Available online at www.sciencedirect.com

ScienceDirect

Journal of Computational Physics 227 (2008) 2198–2211

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

A segregated implicit pressure projection method for incompressible flows

T. Utnes

Sintef ICT Applied Mathematics, N-7465 Trondheim, Norway

Received 12 December 2006; received in revised form 26 October 2007; accepted 4 November 2007
Available online 2 January 2008

Abstract

In this paper a segregated, implicit projection method is presented and investigated. Although high-Reynolds, turbulent flow has been a motivating factor, the present study is mainly restricted to the incompressible Navier–Stokes equations. The proposed method is analyzed for properties like stability, accuracy and consistency, aided by some test examples. In addition, two different flow cases are simulated to evaluate the ability and efficiency of the method compared with other results.

Keywords: Numerical algorithm; Finite elements; Fluid flow

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-like class of algorithms, which are now in use for both finite difference, finite volume and finite element methods [1–3]. 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 projection algorithms [6–9]. 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 [10,11].

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

E-mail address: torbjorn.utnes@sintef.no

0021-9991/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2007.11.022

quality. Although parallelization techniques have improving the computational speed substantially, the core algorithm of the method remains very important. This is especially the case in forecast situations, where a short time window is given 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. The organization of the paper is as follows: Section 2 presents the actual governing equations to be solved, in this case the incompressible Navier–Stokes equations. These equations are discretized (Section 3) using a finite element formulation, and the segregated, implicit solution algorithm is then presented in Section 3. Key properties of the algorithm are analyzed in Section 4. Finally, in Section 5 the method is tested on some high-Reynolds number flows and compared with experimental data.

2. Governing equations

The governing equations are the Navier–Stokes equations for incompressible flow, and we include a scalar advection–diffusion equation for later use:

$$\frac{\mathrm{D}\mathbf{u}}{\mathrm{D}t} = -\nabla P + \nabla \cdot \boldsymbol{\tau} + \mathbf{f}; \quad \nabla \cdot \mathbf{u} = 0, \tag{1}$$

$$\frac{\mathrm{D}\theta}{\mathrm{D}t} = \nabla \cdot (\gamma \nabla \theta) + q. \tag{2}$$

Here $(\mathbf{u}, P, \theta)$ represent velocity, pressure divided by density, and an arbitrary scalar quantity, respectively. Furthermore, $\boldsymbol{\tau}$ is the stress tensor, $\mathbf{f}$ and q are source terms, and γ is a diffusivity.

The governing equations (1) and (2) are discretized in space by use of a mixed finite element method, and these semi-discretized equations may be written in compressed form as

$$M\dot{u} + A(u)u + CP = f; \quad C^{\mathrm{T}}u = 0, \tag{3}$$

$$M_\theta \dot{\theta} + A_\theta(u)\theta = s. \tag{4}$$

The time integration is performed using a semi-implicit two-niveau formulation, and the discretized equation system may then be written in the following form:

$$[\widetilde{M} + \alpha A_{n+\alpha}]u_{n+1} = [\widetilde{M} - (1-\alpha)A_{n+\alpha}]u_n - CP_{n+1} + f_{n+\alpha}, \tag{5}$$

$$C^{\mathrm{T}}u_{n+1} = 0,$$

$$[\widetilde{M}_\theta + \alpha A_{\theta,n+\alpha}]\theta_{n+1} = [\widetilde{M}_\theta - (1-\alpha)A_{\theta,n+\alpha}]\theta_n + s_{n+\alpha}. \tag{6}$$

In these equations M represents the mass matrix, where $\widetilde{M} = M/\Delta t$, $A_{n+\alpha} = K + N(u_{n+\alpha})$ is the sum of diffusion and advection matrices, respectively, C is the gradient matrix, and (f, s) represent source terms. The variables (u, P) are re-defined here as nodal vectors for velocity and scaled pressure, and θ represents the nodal vector for a scalar variable. Subscripts indicate time levels, and superscripts are used to distinguish similar matrices in the momentum and scalar equations. The implicit parameter α is chosen in the interval $1/2 \leqslant \alpha \leqslant 1$, and we use the notation $f_{n+\alpha} = \alpha f_{n+1} + (1-\alpha)f_n$. The global matrices associated with each velocity component of the Navier–Stokes equations are given as

