INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 21, 837–856 (1995)

A LITTLE MORE ON STABILIZED Q_1Q_1 FOR TRANSIENT VISCOUS INCOMPRESSIBLE FLOW

P. M. GRESHO, S. T. CHAN, M. A. CHRISTON AND A. C. HINDMARSH

Lawrence Livermore National Laboratory, Livermore, CA 94551, U.S.A.

SUMMARY

In an attempt to overcome some of the well-known 'problems' with the Q_1P_0 element, we have devised two 'stabilized' versions of the Q_1Q_1 element, one based on a semi-implicit *approximate* projection method and the other based on a simple forward Euler technique. While neither one conserves mass in the most desirable manner, both generate a velocity field that is usually 'close enough' to divergence-free. After attempting to analyse the two algorithms, each of which includes some ad hoc 'enhancements', we present some numerical results to show that they both seem to work well enough. Finally, we point out that any projection method that uses a 'pressure correction' approach is inherently limited to *time-accurate* simulations and, even if treated fully implicitly, is inappropriate for seeking steady states via large time steps.

KEY WORDS: stabilized finite elements; projection method; approximate projections; equal-order interpolation

1. INTRODUCTION

In our quest for a more efficient method for solving time-accurate, viscous incompressible flows over arbitrarily complex geometries such as submarines and automobiles, and perhaps to get back 'in vogue', we have made a brief investigation into the possibility and feasibility of converting our Q_1P_0 (bilinear velocity, piecewise-constant pressure) semi-implicit GFEM code based on a (mostly) second-order projection method into a stabilized *equal-order* interpolation code, namely Q_1Q_1. The principal 'drive' behind our effort was the hope that a simpler 'Laplacian' matrix, used each time step to solve a pressure Poisson equation (PPE), would noticeably reduce the number of conjugate gradient iterations (with diagonal preconditioning) needed, the computing time for which consumes the great bulk (say 80 per cent $\pm$ 10 per cent) of the cost of each time step. A secondary objective is related to the potentially inaccurate approximation to ∇P when using Q_1P_0 on general meshes via distorted isoparametric elements;[1,2] a bilinear approximation for pressure is obviously more accurate than one that is constant over each element.

The desire to use (stable) equal-order interpolation is easy to understand and it is therefore not surprising that much effort has gone into the quest. Thus we begin by listing a sample of related history—some finite element, some finite difference, and some finite volume, but most finite element.[3–19] The 'fixes' in these papers are many and varied and herein we present our own—a 'fix' that we admit to being 'partial' at best and, probably like some of those used in the past (Galerkin/least squares methods excepted), not completely 'understood'. Our fix, like that of many others, is based on the notion of an 'approximate' projection in which we stabilize an otherwise unstable velocity–pressure pair by sacrificing discrete mass conservation. In fact, the nice term 'approximate projection' came from outside the FEM community.[16,18]

CCC 0271–2091/95/100837–20

Received 17 February 1995
Revised 7 April 1995

In the remainder of this paper we shall describe our attempts at generating a useful Q_1Q_1 stabilized element, beginning with the semi-implicit projection method and ending with an explicit method. Finally, both will be demonstrated via some 2D numerical examples.

2. THEORETICAL DEVELOPMENT

The equations of principal interest are

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} + \nabla P = \nu \nabla^2 \mathbf{u}, \tag{1}$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in} \quad \Omega, \tag{2}$$

with

$$\mathbf{u} = \mathbf{w}(t) \quad \text{on} \quad \Gamma \equiv \partial\Omega, \qquad \int_\Gamma \mathbf{n} \cdot \mathbf{w} = 0, \tag{3}$$

$$\mathbf{u} = \mathbf{u}_0, \quad \nabla \cdot \mathbf{u}_0 = 0 \quad \text{in} \quad \Omega \quad \text{at} \quad t = 0, \qquad \mathbf{n} \cdot \mathbf{u}_0 = \mathbf{n} \cdot \mathbf{w}(0) \quad \text{on} \quad \Gamma, \tag{4}$$

although later we will generalize the boundary conditions (BCs) to permit flow-through.

Since these 'primitive equations' are usually considered as too difficult to solve as stated (computationally, fully coupled; at least in 3D), the following pressure Poisson equation (PPE) formulation is of much interest. Using $\partial(\nabla \cdot \mathbf{u})/\partial t = 0$ in (1) leads, using (2), to the PPE

$$\nabla^2 P = \nabla \cdot (\nu \nabla^2 \mathbf{u} - \mathbf{u} \cdot \nabla \mathbf{u}) \equiv \nabla \cdot \mathbf{a} \quad \text{in} \quad \Omega, \tag{5}$$

with[20]

$$\frac{\partial P}{\partial n} = \mathbf{n} \cdot \left(\nu \nabla^2 \mathbf{u} - \mathbf{u} \cdot \nabla \mathbf{u} - \frac{\partial \mathbf{w}}{\partial t} \right) = \mathbf{n} \cdot \mathbf{a} - \mathbf{n} \cdot \frac{\partial \mathbf{w}}{\partial t} \quad \text{on} \quad \Gamma, \tag{6}$$

where we note that $\mathbf{a} \equiv \nu \nabla^2 \mathbf{u} - \mathbf{u} \cdot \nabla \mathbf{u}$ is a *partial* acceleration (sans ∇P).

