algorithm may be applied to both *algebraic* and *continuous* formulations. In the algebraic case the space discretization is performed first, with boundary conditions included. In that case the Laplace matrix is calculated as $L = C^T M_L^{-1} C$, where M_L is the lumped mass matrix (for computational reasons). It may be noted that boundary conditions are included in this matrix, hence explicit pressure boundary conditions are not needed in the algebraic formulation. The continuous formulation is derived from the continuous form of the equations, and the Laplacian is then calculated from a standard element assembling procedure. The continuous formulation therefore needs explicitly stated pressure boundary conditions.

4. In the continuous formulation a consistent mass matrix can be used throughout, while in the algebraic formulation the mass lumping in the pressure equation implies the use of M_L also in the final projection step. In our implementation of the algebraic formulation we use M_L throughout, although it is possible to use consistent mass in the momentum equation (but this may require a factor MM_L^{-1} in the pressure term, see Gresho and Chan [6]).

4.2 Implementation

The implementation used is based on Q_1Q_0 mixed elements, and is an algebraic formulation (necessarily, due to the Q_0 pressure interpolation). As is well known, the Q_1Q_0 element does not satisfy the inf-sup stability condition. However, a consistent stabilization can be included to avoid possible problems, cf. Christon [20]. This is especially advisable when solving the pressure equation iteratively, which is the present case due to three-dimensional applications.

5 Analysis

In the following section an analysis is given of the consistency, time accuracy, and stability of the proposed algorithm. Steady state convergence is also investigated and exemplified. For simplicity only the the incompressible form of the Navier-Stokes equations are regarded, and Backward Euler time integration is assumed in the momentum equation. Also, the mass matrix is assumed to be lumped, although it is possible to modify this, cf. Remark 4 above.

5.1 Consistency analysis

It is a fundamental requirement that the proposed algorithm represents an approximation to the actual differential algebraic equations, i.e. the semi-discretized Navier-Stokes equations. For convenience these equations are re-written here in the form

$$M\dot{u} + A(u)u + CP = f; \;\; C^T u = g, \tag{15}$$

where the the vectors f and g now explicitly include boundary conditions (f also includes source terms). Boundary conditions imply additional time discretization errors, and this explicit way of representing them is therefore advantageously for this kind of analysis. This notation is similar to that employed by Gresho and Chan [6].

Before analysing the solution algorithm, we may write the solution of (15) in a 'continuous projection' form as follows: From the momentum equation (15a) the pressure may be expressed as

$$LP = C^T M^{-1}[f - A(u)u] - C^T \dot{u},$$

where $L = C^T M^{-1} C$, and by using the continuity constraint from (15b), we have

$$P = L^{-1}[C^T M^{-1}(f - A(u)u) - \dot{g}].$$

When this expression is substituted into the momentum equation, the following formulation is obtained:

$$\dot{u} = QM^{-1}[f - A(u)u] + M^{-1}CL^{-1}\dot{g}, \tag{16}$$

where we have introduced a projection matrix defined as

$$Q := I - M^{-1}CL^{-1}C^T. \tag{17}$$

In order to assure consistency with (15), we therefore need to show that the SIP algorithm is a consistent approximation to (16), as $\Delta t \to 0$.

The SIP algorithm (with $\alpha = 1$) applied to (15) is given by:

Prediction:

$$Mu_* = Mu_n - \Delta t(A_n u_n + f_n); \;\; LP_* = C^T u_* - g_{n+1}. \tag{18}$$

Momentum:

$$(M + \Delta t A_n)\tilde{u}_{n+1} = Mu_n + \Delta t(f_{n+1} - CP_*). \tag{19}$$

Projection:

$$Mu_{n+1} = M\tilde{u}_{n+1} - \Delta t C\delta P; \;\; C^T u_{n+1} = g_{n+1}. \tag{20}$$

The projection step (20) is performed by first solving

$$L\,\delta P = C^T\tilde{u}_{n+1} - g_{n+1},$$

and then updating

$$u_{n+1} = \tilde{u}_{n+1} - M^{-1}C\,\delta P.$$

By substituting for δP, this may be expressed as

$$\begin{aligned} u_{n+1} &= \tilde{u}_{n+1} - M^{-1}CL^{-1}C^T\tilde{u}_{n+1} + M^{-1}CL^{-1}g_{n+1} \\ &= (I - M^{-1}CL^{-1}C^T)\tilde{u}_{n+1} + M^{-1}CL^{-1}g_{n+1} \\ &= Q\tilde{u}_{n+1} + M^{-1}CL^{-1}g_{n+1}, \end{aligned} \tag{21}$$

where we have applied the projection operator (17).

From the momentum equation (19) the velocity field can be expressed as

$$\tilde{u}_{n+1} = (I + \Delta t M^{-1}A_n)^{-1}[u_n + \Delta t M^{-1}(f_{n+1} - CP_*)].$$

For small values of Δt this may be approximated as

$$\tilde{u}_{n+1} = u_n + \Delta t\tilde{a}_n - \Delta t^2 M^{-1}A_n\tilde{a}_n + O(\Delta t^3),$$

where we have in addition introduced the acceleration

$$\tilde{a}_n := M^{-1}(f_{n+1} - A_n u_n - CP_*).$$

By substitution of the above expression for $\tilde{u}_{n+1}$ back into the projection expression (21), the following is obtained:

$$\begin{aligned} u_{n+1} &= Q\tilde{u}_{n+1} + M^{-1}CL^{-1}g_{n+1} \\ &= u_n + \Delta t Q\tilde{a}_n + M^{-1}CL^{-1}(g_{n+1} - g_n) + O(\Delta t^2), \end{aligned}$$

where we have made use of the relation

$$Qu_n = u_n - M^{-1}CL^{-1}g_n.$$

It can be shown that

$$QM^{-1}C = 0,$$

and hence the above expression may be written

$$u_{n+1} = u_n + \Delta t QM^{-1}(f_{n+1} - A_n u_n) + M^{-1}CL^{-1}(g_{n+1} - g_n) + O(\Delta t^2).$$

By letting $\Delta t \to 0$, it is seen that this expression is consistent with (16), and the algorithm is therefore consistent with the desired system (15).

5.2 Time accuracy

It is well known that a Backward Euler (BE) time discretization gives a global error of order $O(\Delta t)$, while e.g. a Crank-Nicholsen (CN) scheme gives $O(\Delta t^2)$. This also holds for direct solution of the coupled Navier-Stokes equations, and applies to both velocity and pressure (see Gresho and Sani [18]). Similar results are obtained using incremental projection methods, while most non-incremental projection methods are restricted to $O(\Delta t)$ due to a splitting error (cf. [7, 22, 23]).

The time-centered SIP formulation ($\alpha = 1/2$) yields the following for the momentum equation (neglecting source terms):

$$\left(M + \frac{\Delta t}{2} A_{n+1/2}\right) \tilde{u}_{n+1} = \left(M - \frac{\Delta t}{2} A_{n+1/2}\right) u_n - \Delta t C P_*,$$
$$M u_{n+1} = M \tilde{u}_{n+1} - \Delta t C \delta P,$$

or, by substituting for $\tilde{u}_{n+1}$ from the latter expression:

$$\left(M + \frac{\Delta t}{2} A_{n+1/2}\right) u_{n+1} = \left(M - \frac{\Delta t}{2} A_{n+1/2}\right) u_n - \Delta t C (P_* + \delta P)$$
$$- \frac{\Delta t^2}{2} A_{n+1/2} M^{-1} C \delta P$$

This is to be compared with the corresponding second-order CN formulation

$$\left(M + \frac{\Delta t}{2} A_{n+1/2}\right) u_{n+1} = \left(M - \frac{\Delta t}{2} A_{n+1/2}\right) u_n - \Delta t C P_{n+1},$$

implying an additional term in the SIP formulation given by

$$\frac{\Delta t^2}{2} A_{n+1/2} M^{-1} C \delta P.$$

It is noted that this term is $O(\Delta t^3)$, i.e. the same order of magnitude as the truncation error in the CN formulation. Hence the present time-centered SIP formulation should give second-order accuracy.