$$M_i = \int \phi\phi^{\mathrm{T}}; \quad K_i = v\int \nabla\phi \cdot \nabla\phi^{\mathrm{T}}; \quad C_i = -\int \nabla\phi\psi^{\mathrm{T}}; \quad N_i(u) = \int \phi\mathbf{u} \cdot \nabla\phi^{\mathrm{T}},$$

where the test/basis function ϕ, ψ are associated with velocity and pressure, respectively, and v is the kinematic viscosity. The advection matrix may alternatively be expressed in skew-symmetric form as

$$N_{i,s}(u) = \int \phi\mathbf{u} \cdot \nabla\phi^{\mathrm{T}} + \frac{1}{2}\int \phi\phi^{\mathrm{T}}\nabla \cdot \mathbf{u},$$

which is especially useful when analyzing stability and energy conservation properties.

3. 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-like iterative methods [2,3], non-iterative pressure-correction procedures like the PISO method [10,11], and several projection formulations [6–9,13].

In this study we propose a segregated, implicit projection method that is non-iterative, with corrections within each time-step. For simplicity the algorithm is denoted PMP in the following, reflecting the three main steps: Pressure prediction, Momentum, Projection. This algorithm has several features in common with the SIMPLER type of pressure projection method described in [3], but instead of iterations it applies corrections similar to the PISO method.

3.1. PMP algorithm

The proposed algorithm is given by the following steps:

Predict the pressure and advection velocity via a pseudo-velocity prediction from the system

$$\widetilde{M}\Delta u_* = -A_n u_n + f_n; \quad LP_* = C^{\mathrm{T}} u_*, \tag{7}$$

where $\Delta u_* = u_* - u_n$, and L represents a scaled discretization of the Laplacian operator, cf. Remarks below.

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

$$[\widetilde{M} + \alpha A_{n+\alpha}]\Delta\tilde{u}_{n+1} = -A_{n+\alpha}u_n + f_{n+\alpha} - CP_*, \tag{8}$$

where $\Delta\tilde{u}_{n+1} = \tilde{u}_{n+1} - u_n$, and the advection velocity is given by extrapolation as $\hat{u}_{n+\alpha} = (1+\alpha)u_n - \alpha u_{n-1}$, but see Remark 2.

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

$$L\,\delta P_{n+1} = C^{\mathrm{T}}\tilde{u}_{n+1}; \quad \widetilde{M}(u_{n+1} - \tilde{u}_{n+1}) = -C\,\delta P_{n+1}, \tag{9}$$

where the updated pressure is $P_{n+1} = P_* + \delta P_{n+1}$.

Compute (semi)implicit equations for other scalar quantities:

$$[\widetilde{M}_\theta + \alpha A_{\theta,n+\alpha}]\Delta\theta_{n+1} = -A_{\theta,n+\alpha}\theta^n + s_{n+\alpha}, \tag{10}$$

where $\Delta\theta_{n+1} = \theta_{n+1} - \theta_n$.

This algorithm has some similarities to that of Nonino and Comino [17], 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 considered appealing, and has been a motivating factor in proposing the present algorithm.

It should be noted that this approach requires an extra pressure equation compared to classic projection methods. This implies extra cost, although moderate in the present context, see Remark 5 below. Other modified, pressure-free projection methods, like those given in [14,15], have been developed using the classical pressure correction with only one Poisson equation, while modified boundary conditions are needed in the prediction step to obtain second-order. Another interesting approach is shown in [16], where an exact fractional step method is developed (no splitting error) using low-order finite volume basis functions which satisfy discrete incompressibility.

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 is calculated via extrapolation as shown above. Alternatively, we may use the correction step in a complete Projection 1: $\hat{u}_{n+\alpha} = u_n - \alpha\Delta t\widetilde{M}^{-1}CP_*$. This alternative may give a complete two-level version.
(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 $L = \Delta tC^{\mathrm{T}}M_{\mathrm{L}}^{-1}C$, 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]).
(5) The extra pressure Poisson equation (PPE) required in the present approach implies some extra cost. However, for the applications of interest here, typically three-dimensional (3D) turbulent flow with temperature variations included, the main cpu requirement goes to solve these advection–diffusion equations (ADE). These ADEs are treated implicitly, and the coefficient matrices must be updated due to both advection terms and variable eddy viscosity. As an example, when solving a 3D Navier–Stokes problem by the present model, less than 20% of the total cpu is used to solve one PPE. The addition of one extra PPE increases the total pressure part to roughly 30%, and this implies that the total increase in cpu using two PPE instead of one is about 20%. Results by Christon [18] for a similar 3D problem, using a semi-implicit incremental projection method, show that the PPE requires about 25% of the total cpu. In that case only the symmetric, constant diffusion part of the momentum equation is treated implicitly, and hence this corresponds reasonable well with our experience. Finally, by including turbulence and temperature in a RANS version of this model, the extra cost is reduced to a fraction of roughly 10%, which is moderate.