The PPE (derived) formulation comprises (1) and (3)–(6), i.e. (2) is omitted because it is implied.[21]

The weak (Galerkin) form of the continuum PPE formulation is

$$\int \mathbf{v} \cdot \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) + \nu (\nabla \mathbf{u})^{\mathrm{T}} : \nabla \mathbf{v} - P \nabla \cdot \mathbf{v} = 0 \quad \forall \mathbf{v} \in \mathbf{H}_0^1, \tag{7}$$

$$\int \nabla \varphi \cdot \nabla P = \int \mathbf{a} \cdot \nabla \varphi - \int_\Gamma \varphi \mathbf{n} \cdot \frac{\partial \mathbf{w}}{\partial t} \quad \forall \varphi \in H^1. \tag{8}$$

Remarks

1. The above PPE formulation assures that $\nabla \cdot \mathbf{u} = 0$ only in a very weak (and vague) sense, in that *if* (7) and (8) $\Rightarrow$ (1) and (3)–(6), then $\nabla \cdot \mathbf{u} = 0$, i.e. *if* our weak solution is also a classical solution, then $\mathbf{u}$ will be (strongly) div-free. These are big 'ifs'.
2. The PPE formulation has more solutions than the $\mathbf{u}-P$ formulation, but the extras are spurious, a particular example being that $\nabla \cdot \mathbf{u}_0$ need not vanish in order that PPE solutions exist—any $\mathbf{u}_0$ 'works'. See Reference 22 for others.
3. The integration by parts of both sides of (5) has resulted in a (legitimate) cancellation of boundary integrals via (6), with the result that $\int \mathbf{a} \cdot \nabla \varphi$ is a (negative) weak divergence of $\mathbf{a}$ in Ω *only*.

4. The OBC (open boundary condition) implied by (7), when a portion of Γ is 'open', i.e. when (3) does not apply on all of Γ, is $\nu\partial\mathbf{u}/\partial n - \mathbf{n}P = \mathbf{0}$, a natural BC, while that associated with (8) is the essential BC $P = 0$; the two are compatible if $\nu\partial u_n/\partial n = 0$, a generally desirable OBC. Stated differently: setting $P = 0$ on an open boundary in this formulation implies that $\partial u_n/\partial n = 0$ there.

If we discretize (1)–(4) via the GFEM, we arrive at the index 2 DAE (differential–algebraic equation) system

$$M\dot{u} + N(u)u + CP = Ku + f, \qquad u(0) = u_0, \tag{9}$$

$$C^{\mathrm{T}}u = g, \qquad C^{\mathrm{T}}u_0 = g_0, \tag{10}$$

where f and g correspond to data from the BCs. Note that (9) also corresponds to the finite-dimensional version of (7). Here C corresponds to 'grad' and C^{T} corresponds to '$-$div';[23] in fact,

$$\left(C^{\mathrm{T}}\right)_{ij} \equiv -\left[\int \psi_i \frac{\partial \varphi_j}{\partial x}, \quad \int \psi_i \frac{\partial \varphi_j}{\partial y}\right],$$

where we remark that φ represents a velocity basis function and ψ a pressure basis function, even though $\psi = \varphi$ in our code. (We distinguish between them in the interest of clarity.) Converting this to a lower-index problem (also easier for time integration) is achieved by inserting $\dot{u}$ into the *time-differentiated* version of (10): the resulting index 1 DAE system is (9) and

$$(C^{\mathrm{T}}M^{-1}C)P = C^{\mathrm{T}}M^{-1}[Ku + f - N(u)u] - \dot{g}, \tag{11}$$

wherein we note that $C^{\mathrm{T}}M^{-1}C$ approximates $-\nabla^2$.

As with the continuum PPE formulation, the discrete index 1 formulation uncouples the pressure from the acceleration (its 'raison d'être'). Also as with the continuous formulation, violation of $C^{\mathrm{T}}u_0 = g_0$ results in an ill-posed index 2 problem, but the index 1 version, having more solutions, is not ill-posed. It is just not *right*: the resulting discrete velocity field will not satisfy $C^{\mathrm{T}}u = g$. (It *will* satisfy $C^{\mathrm{T}}u = g + C^{\mathrm{T}}u_0 - g_0$.[24])

For comparison with what follows below, we now introduce the projection matrix

$$\mathscr{P} \equiv I - M^{-1}C(C^{\mathrm{T}}M^{-1}C)^{-1}C^{\mathrm{T}}, \tag{12}$$

which is a 'formal' construction only, i.e. it is never explicitly formed (nor is M^{-1} unless the mass is lumped). $\mathscr{P}$ has the following properties.

1. $\mathscr{P}^2 = \mathscr{P}$; it satisfies the *definition* of a projection.
2. If $\tilde{u}$ is an arbitrary velocity field, then $\mathscr{P}\tilde{u}$ is, for $g = 0$, an M-orthogonal projection to the (discretely) div-free subspace, i.e. $u \equiv \mathscr{P}\tilde{u}$ satisfies $C^{\mathrm{T}}u = 0$. In fact, $C^{\mathrm{T}}\mathscr{P} \equiv 0$.
3. The eigenvalues of $\mathscr{P}$ are either zero or unity and its norm is unity. All discretely divergence-free vector fields are eigenvectors of $\mathscr{P}$ with eigenvalue unity. (If $C^{\mathrm{T}}u = 0$, then $\mathscr{P}u = u$.)