In order to test the actual numerical performance, the standard driven cavity flow is used as an example. At time t = 0 the tangential velocity is set to u = 1 along the upper wall, all other velocity components are zero along the walls. A Reynolds number of Re = 400 is used, and the time evolution is studied for an initial period of 2 s, using a mesh of 40 × 40 uniform elements. The computed result is illustrated in Fig.1, showing stream lines of the velocity field after a time of 2 s. The L_2 error estimates are shown in Fig.2 for velocity and pressure. As can be seen, second-order accuracy is indeed obtained for the velocity, while the pressure is first-order accurate. This is in accordance with other standard second-order projection methods (e.g. [23] with references therein). Possible additional corrections/modifications may improve the pressure accuracy somewhat in these methods, but generally less than an order of magnitude ([24], [25]).

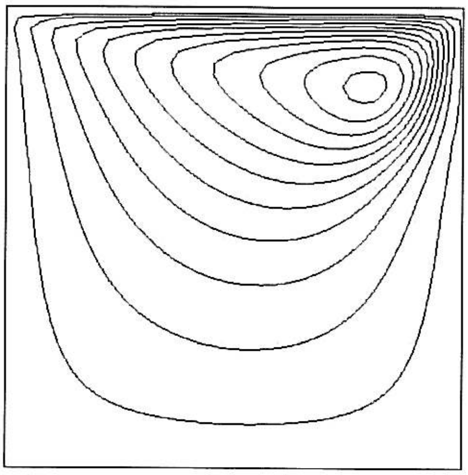


Figure 1: Computed streamlines for driven cavity flow at time t = 2 s, Re = 400.

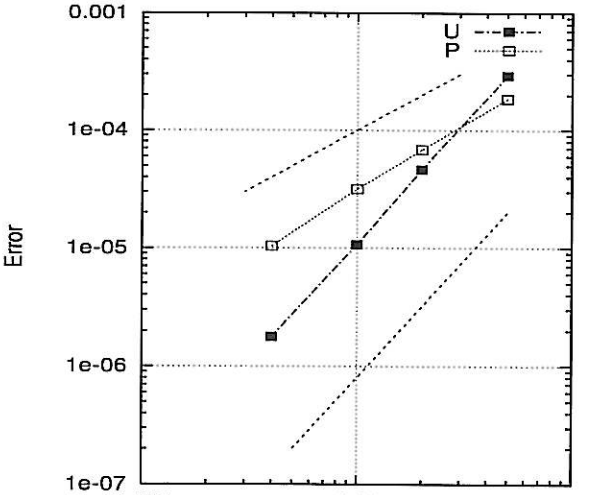


Figure 2: L_2 error estimates for velocity and pressure, time-centered SIP formulation. Steepness lines for Δt and Δt^2 are added as stipled lines.

5.3 Steady state

Although the present formulation is a time stepping method, it is still important that the solution converges to a correct steady state solution independent of the chosen time step. In real simulations time variations may change in a large range. An example may be meteorological predictions, where quasi-steady flows may occur during larger parts of a simulation. By analysing the present formulation as given by eqs. (18 - 20), it may be shown that the steady state solution is indeed independent of the time step, and is consistent with the steady state formulation.

From equation (20a) we have

$$u_{n+1} = \tilde{u}_{n+1} - \Delta t M^{-1} C \delta P,$$

or equivalently

$$\begin{aligned}(M + \Delta t A_n) u_{n+1} &= (M + \Delta t A_n)\tilde{u}_{n+1} - \Delta t (M + \Delta t A_n) M^{-1} C\, \delta P \\ &= M u_n + \Delta t f_{n+1} - \Delta t C P_* - \Delta t (M + \Delta t A_n) M^{-1} C\, \delta P,\end{aligned}$$

where (19) has been substituted. Division by Δt, and assuming steady state $(u_{n+1} - u_n)/\Delta t \to 0$, this implies

$$A_n u_{n+1} = f_{n+1} - C P_{n+1} - \Delta t A_n M^{-1} C \delta P.$$

But steady state also implies $\delta P \to 0$. This can be seen by comparing the prediction and projection steps (18, 20). When $\dot{u} = 0$, time indices are irrelevant, and it can be seen that the correction step using P_* satisfies the continuity condition. It follows that $\delta P = 0$, and the final result becomes

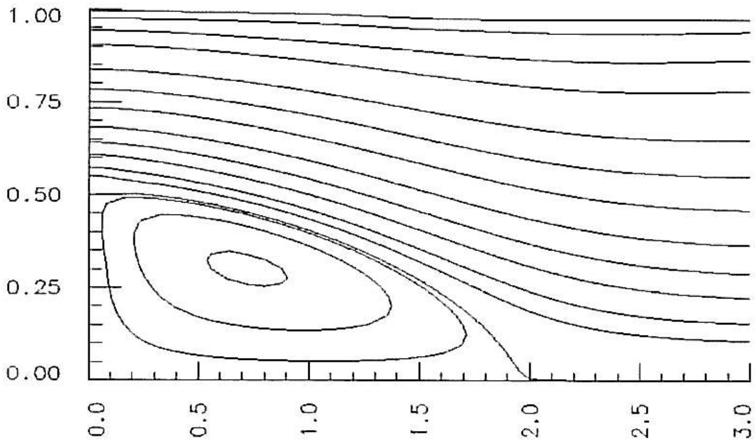
$$Au = f - CP,$$

which is the correct form of the steady state momentum equation.

In order to test this result, the following simple simulation has been performed: Flow over a step in a channel with fixed upper and lower walls. The step height is 0.5 m, the total channel height is 1 m, and the length is set to 8 m. The Reynolds number is $R_h = 100$ based on the step height, and the inflow profile is specified as parabolic:

$$u(y) = 16(y - 0.5)(1 - y), \text{ for } 0.5 \leq y \leq 1.0,$$

where the vertical coordinate (y) is measured from the lower wall, i.e. $y = 0.5$ at the step. For the chosen Reynolds number a mesh of 50 × 100 nodes is fine enough to obtain a fairly accurate space resolution. Fig. 3 shows results from two of these simulations, both terminated after 100 s, where the solutions were regarded as stationary (due to residual evaluations). The shown results are obtained with very different time increments, corresponding to maximum Courant numbers of $Cr = 0.12$ and $Cr = 24.84$, respectively. The interesting point is that virtually no difference is observes between these results, and we may therefore conclude that the method gives a steady state solution that is indeed independent of the time-step applied. Actually, a number of simulations performed on other problems confirm this conclusion.



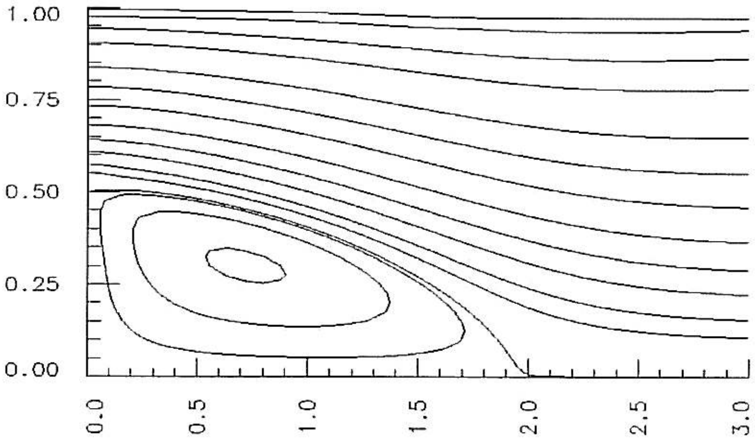


Figure 3: Steady state solutions represented as stream function. The upper figure is for a time-step corresponding to a maximum Courant number of $Cr = 0.12$, the lower figure with $Cr = 24.84$.