3.2. 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. [18]. This is especially advisable when solving the pressure equation iteratively, which is the present case due to three-dimensional applications.

4. 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 the mass matrix is assumed to be lumped, although it is possible to modify this, cf. Remark 4.

4.1. 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 (3). For convenience these equations are re-written here in the form

$$M\dot{u} + A(u)u + CP = f; \quad C^{\mathrm{T}}u = g, \tag{11}$$

where the notation is similar to that in (3), except that 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 analyzing the solution algorithm, we may write the solution of (11) in a 'continuous projection' form as follows: From the momentum equation (11a) the pressure may be expressed as

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

where $L = C^{\mathrm{T}}M^{-1}C$, and by using the continuity constraint from (11b), we have

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

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

$$\dot{u} = QM^{-1}[f - A(u)u] + M^{-1}CL^{-1}\dot{g}, \tag{12}$$

where we have introduced a projection matrix defined as

$$Q := I - M^{-1}CL^{-1}C^{\mathrm{T}}. \tag{13}$$

In order to assure consistency with (11), we therefore need to show that the PMP algorithm is a consistent approximation to (12), as $\Delta t \to 0$.

The PMP algorithm (with $\alpha = 1$ for simplicity) applied to (11) is given by

Prediction:

$$Mu_* = Mu_n - \Delta t(A_n u_n + f_n); \quad LP_* = C^{\mathrm{T}}u_* - g_{n+1}. \tag{14}$$

Momentum:

$$(M + \Delta t A_n)\tilde{u}_{n+1} = Mu_n + \Delta t(f_{n+1} - CP_*). \tag{15}$$

Projection:

$$Mu_{n+1} = M\tilde{u}_{n+1} - \Delta t C\delta P; \quad C^{\mathrm{T}}u_{n+1} = g_{n+1}. \tag{16}$$

The projection step (16) is performed by solving the two equation systems

$$L\,\delta P = C^{\mathrm{T}}\tilde{u}_{n+1} - g_{n+1}; \quad u_{n+1} = \tilde{u}_{n+1} - M^{-1}C\,\delta P.$$

By substituting for δP, this may be expressed as

$$\begin{aligned} u_{n+1} &= \tilde{u}_{n+1} - M^{-1}CL^{-1}C^{\mathrm{T}}\tilde{u}_{n+1} + M^{-1}CL^{-1}g_{n+1} = (I - M^{-1}CL^{-1}C^{\mathrm{T}})\tilde{u}_{n+1} + M^{-1}CL^{-1}g_{n+1} \\ &= Q\tilde{u}_{n+1} + M^{-1}CL^{-1}g_{n+1}, \end{aligned} \tag{17}$$

where we have applied the projection operator (13).

From the momentum equation (15) 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

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

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 (17), the following is obtained:

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

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

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

By letting $\Delta t \to 0$, it is seen that this expression is consistent with (12), and the algorithm is therefore consistent with the desired system (11). The same conclusion is valid for the time-centered case $\alpha = 1/2$.

4.2. Time accuracy

It is well known that a Backward Euler (BE) time discretization gives a global error of order $\mathrm{O}(\Delta t)$, while e.g. a Crank–Nicolsen (CN) scheme gives $\mathrm{O}(\Delta t^2)$. This also holds for direct solution of the coupled Navier–Stokes equations, and applies to both velocity and pressure (see [13]). Similar results are obtained using incremental projection methods, while most non-incremental projection methods are restricted to $\mathrm{O}(\Delta t)$ due to a splitting error (cf. [8,7,20]).

The time accuracy for the PMP method is analyzed by investigating the time-centered ($\alpha = 1/2$) case. The momentum equation can then be written:

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

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

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

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

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

implying an additional term in the PMP formulation given by

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

This term is $\mathrm{O}(\Delta t^3)$ assuming $\delta P = \mathrm{O}(\Delta t)$ (cf. [6]), i.e. the same order of magnitude as the truncation error in the CN formulation. Hence the time-centered PMP formulation should give second-order accuracy. A similar result is also obtained by using a second-ordered BDF2 scheme.