In terms of $\mathscr{P}$ the index 1 problem can be 'condensed' (again formally) to an index 0 problem (i.e. to a set of ODEs)

$$\dot{u} = \mathscr{P}M^{-1}[Ku + f - N(u)u] + M^{-1}C(C^{\mathrm{T}}M^{-1}C)^{-1}\dot{g}, \tag{13}$$

which clearly satisfies $C^{\mathrm{T}}\dot{u} = \dot{g}$; the PPE method *preserves* the divergence of the initial velocity field. Only an *initially* div-free velocity field will *remain* div-free.

Suppose though that we use instead (7) and (8), which is our plan, to generate our GFEM equations. The result is (9) and

$$LP = h, \tag{14}$$

where

$$L_{ij} \equiv \int \nabla\varphi_i \cdot \nabla\varphi_j \tag{15}$$

is the 'conventional' GFEM Laplacian ($-\nabla^2$, in fact) and φ_k is the basis function (bilinear herein) associated with node k. The RHS vector is

$$h_i \equiv \int \nabla\varphi_i \cdot \left(\nu\nabla^2\mathbf{u}^h - \mathbf{u}^h \cdot \nabla\mathbf{u}^h\right) - \int_\Gamma \varphi_i \mathbf{n} \cdot \frac{\partial \mathbf{w}}{\partial t} \tag{16}$$

from the finite-dimensional form of (8), with $\mathbf{u}^h$ representing the GFEM velocity. Clearly some remedial action is required if C^0 basis functions are to be employed (for which $\nabla^2\mathbf{u}^h$ is not well-defined); for one type of response see Reference 8. For a 'rationalization', note that

$$\nu \int \nabla\varphi \cdot \nabla^2\mathbf{u} = \nu \int_\Gamma \varphi \mathbf{n} \cdot \nabla^2\mathbf{u}$$

when $\nabla \cdot \mathbf{u} = 0$ and thus the *omission* of the viscous term in (16) can *only* 'affect things' near Γ—an especially valid approximation for large Reynolds number (small ν). (See Reference 19 in which the viscous term in (16) *was* omitted, yet good results obtained even for steady Stokes flow! This is the 'biharmonic miracle' *in action.*[25]) In our formulation an approximation to (14)–(16) is, for the projection method at least, actually only required at $t = 0$ to estimate the *initial* pressure field; for $t > 0$ the pressure is determined by other methods that are described below.

Let us summarize our method for computing P_0 given a div-free (or nearly so) velocity field $\mathbf{u}_0$. We begin by returning to (8) with, for convenience, $\partial\mathbf{w}/\partial t = 0$ on Γ (time-independent Dirichlet data) and, realizing that the viscous term in $\mathbf{a}$ cannot be evaluated rigorously, we first set up and solve an 'aside' problem for $\mathbf{a}$—a best L^2 fit to the given data (unless we invoke mass lumping):

$$\int \mathbf{v} \cdot \mathbf{a} = \int \nu\mathbf{v} \cdot \nabla^2\mathbf{u} - \mathbf{v} \cdot (\mathbf{u} \cdot \nabla\mathbf{u}) = \int_\Gamma \mathbf{v} \cdot \frac{\partial \mathbf{u}}{\partial n} - \int [\nu\nabla\mathbf{v} : (\nabla\mathbf{u})^{\mathrm{T}} + \mathbf{v} \cdot (\mathbf{u} \cdot \nabla\mathbf{u})] \quad \forall \mathbf{v} \in \mathbf{H}^1, \tag{17}$$

in which we apply *some* of the same BCs to $\mathbf{a}$ that apply to $\mathbf{u}$. We use $\mathbf{a} = \mathbf{0}$ on Γ_D, while on Γ_N we *assume* $\partial\mathbf{u}/\partial n = \mathbf{0}$ and thus omit the boundary integral. Some consequences of these choices will be discussed below, in addition to the following: we select test functions (in H^1) that vanish on Γ_D. The FEM realization of (17) is, via $\mathbf{a}^h = \sum \mathbf{a}_j\varphi_j$, etc., simply

$$a = M^{-1}[Ku - N(u)u], \tag{18}$$

where a is the vector of nodal 'accelerations', which goes into (8), (14) and (16) to give

$$LP = \tilde{D}a, \tag{19}$$

where $\tilde{D}$ is a 'divergence' matrix to be more carefully defined in Section 4. (Also, we have not yet implemented time-varying Dirichlet BCs.) The 'consequences' referred to above are mainly from the use of $\mathbf{a} = \mathbf{0}$ on Γ_D, which causes a few problems.

(i) For a Stokes flow (or for the viscous part in general) the Neumann BC for the pressure, via forming local nodal equations and letting $h \to 0$, turns out to be $\partial P/\partial n = \frac{1}{2}\nu\mathbf{n} \cdot \nabla^2\mathbf{u}$, with the factor of $\frac{1}{2}$ a direct consequence of using $\mathbf{n} \cdot \mathbf{a} = 0$ on Γ_D.