5.4 Stability analysis

As illustrated in the previous section, and also experienced in more complicated examples, the numerical stability of the present algorithm is quite good in practice. Hence, the time increment may be chosen according to accuracy requirements rather than stability limitations. In the present section we aim to perform a closer analysis of the formal stability. For this purpose we re-write the governing equations in semi-discretized form (15) as

$$M\dot{u} + Ku + N_s(u)u + CP = f, \quad C^T u = 0, \tag{22}$$

where advection and diffusion terms are written out separately, and the skew-symmetric form of the advection matrix is applied. For simplicity we assume homogeneous Dirichlet boundary conditions,

$$u_{n+1} = \tilde{u}_{n+1} = 0 \text{ on } \Gamma,$$

as reflected in the formulation above, and assume $f = 0$. In addition appropriate initial conditions are assumed. The SIP algorithm applied to this formulation is as follows:

Prediction:

$$M(u_* - u_n) = -\Delta t(K + N_n)u_n; \quad LP_* = C^T u_*. \tag{23}$$

Momentum:

$$M(\tilde{u}_{n+1} - u_n) + \Delta t(K + N_n)\tilde{u}_{n+1} = -\Delta t CP_*. \tag{24}$$

Projection:

$$M(u_{n+1} - \tilde{u}_{n+1}) + \Delta t C\,\delta P_{n+1} = 0; \quad C^T u_{n+1} = 0. \tag{25}$$

Given a pressure prediction (P_*) from (23), it can be shown that the algorithm (24 - 25) is absolutely stable.

To analyse the stability, we start by multiplying (24) by $2\tilde{u}_{n+1}^T$, which after some manipulation yields:

$$\tilde{u}_{n+1}^T M\tilde{u}_{n+1} + (u_{n+1} - \tilde{u}_{n+1})^T M(u_{n+1} - \tilde{u}_{n+1}) + 2\Delta t\tilde{u}_{n+1}^T K\tilde{u}_{n+1} = \\ u_n^T M u_n - 2\Delta t\tilde{u}_{n+1}^T CP_*.$$

Here the condition $n \cdot \tilde{u}_{n+1}|_\Gamma = 0$ has been imposed to eliminate advection terms, provided the skew-symmetric form is used.

Secondly, (25a) is multiplied by $2\Delta t(CP_*)^T M^{-1}$, resulting in the expression

$$-2\Delta t\tilde{u}_{n+1}^T(CP_*) + 2\Delta t^2 P_*^T L\,\delta P = 0,$$

where the continuity condition $C^T u_{n+1} = 0$ has also been imposed.

By adding together these results, we have:

$$\tilde{u}_{n+1}^T M\tilde{u}_{n+1} + (u_{n+1} - \tilde{u}_{n+1})^T M(u_{n+1} - \tilde{u}_{n+1}) + 2\Delta t\tilde{u}_{n+1}^T K\tilde{u}_{n+1} \\ +2\Delta t^2 P_*^T L\,\delta P = u_n^T M u_n.$$

Finally, (25a) is multiplied by u_{n+1}^T, and we obtain

$$u_{n+1}^T M u_{n+1} - u_{n+1}^T M \tilde{u}_{n+1} = 0,$$

where the pressure term has been eliminated due to the continuity constraint and the homogeneous condition $n \cdot u_{n+1} = 0$ on Γ. By adding this to the previous result, we obtain the following estimate:

$$u_{n+1}^T M u_{n+1} + 2\Delta t \tilde{u}_{n+1}^T K \tilde{u}_{n+1} + 2\Delta t^2 P_*^T L P_{n+1} \leq u_n^T M u_n + 2\Delta t^2 P_*^T L P_*. \tag{26}$$

Here the predicted pressure P_* is given from (23) as

$$P_* = -\Delta t L^{-1} C^T M^{-1} (K + N_n) u_n,$$

and hence the last term on the right hand side of (26) is of the form

$$2\Delta t^2 P_*^T L P_* \propto \Delta t^4 f(u_n),$$

where $f(u_n)$ is a function of the previous velocity u_n and time-independent matrices. The estimate (26) is therefore restricted by the previous velocity, and the stability is secured provided appropriate initial values.

6 Computational results

The SIP algorithm has been implemented in a numerical code, presently using a mixed Q_1Q_0 formulation for the velocity-pressure interpolation. In the following we show some examples of results compared with data; all of these are for high-Reynolds number turbulent flow, using the standard (k, ϵ) turbulence model.

For all of these test examples the implicit parameter is chosen to be $\alpha = 1$, i.e. fully implicit. The computations were run with a Courant number in the range 2 - 3.

6.1 Boundary conditions

All of the present examples use Dirichlet inflow conditions and free outflow conditions in the finite element sense, i.e. a weak form of zero traction for the momentum equations and zero normal derivative for other scalar quantities.

For boundary layer flows the inflow profiles are specified using a logarithmic velocity profile and corresponding profiles for turbulent kinetic energy and dissipation:

$$\begin{gathered} u = \frac{u_*}{\kappa} \ln\left(\frac{z}{z_o}\right); \quad k = max\left\{C_\mu^{-1/2} u_*^2 (1 - z/\delta)^2, k_{min}\right\}, \\ \epsilon = C_\mu^{3/4} k^{3/2} / \ell; \text{ where } \ell = \frac{\kappa z}{1 + 4z/\delta}. \end{gathered} \tag{27}$$

Here δ is the boundary layer thickness, k_{min} is a minimum turbulence, z_o is a roughness parameter, $\kappa \approx 0.4$, and other quantities are as defined before.

Walls or terrain boundaries are assumed to be rough, and standard wall conditions are used:

$$\frac{u_t}{u_*} = \frac{1}{\kappa} \ln \left(\frac{d}{z_o} \right); \quad u_n = 0, \tag{28}$$

where (u_t, u_n) are the tangential and normal velocity components at a small distance d from the wall. The implementation is performed by applying (28) at the second node from the wall. In addition it is secured that the velocity gradient at the second element from the wall satisfies

$$\frac{\partial u_t}{\partial d} = \frac{u_*}{\kappa d}, \tag{29}$$

which follows directly from (28). Another way to implement such conditions is by using special linear-logarithmic elements [26].

Near-wall conditions for the turbulence variables are given by

$$k = \frac{u_*^2}{C_\mu{}^{1/2}}; \quad \epsilon = \frac{u_*^3}{\kappa d}. \tag{30}$$

For stratified problems, the temperature wall condition used in the present case is a free condition in the finite element sense, that is a weak form of zero normal derivative.

6.2 Turbulent flow over a two-dimensional hill

This example is chosen to illustrates the ability of the model to calculate neutral turbulent flow with separation and recirculation. The geometry of the hill is defined by

$$h(x) = h_o cos^2\theta; \quad \theta = x/h_o \tan 50^o$$

, where h_o is the maximum height of the hill. Experimental setup and results are given in [27, 28], and the flow is characterized by a boundary layer height of $\delta/h_o = 5$, a roughness parameter $z_o/h_o = 3 \times 10^{-4}$, and a Reynolds number $Re = U_o h_o/\nu = 5.6 \times 10^4$.

The computational domain is chosen as $-8 \leq x/h_o \leq 25$, $0 \leq z/h_o \leq 10$, where the center of the hill is at the origo of the coordinate system. The computations are performed with a fairly fine mesh of (210 x 40) elements, with clustering close to the hill and the bottom boundary. Close to the bottom boundary the minimum mesh spacing is only $\Delta z_{min}/z_o \approx 20$.