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. The reference solution is obtained from a simulation using a very small time-step (1/10 of the smallest test simulations), similar to the approach in [7,20] (cf. also [13]). 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. [20] with references therein). It is interesting to note that the magnitude of the pressure error seems to be about 5 times smaller than a comparable P2 result shown in Armfield and Street [20]. Possible modifications to improve the pressure accuracy is discussed in [21,22], but generally less than an order of magnitude is obtained.

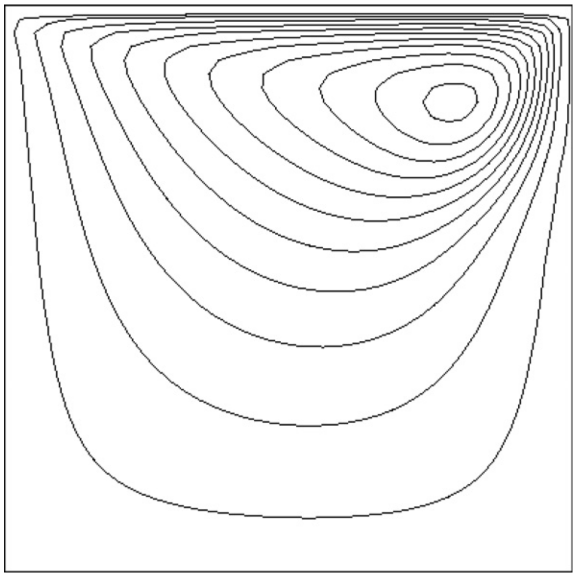


Fig. 1. Computed streamlines for driven cavity flow at time $t = 2$ s, $Re = 400$.

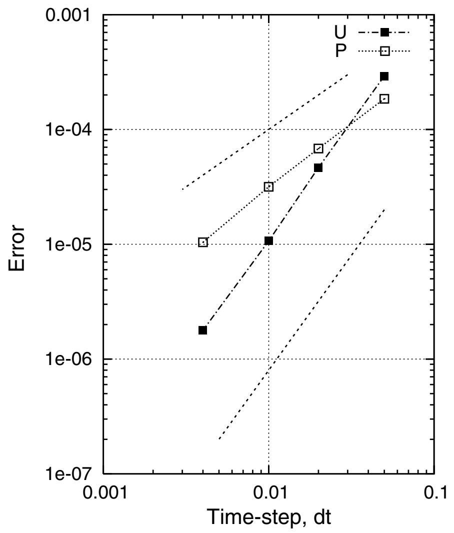


Fig. 2. L_2 error estimates for velocity and pressure, time-centered PMP formulation. Steepness lines for Δt and Δt^2 are added as stipled lines.

4.3. 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 analyzing the present formulation it may be shown that the steady state solution is indeed independent of the time step, and is consistent with the steady state formulation. For simplicity the Backward Euler formulation (14)–(16) is used, but the final result is exactly the same for $\alpha = 1/2$.

From equation (16a) 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 (15) 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 (14) and (16). 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:

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

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

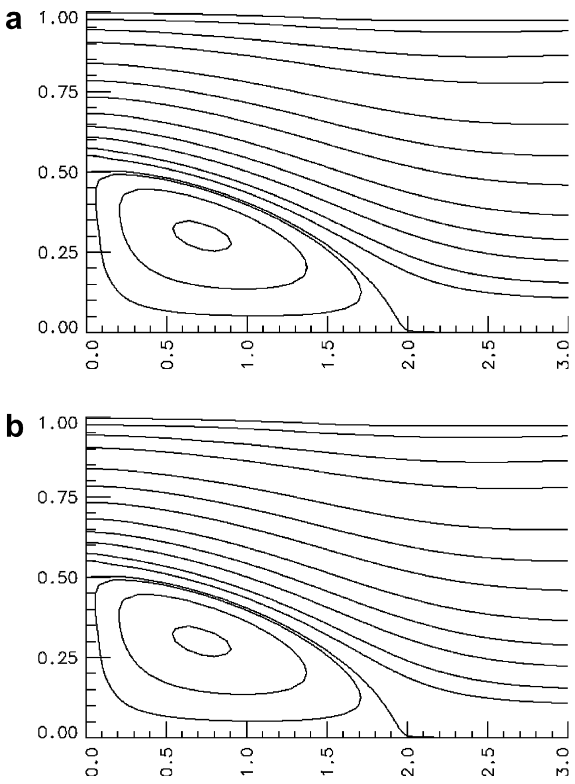


Fig. 3. Steady state solutions represented as stream function. The upper figure is for a time-step corresponding to a maximum Courant number of $Cr = 0.12$, the lower figure with $Cr = 24.84$.