(ii) However, the error seems to be 'local' in the following sense. A Poiseuille flow test case did indeed produce one-half of the proper pressure gradient *at the inlet only*, i.e. the slope was one-half of the correct value in the first column of elements, but it recovered to virtually the correct value in the remainder of the channel and the velocity field was nearly perfect.

(iii) The 'viscous error' at the boundary is mainly a 'low-Reynolds-number' problem.

Next we introduce and immediately dispense with an idea that we actually tested: just replace C^TM^{-1} by L in (11) and solve the resulting index 1 problem comprising (9) and (11) with $C^TM^{-1}C$ replaced by L. It then follows that $LP = C^TM^{-1}(M\dot{u} + CP) - \dot{g}$, i.e. $C^T\dot{u} - \dot{g} = (L - C^TM^{-1}C)P \neq 0$; this approach can *not* 'preserve the div.' In fact, time integration yields $C^Tu(t) - g(t) = C^Tu_0 - g_0 + (L - C^TM^{-1}C)\int_0^t P(\tau)d\tau$—the div can be 'anything,' even if $C^Tu_0 = g_0$. Also, if a steady state is attained, the pressure field would necessarily satisfy $LP = (C^TM^{-1}C)P$. This was a bad idea; the results of a numerical experiment agreed with the above analysis—and of course the experiment preceded the analysis! Figure 1 shows the result of a lid-driven cavity simulation ($Re = 100$) that went steady state; the velocity is far from div-free and the pressure does indeed satisfy $LP = (C^TM^{-1}C)P$.

We and many others have generated codes based on the index 1 formulation of (9) and (11) using the Q_1P_0 element. However, with the exception of our ad hoc procedure in Reference 26, all Q_1P_0 index 1 codes 'required'—as did ours in the test just described—the (also) ad hoc (and more deleterious) approximation of *mass lumping* (because M^{-1} is otherwise dense), which introduces a *serious loss of accuracy* for the bilinear element when the flow is advection-dominated (vortex shedding, for example). Additional bad features of this element are the following.

1. It suffers from the 'bent element blues'[2] ; i.e. the CP approximation to ∇P is not very accurate when the elements are quite distorted).
2. The staggered mesh 'bookkeeping' is not fun.
3. The Laplacian matrix $C^TM_L^{-1}C$ is 'awkward' and is suspected to converge 'too slowly' when iterative solvers are used. (M_L is the lumped version of M.)
4. It fails the LBB (Ladyzhenskaya–Brezzi–Babushka) test, i.e. the velocity and pressure spaces are not quite compatible. While this is a *fatal* flaw in the eyes of many mathematicians, those 'engineers' who have been bold enough to try it anyway know that it can and does deliver good velocities and—perhaps (when needed, $\tau \cdot \mathbf{u}$ specified on all of Γ) after postprocessing via the

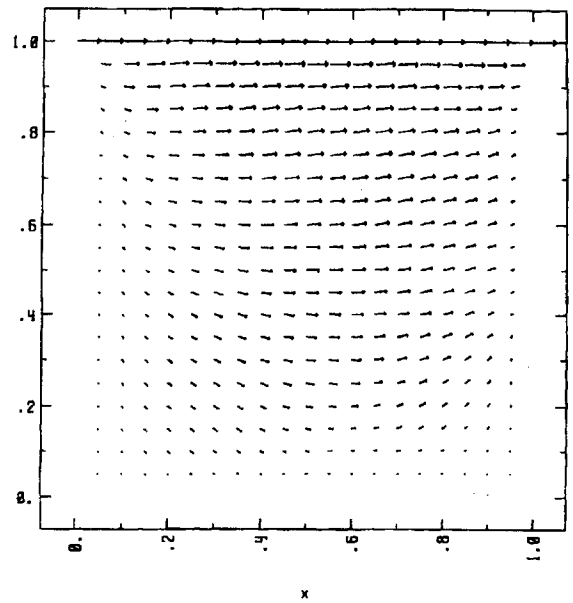

Figure 1(a). Steady vector field for a lid-driven cavity at $Re = 100$

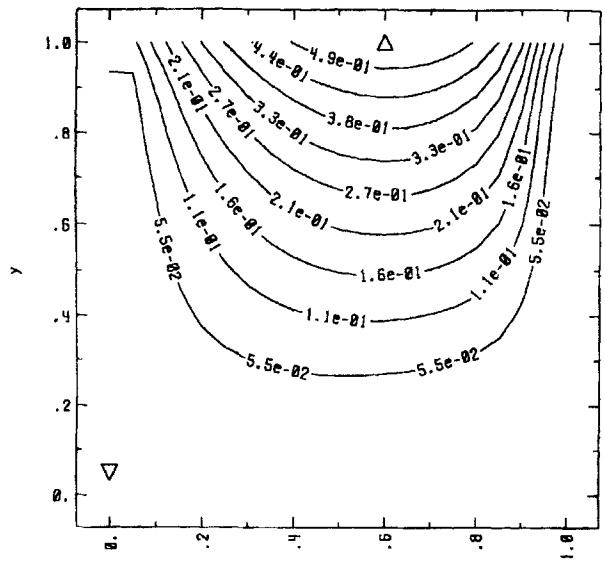

Figure 1(b). Streamfunction from Figure 1(a)