Results from this calculation are shown in figures 2 and 3. Figure 2 illustrates the overall results of flow field (stream lines) and turbulence $(\sqrt{k})$. It is well known that the standard k, ϵ model tends to under-predict re-attachment lengths. The present calculation gives a recirculation length of $x_L/h_o \approx 6.5$, while the experiments show $x_L/h_o \in (6.6, 6.8)$. This is an acceptable result, although the tendency is somewhat under-prediction. The maximum turbulence behind the hill is about $\sqrt{k}/U_o \approx 0.25$, and this is also fairly close to the experimental results.

Figures 3 and 4 show comparisons with experimental data for both velocity and turbulence. Figure 3 illustrates comparisons of horizontal velocity profiles at the three sections

$x/h_o = 0, 5, 8$. The agreement is fairly good, although one may observe a tendency of too slow re-adjustment downstream the recirculation zone. This is a well-known effect of the turbulence model.

Comparisons of turbulence profiles for the same sections are given in figure 4. These profiles represent the variance $\sigma_1/U_o = \sqrt{\overline{u'u'}}/U_o$, and the corresponding numerical results are estimated as $\sigma_1/U_o \approx \sqrt{1.2k}/U_o$ (cf. Paterson and Holmes [29]). The numerical results correspond relatively well with the experimental data, especially in the boundary layer domain, while there is somewhat more discrepancy in the outer domain.

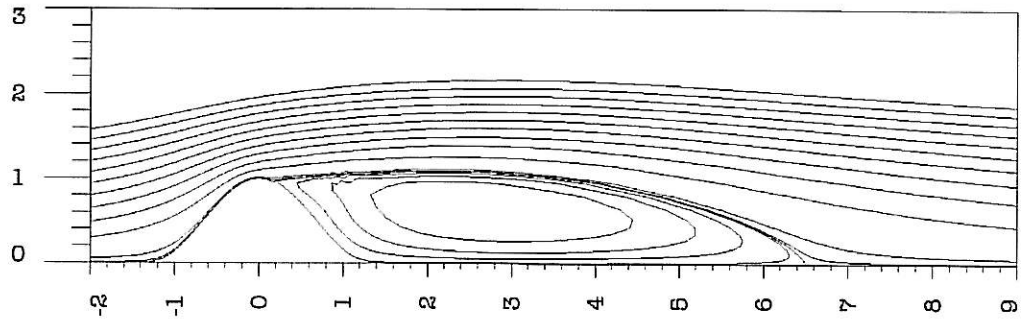


Figure 4: Turbulent flow over a two-dimensional hill. The figure illustrates streamlines, indicating a recirculation length of $x_L/h_o \approx 6.5$.

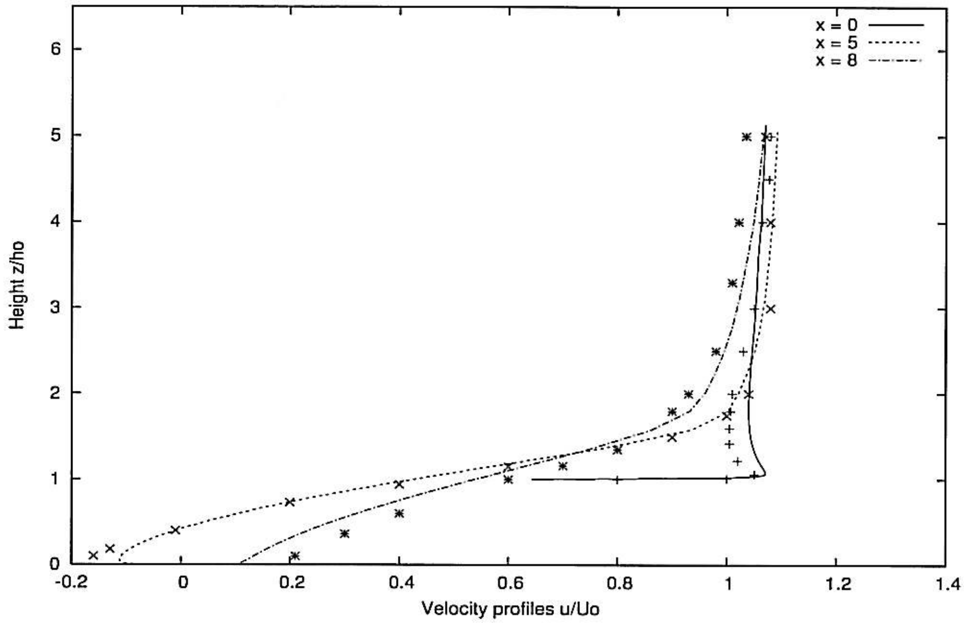


Figure 5: Comparison of computed velocity profiles (lines) with experimental data (dots) at sections $x/h_o = 0$, 5, and 8.

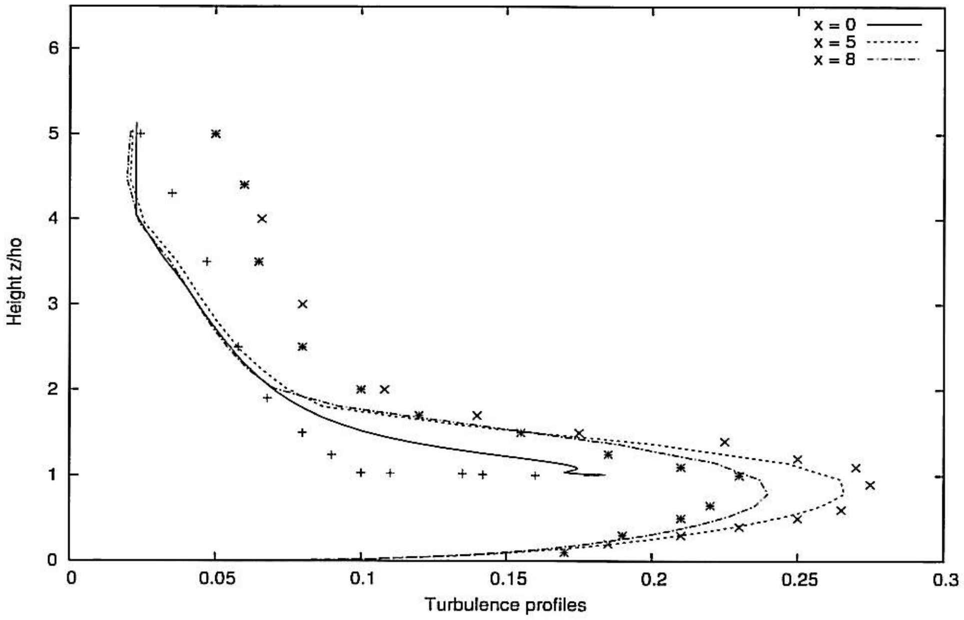


Figure 6: Comparison of computed velocity profiles (lines) with experimental data (dots) at sections $x/h_o = 0$, 5, and 8.

6.3 Flow around a circular cylinder at high Re number

Flow around a cylinder at high Reynolds number represents unsteady flow, and therefore requires a real time simulation. The present case is performed both to test the ability of the algorithm for this kind of flow, and also to investigate the ability of the simplified, approximate projection discussed in Section 4.2.

Simulations are performed for a Reynolds number $Re = 3 \times 10^6$. The domain is chosen to $-10 \leq x/D \leq 20$, $-10 \leq y/D \leq 10$, and the presented results are obtained on a mesh with 12000 elements, using a fine grading towards the cylinder surface with a minimum mesh size of $\Delta/D = 5 \times 10^{-4}$.

The inflow boundary conditions are specified as a constant velocity $U_\infty = 1$ m/s, a minimum turbulence $(k_{min}, \epsilon_{min})$, and with all other boundaries treated as open/free boundaries.