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 observed 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.

4.4. 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 (11) as

$$M\dot{u} + Ku + N_s(u)u + CP = f, \quad C^{\mathrm{T}}u = 0, \tag{18}$$

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

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

and source terms are neglected, $f = 0$. In addition appropriate initial conditions are assumed. The general PMP algorithm applied to this formulation is as follows:

Prediction:

$$M(u_* - u_n) = -\Delta t(K + N_n)u_n; \quad LP_* = C^{\mathrm{T}}u_*. \tag{19}$$

Momentum:

$$M(\tilde{u}_{n+1} - u_n) + \Delta t(K + N_{n+\alpha})\tilde{u}_{n+\alpha} = -\Delta t CP_*. \tag{20}$$

Projection:

$$M(u_{n+1} - \tilde{u}_{n+1}) + \Delta t C\,\delta P_{n+1} = 0; \quad C^{\mathrm{T}}u_{n+1} = 0. \tag{21}$$

The semi-implicit velocity in (20) is given by $\tilde{u}_{n+\alpha} = \alpha\tilde{u}_{n+1} + (1-\alpha)u_n$.

For stability analysis, we use the energy method and investigate the time-centered case $\alpha = 1/2$. Premultiplying equation (20) by $(\tilde{u}_{n+1} + u_n)^{\mathrm{T}}$, yields after some manipulation:

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

where condition $n \cdot \tilde{u}_{n+1}|_{\Gamma} = 0$ has been imposed to eliminate advection terms.

Secondly, (21a) is premultiplied by $2\Delta t(CP_*)^{\mathrm{T}}M^{-1}$, resulting in the expression

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

where the continuity condition $C^{\mathrm{T}}u_{n+1} = 0$ has also been imposed. Finally, (21a) is premultiplied by u_{n+1}^{T}, leading to

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

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 assembling these results, we obtain:

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

where the continuity condition has again been applied. By re-arranging the pressure term and exploiting the positive-definite diffusion matrix, this implies

$$u_{n+1}^{\mathrm{T}} M u_{n+1} + \Delta t^2 P_*^{\mathrm{T}} L P_{n+1} \leqslant u_n^{\mathrm{T}} M u_n + \Delta t^2 P_*^{\mathrm{T}} L P_*. \tag{22}$$

The pressure prediction is given from (19) as $P_* = \Delta t L^{-1} C^{\mathrm{T}} M^{-1} (K + N_n) u_n$. Hence, the estimate (22) is restricted by the previous velocity, and the stability is secured provided appropriate initial conditions.

5. Computational results

5.1. 2D flow around a circular cylinder at Reynolds number $R_D = 400$

This case represents time-dependent flow with separation and vortex shedding. The Reynolds number is chosen as

$$R_D = U_\infty D / \nu = 400,$$

where U_∞ is the free stream velocity, D is the cylinder diameter, and the computational domain is given by $-5 \leqslant x/D \leqslant 15$, $-5 \leqslant y/D \leqslant 5$. The mesh consists of 11832 nodes, and the minimum distance in the radial direction from the cylinder surface is $0.005D$. With the present Reynolds number this resolution may be somewhat coarse, however, the main purpose here is to test the efficiency and stability of the method.

The present (PMP) method has been applied over a range of time increments, corresponding to maximum Courant numbers $Cr \in (1, 5)$. The largest of these yields less than 50 time-steps over one vortex shedding period, and is regarded as fairly coarse. Fig. 4 illustrates results from the computations for different time-steps. The figure shows the time evolution of drag and lift coefficients, defined as

$$C_{\mathrm{D}} = \frac{F_X}{0.5\rho D U_\infty^2}; \quad C_{\mathrm{L}} = \frac{F_Y}{0.5\rho D U_\infty^2},$$

where F_X and F_Y are the in-line and transverse force components, respectively. It is seen that the force coefficients are rather insensitive to variations in time-step within these limits. However, for the largest time-step