'CB filter' described in Reference 27—good pressures. (See Reference 28 for further defensive arguments, *including a new convergence proof.*)

5. It is necessary to integrate ∇P by parts in the weak formulation, a consequence of which is outflow boundary condition difficulties when body forces are present.[28,29]

Thus, to get back 'in vogue', we made an attempt, described below, at creating a *stabilized* equal-order element, namely Q_1Q_1, which overcomes at least the first two flaws. However, features 3 and 4 still 'shoot it down' *and* mass lumping still seems required. To implement a *consistent mass* Q_1Q_1 element *and* to pass 'LBB' (which Q_1Q_1 fails in spades), we invoke an 'approximate projection' in much the same way that many before us have done: we replace the 'bad' Laplacian in (11) by the 'good' (conventional GFEM) Laplacian (15); and, after reverting to mass lumping, we also devise an *explicit* PPE method somewhat along the lines followed in Reference 12. However, in both cases the 'replacement' is only part of the fix; more is needed, in the form of 'tricks', to obtain useful algorithms.

We shall now describe our two methods for stabilizing Q_1Q_1, beginning with the semi-implicit approximate projection method.

3. FIRST METHOD—SEMI-IMPLICIT APPROXIMATE PROJECTION

There are three major steps to the derivation (and execution) of this technique.

Step 1

Replace the weak form in (7) by

$$\int \mathbf{v} \cdot \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} + \nabla P \right) + \nu (\nabla \mathbf{u})^{\mathrm{T}} : \nabla \mathbf{v} = 0, \tag{20}$$

which, because ∇P is not integrated by parts, implies $\nu \partial \mathbf{u}/\partial n = 0$ as an OBC, which is generally and ill-posed boundary condition.[28,29] However, we get away with this 'variational crime' because we will no longer require a 'stringent' version (even discretely) of $\nabla \cdot \mathbf{u} = 0$.

Step 2

Generate the following index 2 DAEs from (20) and (2):

$$M\dot{u} + N(u)u + GP = Ku + f, \qquad u(0) = u_0, \tag{21}$$

$$Du = g, \qquad Du_0 = g_0, \tag{22}$$

where G and D are no longer transposes of each other; in fact,

$$G_{ij} \equiv \begin{bmatrix} \int \varphi_i \dfrac{\partial \psi_j}{\partial x} \\ \int \varphi_i \dfrac{\partial \psi_j}{\partial y} \end{bmatrix}$$

is the gradient operator and

$$D_{ij} \equiv \left[\int \psi_i \frac{\partial \varphi_j}{\partial x}, \quad \int \psi_i \frac{\partial \varphi_j}{\partial y} \right]$$

is the divergence operator and we note that $D = -C^{\mathrm{T}}$. It is true, however, up to the boundary at least (owing to the boundary integral when integrating by parts), that div–grad symmetry is respected, i.e. $D = -G^{\mathrm{T}}$ in the *interior.*

Step 3

Solve the DAEs via the following *approximate* projection method. Given u_n and P_n:

(i) Solve

$$M \frac{\tilde{u}_{n+1} - u_n}{\Delta t} + N(u_n)u_n + GP_n = \tfrac{1}{2}K(\tilde{u}_{n+1} + u_n) + f_n$$

for $\tilde{u}_{n+1}$, the intermediate velocity, i.e. solve

$$\left(\frac{1}{\Delta t}M - \tfrac{1}{2}K\right)\tilde{u}_{n+1} = \left(\frac{1}{\Delta t}M + \tfrac{1}{2}K\right)u_n + f_n - N(u_n)u_n - GP_n. \tag{23}$$

(ii) Project $\tilde{u}_{n+1}$ to the *approximately* discretely div-free subspace as follows.

(a) Solve

$$L(P_{n+1} - P_n) = -[D(\tilde{u}_{n+1} - u_n) - (g_{n+1} - g_n)]/\Delta t \tag{24}$$

for ΔP.

(b) Compute the projected velocity from

$$u_{n+1} = \tilde{u}_{n+1} - \Delta t M_{\mathrm{L}}^{-1} G(P_{n+1} - P_n). \tag{25}$$

(iii) Finally, update P and go to the next time step; i.e. go to (i).

The approximate projection algorithm in (23)–(25) is *derived* as follows. Given $\tilde{u}_{n+1}$, solve for $P_{n+1} - P_n$ and u_{n+1} from

$$M_L \frac{u_{n+1} - \tilde{u}_{n+1}}{\Delta t} + G(P_{n+1} - P_n) = 0, \tag{26}$$

$$Du_{n+1} = g_{n+1} + (Du_n - g_n), \tag{27}$$

which is called 'projecting the difference'—a required 'trick' obtained from J. Bell (LLNL, personal communication) and *perhaps* better motivated as follows: solve $D\dot{u} = \dot{g}$, with $Du_0 = g_0$, via the approximation $D(u_{n+1} - u_n)/\Delta t = (g_{n+1} - g_n)/\Delta t$. Note, however, that because of the next trick below, $Du = g$ is never quite achieved in this method, thus explaining the name 'approximate' projection. Equations (26) and (27) imply the following 'PPE' for the pressure update:

$$(DM_L^{-1}G)(P_{n+1} - P_n) = [D(\tilde{u}_{n+1} - u_n) - (g_{n+1} - g_n)]/\Delta t, \tag{28}$$

which is replaced by (24) ($DM_L^{-1}G$ in (28) is replaced by $-L$) and is the final 'trick' referred to above—a replacement made necessary because $DM_L^{-1}G$ has too many spurious pressure modes (fails LBB; it also describes a very 'broad' stencil, coupling 25 nodes in 2D and 125 in 3D!) and made 'interesting' because iterative solvers should be better behaved with L (i.e. require fewer iterations), or so we thought (hoped). The L-matrix is also much more 'compact': it couples only nine nodes in 2D and 27 in 3D.