For this kind of flow the force coefficients on the cylinder are of interest:

$$C_D = \frac{F_1}{1/2\rho D U_\infty^2}; \quad C_L = \frac{F_2}{1/2\rho D U_\infty^2},$$

where (F_1, F_2) are the longitudinal and transversal force components, respectively, given by:

$$F_1 = \int (p n_1 + \tau_w n_2) ds; \quad F_2 = \int (p n_2 + \tau_w n_1) ds.$$

Here (p, τ_w) are the dynamic pressure and wall shear stress, respectively, (n_1, n_2) are the normal components in the two directions, and the integration is around the cylinder.

Table 1: Flow around a circular cylinder at $Re = 2 \times 10^{-6}$

Case	Drag coefficient C_D	Lift coefficient C_L
Experimental	0.55 ± 0.15 [31, 32]	± 0.08 to ± 0.16 [33]
Computation	0.56	± 0.17

Table 1 summarizes the results regarding force coefficients. As can be seen, the present results compare well with experimental data, although there is a large spread in the data for this Reynolds number.

Figure 7 shows the computed force coefficients as a function of time (upper), and the time-averaged pressure coefficient around the cylinder (lower). The latter is compared with experimental results for a similar Reynolds number, and is defined by

$$C_P = \frac{p - p_\infty}{1/2\rho U_\infty^2}.$$

It is seen that the computed mean drag and lift coefficients (C_D, C_L) lie in the same range as the experimental data. Furthermore, it is found that the computed separation

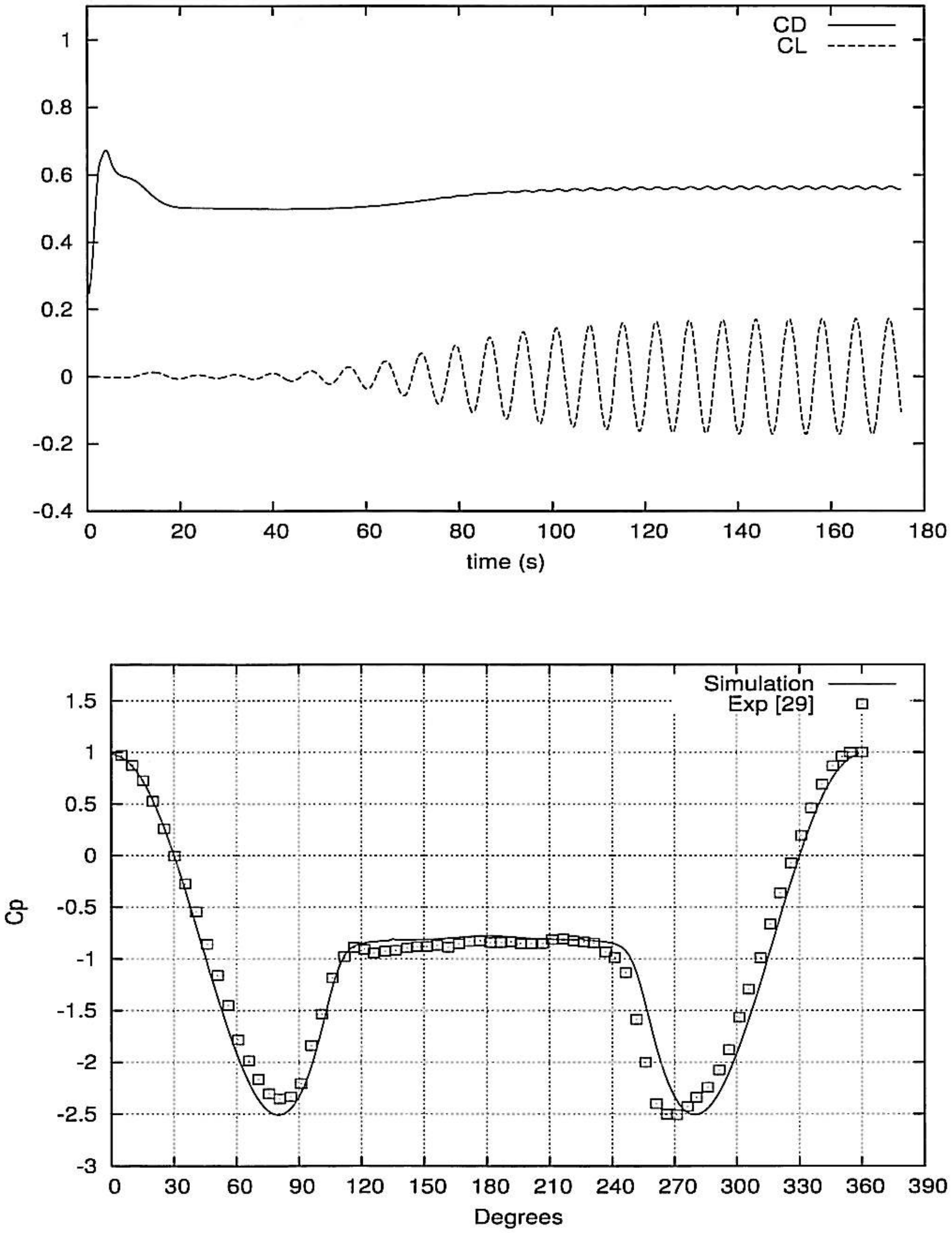


Figure 7: $Re = 3 \times 10^6$, smooth cylinder. Upper: computed force coefficients. Lower: computed mean pressure coefficient compared with experimental data [34].

point has a mean position close to 115 degrees (measured from the stagnation point), which compares well with experimental data [34].

The mean pressure coefficient is compared with experimental data [34] in the lower part of figure 7. Although there is a certain discrepancy, the computed pressure distribution is fairly close to the data.

6.4 Turbulent flow around a three-dimensional hill

Hunt and Snyder [30] documented a series of laboratory experiments on flows over a three-dimensional model hill. The following examples are based on two of these experiments, using a hill shape

$$h(x,y) = H\left[\frac{1.04}{1+r^4} - \frac{0.083}{1+(r-r_1)^2/a_1^2} - 0.03\right],$$

where $r = \sqrt{(x^2+y^2)}/H$, $r_1 = 20.3/H$, $a_1 = 7.6/H$, and the hill height is given by $H = 22.9$ cm.

The inflow profiles for velocity, turbulent kinetic energy, turbulent length scale and dissipation are specified as given in (27). For the inflow conditions the boundary layer height is $\delta = 0.3H$. The Reynolds number is $Re = 10^4$ based on the hill height, and the roughness parameter is $z_o/H = 0.0003$. The computational domain is given by $-6 \leq x/H \leq 8$, $-5 \leq y/H \leq 5$, $0 \leq z/H \leq 7$, and the present computations are performed on a structured mesh consisting of (100 x 100 x 50) tri-linear elements, with clustering around the hill and towards the boundary layer.

6.4.1 Neutral Flow

Results from this simulation are shown in figure 8, which illustrates velocity field at a near-ground section, and velocity and turbulence intensity along the vertical symmetry plane. The flow is characterized by two symmetric eddies downstream of the hill, as illustrated in figure 8a; and along the symmetry plane these eddies combine to a back-flow circulation (figure 8b). A simple way to characterize these eddies is to specify their center at the downstream hill surface, and to identify the recirculation point downstream along the symmetry plane. A comparison between experimental results and computations is given in Table 2 for these quantities, and indicates a reasonable agreement. A main reasons for deviations in such flows may often be attributed to inaccuracies in the turbulence model.