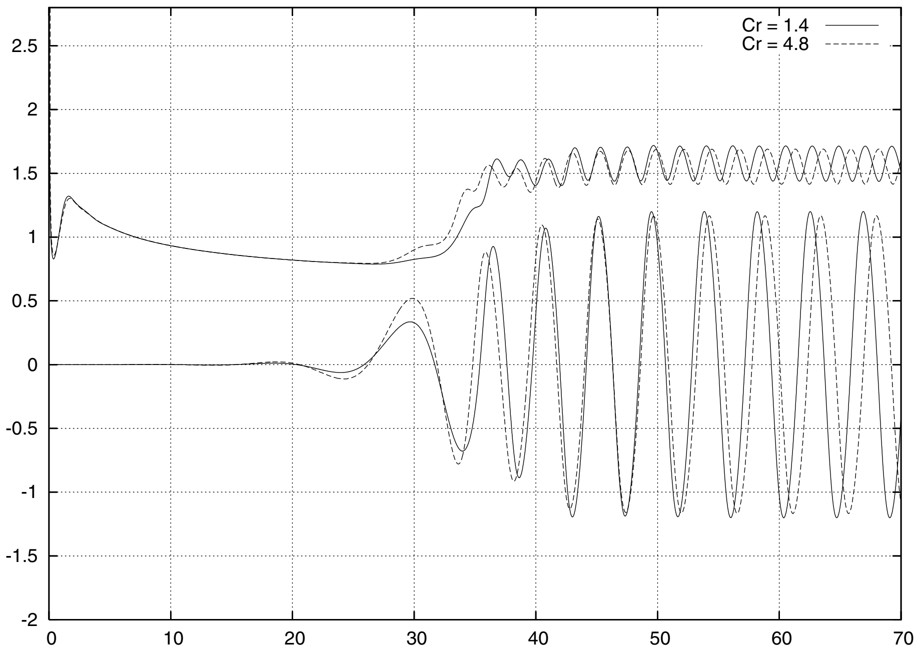


Fig. 4. Force coefficients (C_{D}, C_L) on a circular cylinder at $R_D = 400$. PMP results for time increments corresponding to $Cr = 1.4$ and 4.8, respectively.

used, a weak damping is observed in the coefficients, and the corresponding shedding period is then somewhat increased.

For comparison the same problem has been simulated with a Projection 1 (P1) formulation using implicit diffusion and explicit advection [6]. The stability is then limited by $Cr \leqslant 1$, and the present computation is with $Cr \approx 0.7$. These results are collected in Table 1, including the computer time needed per vortex shedding period (T). The vortex shedding frequency in the table is given in non-dimensional form as the Strouhal number $St = fD/U_\infty$, where f is the frequency [Hz].

It is seen that although the PMP model needs more cpu per time-step, it may gain in efficiency compared to the simpler P1 method. The limiting factor in this case is accuracy rather than stability; a minimum number of time-steps is needed to resolve one vortex shedding period.

Corresponding experimental data for mean drag and Strouhal number are about 1.4 and 0.18, respectively [23]. For this relatively high-Reynolds number three-dimensional turbulence effects are present in the flow, which may explain some of the discrepancy between computational results and experiments.

5.2. Turbulent flow around a three-dimensional hill

In order to illustrate the ability of the method, we include a more realistic computational example, represented by a three-dimensional turbulent flow around a hill. The turbulence is modeled by a standard (k, ϵ) closure, and is here merely taken as given (see e.g. [24] for an extensive analysis). The turbulence equations may be regarded as additional scalar equations in the algorithm, although this actually implies extra non-linearities, several complications and more stability demands. However, the main point here is to show the ability of the algorithm applied to such flows.

The example is based on laboratory experiments [25], and concerns the flow over a hill with 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 are specified using a logarithmic velocity profile and corresponding profiles for turbulent kinetic energy (k) and dissipation (ϵ):

$$u = \frac{u_*}{\kappa}\ln\left(\frac{z}{z_0}\right); \quad k = \max\{C_\mu^{-1/2}u_*^2(1-z/\delta)^2, k_{\min}\},$$
$$\epsilon = C_\mu^{3/4}k^{3/2}/\ell; \quad \text{where } \ell = \frac{\kappa z}{1+4z/\delta}.$$

Here z is the vertical coordinate, z_0 is a roughness parameter, δ is the boundary layer thickness, $k_{\min}$ is a minimum turbulence, $C_\mu = 0.09$, $\kappa = 0.4$, and u_* is the friction velocity. Terrain boundaries are assumed to be rough, and standard wall conditions are used:

$$\frac{u_t}{u_*} = \frac{1}{\kappa}\ln\left(\frac{d}{z_0}\right); \quad u_n = 0,$$