Remarks

1. The use of $(P_{n+1} - P_n)/2$ rather than $P_{n+1} - P_n$ also 'works'[26,30] and may even seem more consistent, but it is less robust in that the pressure sometimes displays $2\Delta t$ oscillations.
2. If $g_n - Du_n$ is omitted from the RHS of (24), the results (which we have also seen experimentally) are disappointing in that (at least) a steady pressure cannot be attained even when the velocity becomes steady—it changes linearly in time. This is explained as follows ($\tilde{u}_{n+1} = u_n = u_{n+1}$ at steady state).
   (i) Equation (24) becomes $\Delta t L(P_{n+1} - P_n) = g_s - Du_s \equiv e_s \neq 0$ because u is always only *approximately* div-free; there is then a constant increment to the pressure field at each step even though the velocity is steady. (With $g_n - Du_n$ present, $e_s = 0$ and thus $P_{n+1} - P_n = P_s$.)
   (ii) There is only one way *possible* to obtain a steady velocity and a linearly changing pressure: P_n in (24) must contain a steady part and some *pressure modes* (one or more) from the null space of G. Thus from (24) (sans $Du_n - g_n$) applied at steady state we must have $P_{n+1} = P_s + [(n+1)/\Delta t]L^{-1}e_s$, where P_s is the steady pressure corresponding to u_s and $L^{-1}e_s = P_m$ is some linear combination of the vectors in the null space of G; the pressure 'grows' linearly with time because $e_s = LP_m$ excites (in general) one or more pressure modes.
3. BTD (balancing tensor diffusivity) is employed (K is modified to compensate for the problems associated with the explicit Euler treatment of advection; see Reference 31 for details).

Related to the pressure mode remark is the following: we have implemented a start-up procedure in which we can *iterate* on the approximate projection of an 'arbitrary' IC vector u_0 to try to bring its discrete divergence closer to zero.

(i) Given u_0 with $Du_0 \neq g_0 = 0$ (for simplicity).
(ii) Project u_0 *towards* the div-free subspace via $u_1 = \mathscr{P}_a u_0$, where

$$\mathscr{P}_a \equiv I + M_L^{-1}GL^{-1}D \tag{29}$$

is the 'approximate' projection matrix (i.e. the exact matrix of the approximate projection) implied by (24) and (25). Note that $\mathscr{P}_a^2 \neq \mathscr{P}_a$, and the sign change.

(iii) Since $Du_1 = D\mathscr{P}_a u_0 = (D + DM_L^{-1}GL^{-1}D)u_0 \neq 0$ but is hopefully small (especially, smaller than Du_0!), one may try *repeated/iterated projections*.

(iv) $u_m = \mathscr{P}_a^m u_0$, $m = 1, 2, \ldots$, which leads to a consideration of the spectrum of $\mathscr{P}_a$ and its 'relative', the *error* matrix

$$E \equiv I + DM_L^{-1}GL^{-1}. \tag{30}$$

so-called because $E = 0$ if $DM_L^{-1}G = -L$; we hope that $-DM_L^{-1}GL^{-1} \approx I$. They are 'related' because (a) they have the same spectrum and (b) $Du_m = E^m Du_0$ is the 'iterated' divergence since $D\mathscr{P}_a^m = E^m D$. We shall *assume* that $\| E \|$ is 'small', $O(h)$ or better, in the sense that $Eq = O(h)$ for a sufficiently smooth vector q. (The 'iterates' are of course obtained as follows: solve $Lq_m = -Du_{m-1}$ for q_m and compute $u_m = u_{m-1} - M_L^{-1}Gq_m$.)

(v) After *assuming* that $\mathscr{P}_a$ has a complete set of eigenvectors, we have: since $\mathscr{P}_a^m u_0 = \sum_{j=1}^N a_j \lambda_j^m x_j$ and $Du_m = \sum_{j=1}^N a_j \lambda_j^m z_j$, where (λ_j, x_j) are the eigenvalues and eigenvectors of $\mathscr{P}_a$ and $(\lambda_j, z_j = Dx_j)$ are those of E and where $u_0 = \sum_{j=1}^N a_j x_j$ and $Du_0 = \sum_{j=1}^N a_j z_j$, it follows that u_m (and Du_m) will tend to (1) zero, (2) infinity or (3) a constant, according to whether

(1) all λs are less than unity in magnitude
(2) one or more λs are larger than unity in magnitude
(3) some are less than unity and one or more are equal to unity in magnitude.