Table 2: Neutral flow around a three-dimensional hill

Case	Vortex center $(x,y)/H$	Recirculation x_R/H
Experimental	1.3 ± 0.7	3.5
Computation	1.2 ± 0.7	3.2

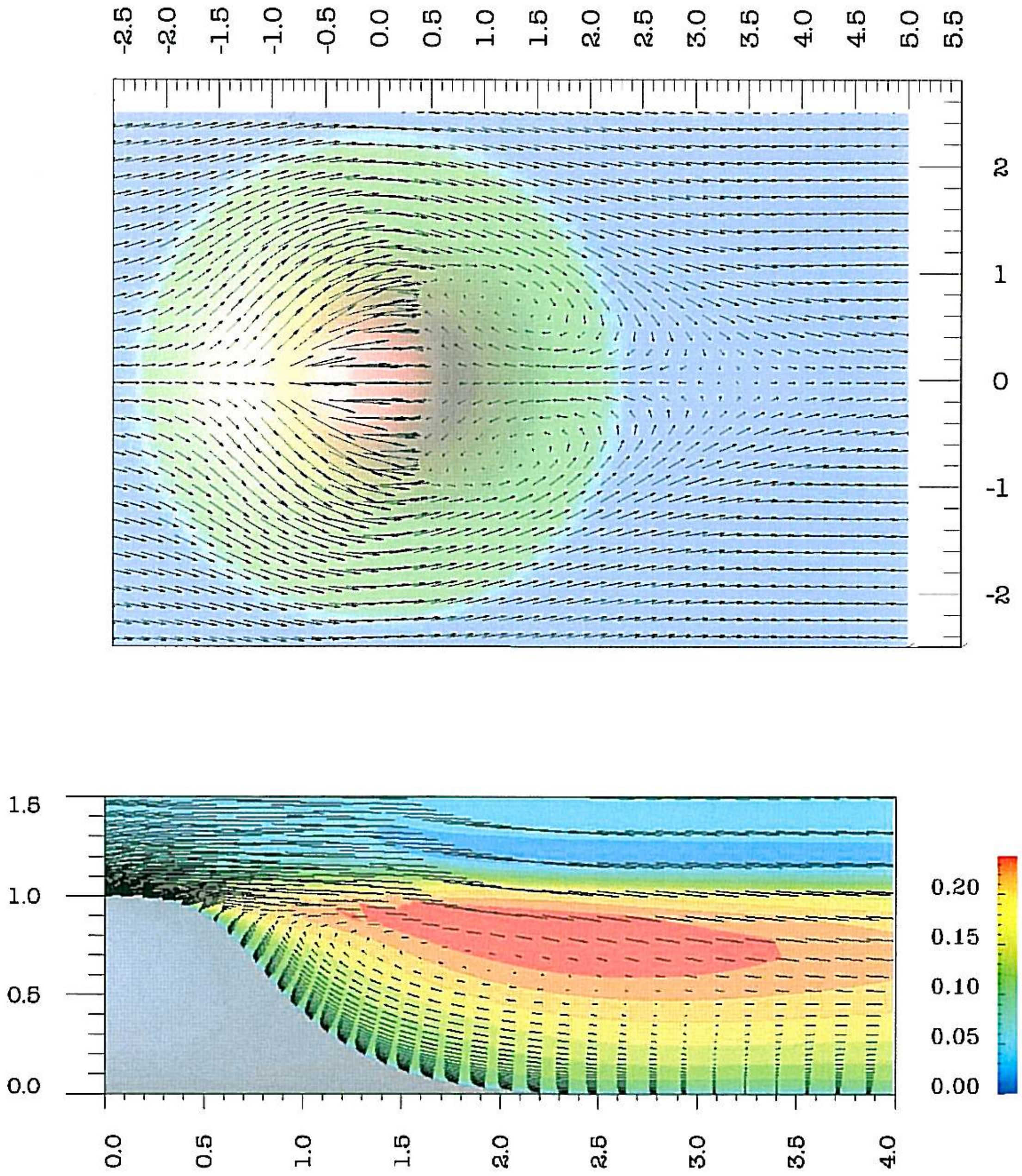


Figure 8: Flow around a 3D hill. Upper figure: Velocity field at near-ground level. Lower figure: Vertical symmetry section of velocity and turbulence intensity. (Arrows are selected, and do not represent mesh nodes.)

6.4.2 Stratified Flow

The geometry is the same as in the previous example, but now the inflow is specified with a temperature stratification given by a Froude number of $Fr = U_\infty/(Nh) = 1$, where U_∞ is the free stream velocity and N is the buoyancy frequency defined by $N^2 = (g/\rho)\partial\theta/\partial z$. The inflow temperature profile corresponding to this is

$$\theta = \theta_o \exp(N^2 z/g).$$

Free conditions are assumed for θ along other side boundaries and along the ground/bottom of the topography.

Results from this simulation are shown in figures 9 and 10. The flow is characterized by an internal wave as illustrated in figure 9. The separation effects from the neutral case has more or less disappeared, although there are some weak effects left as illustrated in figure 10. Similar effects are found in the experiments, see [30]. A quantitative comparison between experimental and computational results is shown in Table 3. Due to the stratification this case is more sensitive, and only small variations in Froude number may give changes in the results. Still the comparison in table 3 shows a reasonable agreement. The simulated wave length is approximately $\lambda/h \approx 5.5$, while the experimental data give $\lambda/h = 6.0 \pm 1.5$.

Table 3: Stratified flow around a three-dimensional hill, Fr = 1

Case	Vortex center $(x,y)/H$	Separation x/H	Recirculation x/H
Experiment	1.2 ± 0.4	0.5	1.6
Computation	0.9 ± 0.7	0.7	1.7

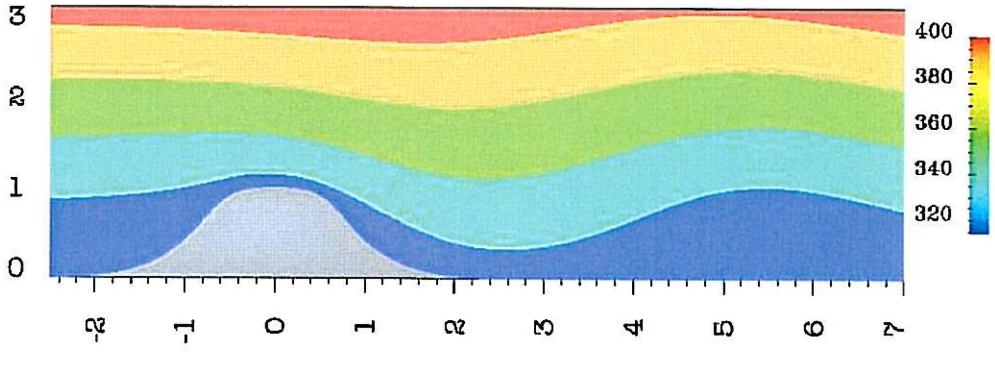
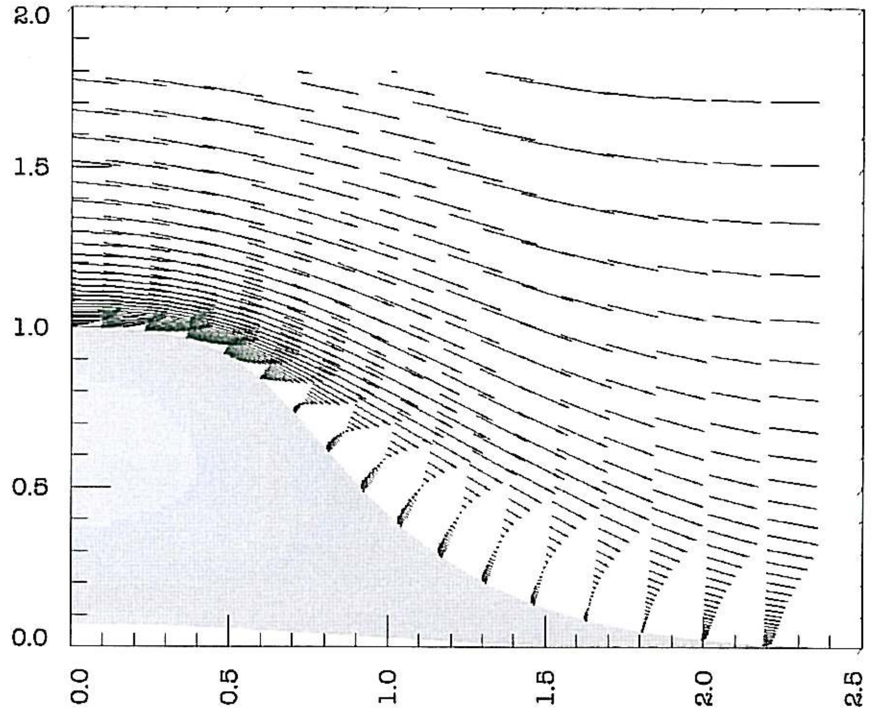


Figure 9: Potential temperature along vertical symmetry section at Fr = 1.