Table 1
Cylinder flow at $R_D = 400$: Comparison of results

Model	Max Cr	$\overline{C}_D$	C_L	St	CPU/T
PMP	1.4	1.56	±1.21	0.22	18.6
PMP	2.9	1.56	±1.21	0.22	11.6
PMP	4.8	1.54	±1.20	0.21	8.5
P1	0.7	1.56	±1.21	0.22	14.7

where (u_t, u_n) are the tangential and normal velocity components at a small distance d from the wall, and z_0 is the roughness parameter. Near-wall conditions for the turbulence variables are given by

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

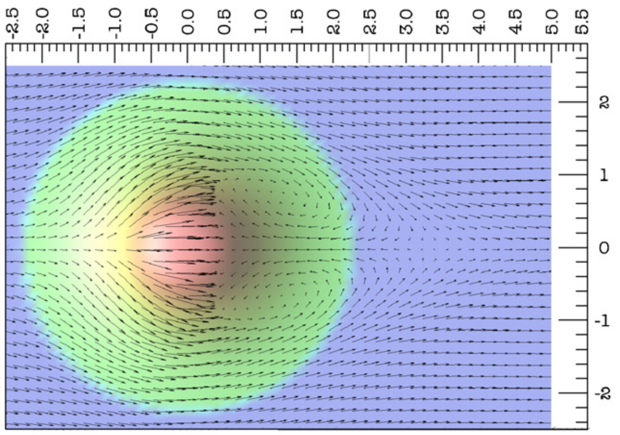
For the inflow conditions the boundary layer height is $\delta = 0.3H$. The Reynolds number is $Re = 10^4$ based on the hill height, and the roughness parameter is given by $z_0/H = 0.0003$. The computational domain is given by $-6 \leqslant x/H \leqslant 8$, $-5 \leqslant y/H \leqslant 5$, $0 \leqslant z/H \leqslant 7$, and the shown computation is performed on a structured mesh consisting of $(100 \times 100 \times 50)$ tri-linear elements, with clustering around the hill and to-wards the boundary layer.

Computational results are shown in Fig. 5. The upper figure illustrates velocity field at a near-ground section, while the lower figure shows velocity and turbulence intensity along the vertical symmetry plane. The flow is characterized by two symmetric eddies downstream of the hill, as illustrated in Fig. 5a; and along the symmetry plane these eddies combine to a back-flow circulation (Fig. 5b). 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 reasonable agreement.

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:

$$\epsilon_u = \|\mathbf{u}_{n+1} - \mathbf{u}_n\| / \|\mathbf{u}_n\|,$$

where the Euclidean norm $\|\mathbf{u}\| = \left(\sum_i \mathbf{u}_i^2\right)^{1/2}$ is used. Results from these computations are shown in Fig. 6 for three different time steps, corresponding to maximum Courant numbers of 2.5, 5, and 10, respectively. For a

a


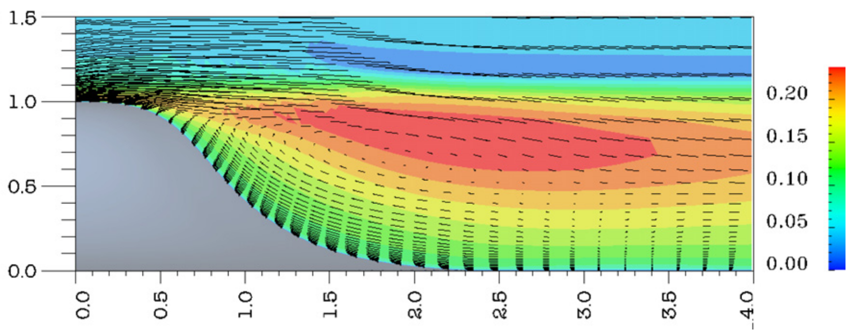
b


Fig. 5. Flow around a 3D hill. Upper figure: Velocity field at near-ground level. Lower figure: Vertical symmetry section of velocity and turbulence intensity $(\sqrt{k}/U_\infty)$.