(iv) We have found experimentally that case (3) is observed—iterating on the projection reduces the div, but only up to a point (asymptote), not to zero; also, $\mathscr{P}_a^m u_0$ approaches a constant. The explanation is the following: u_m approaches a constant implies that (a) $\mathscr{P}_a$ is norm-reducing ('stable'; $u = \mathscr{P}_a u_0 \Rightarrow \| u \| \leq \| u_0 \|$) and (b) $\mathscr{P}_a$ has some eigenvectors with $\lambda = 1$. That (b) is true is actually quite obvious, since, like the true projection matrix discussed earlier, any discretely div-free vector ($Dx = 0$) is such an eigenvector. A heuristic argument that (a) is true follows from a Rayleigh quotient argument and utilizing $D = -G^T$ (up to boundaries): $\mathscr{P}_a x_j = \lambda_j x_j \Rightarrow (\lambda_j - 1)x_j = M_L^{-1}GL^{-1}Dx_j \Rightarrow (\lambda_j - 1)Dx_j = -G^T M_L^{-1}GL^{-1}Dx_j \Rightarrow 1 - \lambda_j = (Gy_j)^T M_L^{-1}(Gy_j)/y_j^T L y_j \geq 0$, where $y_j \equiv L^{-1}Dx_j = L_j^{-1}z_j$ since L is SPD; thus $\lambda_j \leq 1$, with $\lambda_j = 1$ corresponding either to $Dx_j = 0$ or to a pure pressure mode of G, i.e. $Gy_j = 0$. Since this analysis is only rigorous when there are no open boundaries, yet we observe the stated results for all types of BCs, it must be the case that boundary differences are unimportant.

(vii) Similarly, Du_m approaches a constant implies that E is norm-reducing and has at least *one* eigenvector (with $\lambda = 1$) that satisfies $DM_L^{-1}G^{-1}z = 0$, which vector is *also* easy to find; just take $L^{-1}z = P_m$, any null vector of G (which often has quite a few[28]).

The 'bottom line' here is this: if Du_0 has any projection onto the null space of G, the iterated projection will tend to a constant with a *non-zero* (but 'small') constant div—and this appears to be the 'general' case.

Now comes the 'hard part'—analysing the resulting algorithm to show that it is actually *useful*. We admit up front that we have not yet been totally successful. However, being what Strang and Fix[32] call 'mathematical engineers', we tested it in the computational laboratory anyway. (It works. This happens often with engineers, but definitely not always.) If we study (23)–(25) for $\Delta t \to 0$, we get, with $F_n \equiv f_n + Ku_n - N(u_n)u_n - GP_n$,

[1] $\tilde{u}_{n+1} = u_n + \Delta t M^{-1}F_n + O(\Delta t^2)$
[2] $L(P_{n+1} - P_n) = -DM^{-1}F_n + \dot{g}_n + O(\Delta t)$

[3] $u_{n+1} = \tilde{u}_{n+1} + \Delta t M_L^{-1} G L^{-1}(DM^{-1}F_n - \dot{g}_n) + O(\Delta t^2)$
$= u_n + \Delta t(\mathscr{P}_a M^{-1} F_n - M_L^{-1} G L^{-1} \dot{g}_n) + O(\Delta t^2)$
[4] $Du_{n+1} = Du_n - \Delta t[EL(P_{n+1} - P_n) - \dot{g}_n] + O(\Delta t^2)$
[5] thus $\Delta t \to 0 \Rightarrow \dot{u} = \mathscr{P}_a M^{-1}[f + Ku - N(u)u - GP] - M_L^{-1} G L^{-1} \dot{g}$ and thus
[6] $D\dot{u} = D\mathscr{P}_a M^{-1} F - DM_L^{-1} G L^{-1} \dot{g} = EDM^{-1}[f + Ku - N(u)u - GP] - (DM_L^{-1} G) L^{-1} \dot{g}$

and the *problem* is to *reconcile* [4] with [6]; the former implies $D\dot{u} = -EL[\lim_{\Delta t \to 0}(P_{n+1} - P_n)] + \dot{g}$ and the two are only in accord if

$$E\{(DM^{-1}G)P - DM^{-1}[f + Ku - N(u)u] + \dot{g}\} = EL \lim_{\Delta t \to 0}(P_{n+1} - P_n), \tag{31}$$

which, with $\lim_{\Delta t \to 0}(P_{n+1} - P_n) = 0$ and without the E-factor, is *just* the PPE that we would *like* the pressure to obey—corresponding as it does to an exact projection! Also, if this is true, then [5] above yields

$$\dot{u} = M^{-1}[f + Ku - GP - N(u)u], \tag{32}$$

also the 'desired' result, i.e. if (31) and (32) were truly satisfied (in the limit) by our approximate projection method, we would have a discretely div-free velocity and presumably a better solution—at least up to the spurious pressure modes of G. (Note the mysterious disappearance of the L-matrix.) However, we have not been able to show that $P_{n+1} - P_n = O(\Delta t)$. In fact, a sort of 'converse' seems to be true; namely, starting from the initial pressure P_0 obtained from

$$LP_0 = -DM_L^{-1}[f_0 + Ku_0 - N(u_0)u_0] + \dot{g}_0, \tag{33}$$

we find, at the end of the first step, $L(P_1 - P_0) = ELP_0 + O(\Delta t) + O(H)$ and $Du_1 = g_1 + \Delta t E^2 LP_0 + O(\Delta t^2) + O(\Delta t H)$, where $H \equiv M_L^{-1} M - I$ is a 'small' ($O(h^k)$, where $k = 1$ or 2) matrix in the sense that $Hu = O(h^k)$ for a sufficiently smooth vector u. These are not encouraging results—even with mass lumping ($H = 0$).