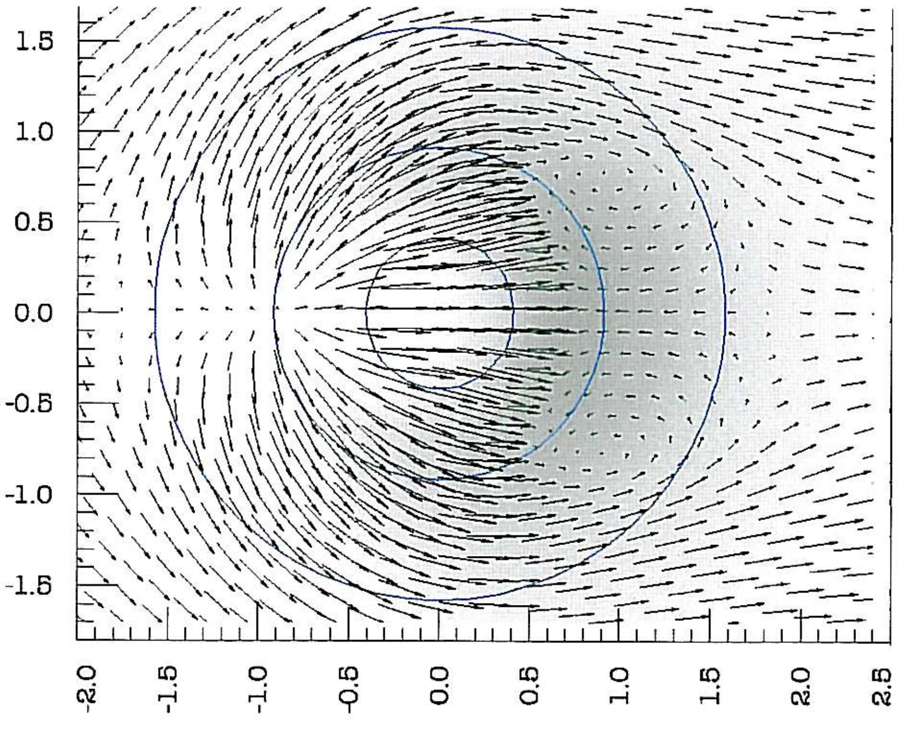


Figure 10: Stratified flow around a 3D hill at Fr = 1. Velocity field along vertical symmetry plane (upper figure) and near-bottom level (lower figure).

6.4.3 Computational efficiency

In order to evaluate the computational efficiency, the convergence rate to steady state is calculated for different time increments. The residual is measured as the relative difference in velocity between each time-step:

$$\epsilon_u = \|u_{n+1} - u_n\| / \|u_n\|,$$

where the Euclidian norm $\|u\| = (\sum_i u_i^2)^{1/2}$ is used. Results from these computations are shown in Fig.11 for three different time steps, corresponding to maximum Courant numbers of 2.5, 5, and 10, respectively. For a given convergence, the number of steps decrease with increasing step size. On the other hand, the cpu per time-step increases with increasing step size due to more iterations. These results are collected in Table 4. The last line in the table shows the same characteristics for a standard Projection 1 (P1) method with explicit advection and implicit diffusion. This formulation is therefore restricted by a Courant number of $Cr \leq 1$, in practice $Cr \leq 0.6$ used with the turbulence model. As can be seen, the SIP model can be a factor of 3 more efficient than P1 in this case, although it needs more cpu per time-step. In addition, it should be noted that a standard Projection 1 formulation of this kind has a steady state solution that is dependent on Δt (cf. the analysis in [6]), while the SIP model has no such bias.

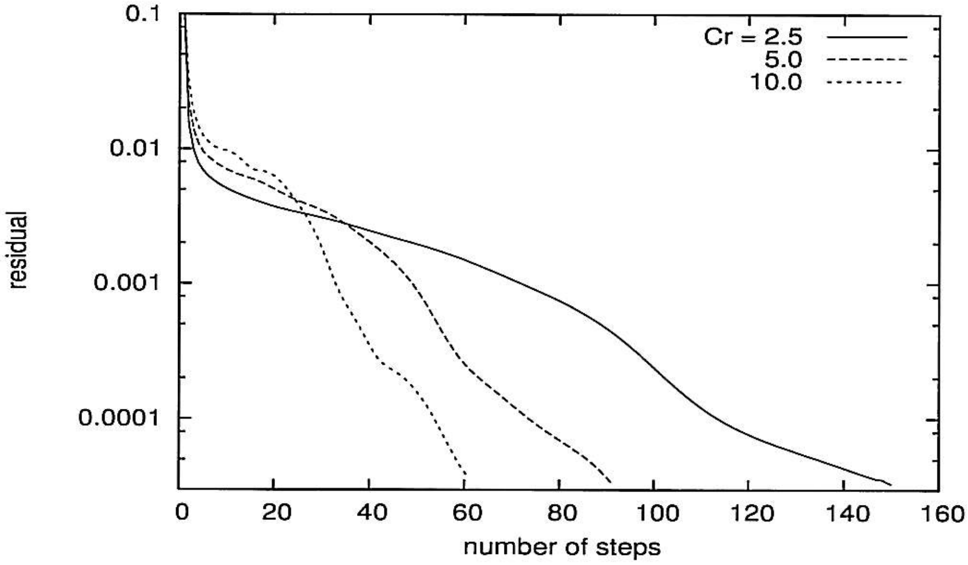


Figure 11: The effect of time increment on the convergence rate to steady state.

7 Concluding remarks

The background for this study has been to investigate numerical algorithms for efficient solution of turbulent flow problems, i.e. flows that can be solved by an unsteady RANS

Table 4: Comparison of efficiency using different models and step size.

Model	Cr-number	CPU/step	Total CPU ($\epsilon_u = 5 \times 10^{-5}$)
SIP	2.5	2.68	402
"	5.0	3.06	275
"	10.0	4.06	244
P1	0.6	1.57	785

formulation. Robustness, efficiency and acceptable accuracy are regarded as key properties. As a possible candidate a segregated, implicit projection method has been chosen for closer investigation. This method has proved to be robust and stable, and it converges to steady state in a reliable manner. The same order of accuracy is obtained as for standard second-order projection methods, namely second-order velocity and first-order pressure. In practical applications all of these properties are regarded as very important.

The method needs more computer time per time-step than a simpler projection method, but the gain in robustness and stability is substantial. For example, compared to a simple semi-implicit projection method (Projection 1) using explicit advection, an overall increased efficiency by a factor of 3 - 4 seems possible. In real case simulations, this factor may even be larger due to the difference in robustness between these two methods.