Table 2
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

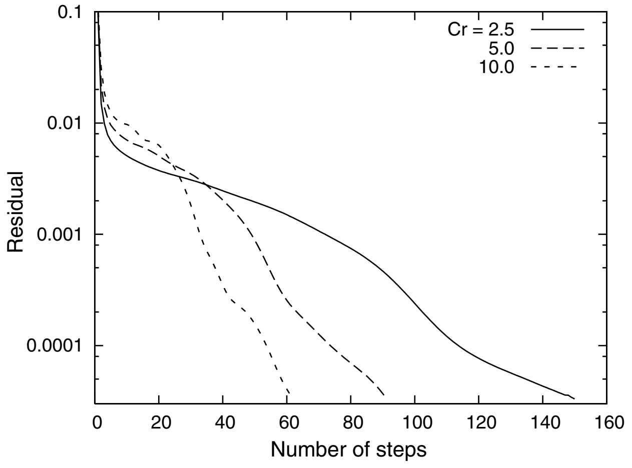


Fig. 6. The effect of time increment on the convergence rate to steady state.

Table 3
Comparison of efficiency using different algorithms and step size

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

given convergence tolerance, 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 inner iterations in the solver. These results are collected in Table 3. 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 $Cr \leqslant 1$, in practice $Cr \leqslant 0.6$ used with the turbulence model. As can be seen, the PMP 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 PMP model has no such bias, as previously shown (Section 4.3).

6. Concluding remarks

In this paper a segregated, implicit projection method has been presented and investigated. Although high-Reynolds, turbulent flow has been a motivating factor, the present study is basically restricted to the incompressible Navier–Stokes equations, which is at the core of any such model. A special property of the method is that although it resembles incremental projection methods, the pressure is predicted anew for each time-step, and therefore avoids accumulating memory effects. This has been a motivating factor in proposing the present alternative algorithm.

The method has proved to be robust and stable, and it converges to steady state in a reliable manner. The same order of accuracy can be 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. Compared to a simple semi-implicit projection method (Projection 1) with explicit advection, an overall increased efficiency by a factor of 2–3 seems possible. Experience with turbulent flows over much more complicated geometries (not shown here) seems to pronounce this effect, even in comparison with a stabilized (balancing tensor diffusion) P1 model. However, the use of two pressure Poisson equations implies more cost per time-step than a classical implicit projection method, although the extra work is a fairly moderate fraction of the total cost required in applications to three-dimensional turbulent flows.

Acknowledgement

The present study has been supported by the Norwegian Civil Aviation Administration (AVINOR).

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 and 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. Dukowicz, A. Dvinsky, Approximate factorisation as a higher order splitting for the incompressible flow equations, J. Comput. Phys. 102 (1992) 336–347.
[8] 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.
[9] 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.
[10] R.I. Issa, Solution of the implicitly discretized fluid flow equation by operator splitting, J. Comput. Phys. 62 (1986) 49–65.
[11] P.J. Oliveira, R.I. Issa, An improved PISO algorithm for the computation of buoyancy-driven flows, Numer. Heat Transfer Part B 40 (2001) 473–493.
[12] K.J. Eidsvik, A. Holstad, I. Lie, T. Utnes, A prediction system for local wind variations in mountainous terrain, Bound.-Layer Meteorol. 112 (2004) 557–586.
[13] P.M. Gresho, R.L. Sani, Incompressible Flow and the Finite Element Method, J. Wiley and Sons, West Sussex, UK, 2000.
[14] J. Kim, P. Moin, Application of a fractional step method to incompressible Navier–Stokes equations, J. Comput. Phys. 59 (1985) 308–323.
[15] J.B. Perot, An analysis of the fractional step method, J. Comput. Phys. 108 (1993) 249–253.
[16] W. Chen, F. Giraldo, B. Perot, Analysis of an exact fractional step method, J. Comput. Phys. 180 (2002) 183–199.
[17] C. Nonino, G. Comini, An equal-order velocity-pressure algorithm for incompressible thermal flows, Part 1: Formulation, Numer. Heat Transfer 32 (1997) 1–15.
[18] 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.
[20] 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.
[21] 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.
[22] J.-L. Guermond, J. Shen, On the error estimates for the rotational pressure-correction projection methods, Math. Comput. 73 (248) (2003) 1719–1737.
[23] H. Oertel Jr., Wakes behind blunt bodies, Annu. Rev. Fluid Mech. 22 (1990) 539–564.
[24] B. Mohammadi, O. Pironneau, Analysis of the K-Epsilon Turbulence Model, J. Wiley and Sons, Masson, 1994.
[25] 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.