Computations for various high-Reynolds flows show that the method is robust, and a maximum Courant number in the range 2 - 3 is usually acceptable. It should be noted that these computations also include a two-equation (k, ϵ) turbulence model, which increases the stability requirements substantially.

The computational results show fairly good agreements compared to available experimental data. Hence, although more testing and investigations are needed, the present algorithm seems to be an interesting alternative with several promising qualities.

References

[1] S.V. Patankar, Elliptic Systems: Finite-Difference Methods I, in: G.E. Schneider, R.H. Pletcher (Eds.), Handbook of Numerical Heat Transfer, Wiley, New York, 1988.

[2] G. Comini, W.J. Minkowycz, W. Shyy, General algorithms for the finite element solution of incompressible flow problems using primitive variables, in: W.J. Minkowycz, E.M. Sparrow (Eds.), Advances in Numerical Heat Transfer, Vol 1, Taylor & Francis, Washington, DC, 1997.

[3] V. Haroutunian, M.S. Engelman, I. Hasbani, Segregated finite element algorithms for the numerical solution of large-scale incompressible flow problems, Int. J. Numer. Methods Fluids 17 (1993) 323-348.

[4] J.A. Chorin, Numerical solution of the Navier-Stokes equations. Math. Comput. 22(104) (1968) 745-762.

[5] R. Temam, Sur l'approximation de la solution des equations de Navier-Stokes par la methodes des pas fractionnaires I. Arch. Rat. Mech. Anal. 32 (1969) 135.

[6] P.M. Gresho, S.T. Chan, On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix. Part 2: Implementation, Int. J. Numer. Methods Fluids 11 (1990) 621-659.

[7] J.-L. Guermond, L. Quartapelle, On the approximation of the unsteady Navier-Stokes equations by finite element projection methods, Numer. Math. 80(2) (1998) 207-238.

[8] J. Blsaco, R. Cordina, A. Huerta, A fractional-step method for the incompressible Navier-Stokes equations related to a predictor-multicorrector algorithm, Int. J. Numer. Methods Fluids 28(10) (1998) 1391-1419.

[9] R.I. Issa, Solution of the implicitly discretized fluid flow equation by operator splitting, J. Comput. Phys. 62 (1986) 49-65.

[10] P.J. Oliveira, R.I. Issa, An improved PISO algorithm for the computation of buoyancy-driven flows, Numerical Heat Transfer, Part B 40 (2001) 473-493.

[11] K.-J. Bathe, H. Zhang, A flow-condition-based interpolation finite element procedure for incompressible fluid flows, Computers & Structures 80 (2002) 1267-1277.

[12] K.J. Eidsvik, A. Holstad, I. Lie, T. Utnes, A prediction system for local wind variations in mountainous terrain, Boundary-Layer Meteorology 112 (2004) 557-586.

[13] P.R. Bannon, Potential Vorticity Conservation, Hydrostatic Adjustment, and the Anelastic Approximation, J. Atmos. Sci. 52 (1995) 2302-2312.

[14] P.R. Bannon, Potential vorticity conservation, hydrostatic adjustment, and the anelastic approximation, J. Atmos. Sci. 53 (1996) 3618-3624.

[15] D.R. Durran, Numerical Methods for Wave Equations in Geophysical Fluid Dynamics, Springer, 1998.

[16] W. Rodi, Turbulence models and their application in hydraulics. A state-of-the-art review, IAHR Monograph Series, third ed. A.A. Balkema, Rotterdam, 1993.

[17] W. Rodi, Examples of calculation methods for flow and mixing in stratified fluids, J. Geophys. Res. 92 C5 (1987) 5305-5328.

[18] P.M. Gresho, R.L. Sani, Incompressible Flow and the Finite Element Method, J. Wiley & Sons, West Sussex, UK, 2000.

[19] C. Nonino, G. Comini, An equal-order velocity-pressure algorithm for incompressible thermal flows, Part 1: Formulation, Numer. Heat Transfer 32 (1997) 1-15.

[20] M. A. Christon, Dealing with pressure: FEM solution strategies for the pressure in the time-dependent Navier-Stokes equations, Int. J. Numer. Methods Fluids 38 (2002) 1177-1198.

[21] J. Donea, A. Huerta, Finite element methods for flow problems, John Wiley, 2003.

[22] J. Dukowicz, A. Dvinsky, Approximate factorisation as a higher order splitting for the incompressible flow equations. J. Comput. Phys., 102 (1992) 336-347.

[23] S. Armfield, R. Street, An analysis and comparison of the time accuracy of fractional-step methods for the Navier-Stokes equations on staggered grids, Int. J. Numer. Methods Fluids 38 (2002) 255-282.

[24] D. L. Brown, R. Cortez, M. L. Minion, Accurate Projection Methods for the Incompressible Navier-Stokes Equations. J. Comput. Phys., 168 2 (2001) 464-499.

[25] J.-L. Guermond, J. Shen, On the error estimates for the rotational pressure-correction projection methods. Math. of Comp., 73 248 (2003) 1719-1737.

[26] T. Utnes, T.S. Meling, Treatment of turbulent wall boundary conditions using linear-logarithmic elements, Comput. Methods Appl. Mech. Engrg. 169 (1999) 123-134.

[27] J. Dalsveen, L.R. Sætran, Wind conditions at Værøy Airport. Wind Tunnel Investigations, HOG-r 1991:149(C), Norwegian Institute of Technology (in Norwegian), 1991.

[28] K.J. Eidsvik, T. Utnes, L.R. Sætran, B. Venås, N. Kubberud, H. Nørstrud, Turbulent separated flows over hills, ISBN No 82-7482-022-3, Norweian Institute of Technology, 1994.

[29] D.A. Paterson, J.D. Holmes, Computation of wind flow over topography, J. Wind Engineering and Industrial Aerodynamics, 46 & 47 (1993) 471-476.

[30] J.C.R. Hunt, W.H. Snyder, Experiments on Stably and Neutrally Stratified Flow over a model three-dimensional hill, J. Fluid Mech. 96 (1980) 671-704.

[31] E. Achenbach, E. Heinecke, (1981): On vortex shedding from smooth and rough cylinders in a range of Reynolds numbers 6×10^3 to 5×10^6, J. Fluid Mech. 109 (1981) 239-251.

[32] G. Schewe, On the force fluctuations action on a circular cylinder in cross-flow from subcritical to transcritical Reynolds numbers, J. Fluid Mech. 133 (1983) 265-285.

[33] M.G. Hallam, N.J. Heaf, L.R. Wootton, Dynamics of Marine Structures. CIRIA Underwater Engineering Group, Report UR8, Atkins Research and Development, London, 1977.

[34] E. Achenbach, Distribution of local pressure and skin friction around a circular cylinder in cross-flow up to $Re = 5 \times 10^6$, J. Fluid Mech. 34(4) (1968) 625-639.

NOMENCLATURE OF MAIN VARIABLES

$\mathbf{g}$	gravity
k	turbulent kinetic energy
ℓ	turbulent length scale
p	pressure
t	time
Δt	time-step
$\mathbf{u}$	velocity vector
u, v, w	velocity components (w in vertical direction)
u_*	friction velocity
x, y, z	space variables (z in vertical direction)
z_o	roughness parameter
$A = K + N$	sum of diffusion and advection matrices
C	gradient matrix
Fr	Froude number
K	diffusion matrix
L	Laplacian matrix
M	mass matrix
N	advection matrix
Re	Reynolds number
R_{ij}	Reynolds stress
U_∞	free stream (hight) velocity
α	implicit parameter
γ	diffusivity
δ	boundary layer height
ϵ	turbulent dissipation
ν	viscosity
ρ	density
θ	potential temperature
$\boldsymbol{\tau}$	stress tensor
ϕ	general scalar variable

Subscripts

s	static variable
d	dynamic variable
n	time index
T	turbulence index