However, the resulting code seems to perform much better than these gloomy results would indicate—as we shall show.

4. SECOND METHOD—EXPLICIT EULER

The *other* method used to stabilize $Q_1 Q_1$ is simple explicit time integration. As with the 'modified' projection method described above, some 'tricks' are needed to come up with a method that 'works'. Also, like the projection method, the resulting algorithm seems to defy rigorous analysis that proves convergence—but we will present what we *do* have.

We begin 'conceptually' by applying the forward Euler method to (21) to get

$$M(u_{n+1} - u_n)/\Delta t + N(u_n)u_n + GP_n = Ku_n + f_n, \tag{34}$$

where the pressure is first computed from the PPE, derived from (21) and (22),

$$(DM^{-1}G)P_n = DM^{-1}[f_n + Ku_n - N(u_n)u_n] - (g_{n+1} - g_n)/\Delta t, \tag{35}$$

a simple algorithm that merits the following remarks.

Remarks

1. It is only viable if mass lumping is invoked—a degradation we would like to avoid.
2. The 'consistent' pressure matrix $DM^{-1}G$ is plagued with multiple spurious pressure modes and associated solvability constraints.[28]

We must live with mass lumping, but we did succeed in stabilizing the above 'LBB-unstable' method by the *combination* of (i) replacing $DM^{-1}G$ by $-L$ (which, as shown earlier, is not in itself viable), (ii) replacing g_n on the RHS of (35) by Du_n and (iii) replacing D by $-\tilde{D}$, where

$$\tilde{D}_{ij} \equiv \left[\int \frac{\partial \psi_i}{\partial x} \varphi_j, \quad \int \frac{\partial \psi_i}{\partial y} \varphi_j \right]$$

à la (8), and we remark that $\tilde{D}$ corresponds to (-div) except at boundaries (owing to the integration by parts in (8)). Also true is that $\tilde{D} = B + C^{\mathrm{T}} = G^{\mathrm{T}}$ and $D + \tilde{D} = B$, where B is the boundary matrix,

$$B_{ij} \equiv \left[n_x \int_\Gamma \psi_i \varphi_j, \quad n_y \int_\Gamma \psi_i \varphi_j \right].$$

Thus (35) is replaced by

$$LP_n = \tilde{D} M_{\mathrm{L}}^{-1} [f_n + Ku_n - N(u_n)u_n] + (g_{n+1} - Du_n)/\Delta t \tag{36}$$

and (34) by

$$u_{n+1} = u_n + \Delta t M_{\mathrm{L}}^{-1} [f_n + Ku_n - N(u_n)u_n - GP_n]. \tag{37}$$

This is the explicit algorithm that works—except that the last term on the RHS of (36) is omitted at $t = 0$—on the assumption that we either start up with a div-free velocity or we perform multiple (iterated) approximate projections so that $Du_0 \approx g_0$. That a portion of the resulting 'pressure' might be interpretable as a Lagrange multiplier that penalizes divergence error can be easily seen by considering (36): unless $Du_n - g_n = O(\Delta t)$ or less, the $\Delta t \to 0$ result is

$$\Delta t LP_n \to g_{n+1} - Du_n = g_n - Du_n + O(\Delta t), \tag{38}$$

i.e. $\Delta t P_n \equiv \lambda_n$ is finite even for $\Delta t \to 0$ if $Du_n \neq g_n$. This 'part' of P_n can clearly *not* be a pressure—which calls to mind the statement in Reference 33, 'Not all quantities called P are equal'. Our version is 'Not all quantities called P are pressure'.

As with the approximate projection method, our analysis of the algorithm is incomplete, but we present now what little we do have. We begin by applying the divergence operator to u_{n+1} from (37):

$$(DM_{\mathrm{L}}^{-1}G)P_n = DM_{\mathrm{L}}^{-1}[f_n + Ku_n - N(u_n)u_n] - D(u_{n+1} - u_n)/\Delta t, \tag{39}$$

which, when combined with (36), yields, at least away from boundaries (whence $\tilde{D} = -D$) and for $n \geq 1$,

$$Du_{n+1} - g_{n+1} = -\Delta t(L + DM_{\mathrm{L}}^{-1}G)P_n = -\Delta t ELP_n, \tag{40}$$

showing that, if $P_n = O(1)$ in Δt (which may not be true for all n), the error in the discrete divergence is $O(h\Delta t)$. (Clearly, however, if we started div-free, then P_0 is surely independent of Δt and thus $Du_1 - g_1 = O(\Delta t)$.)

Next we eliminate Du_n from (36) via (40) with the index dropped by one to obtain (for $n \geq 2$)

$$-(DM_{\mathrm{L}}^{-1}G)P_{n-1} = \tilde{D} M_{\mathrm{L}}^{-1} [f_n + Ku_n - N(u_n)u_n] + (g_{n+1} - g_n)/\Delta t - L(P_n - P_{n-1}), \tag{41}$$

which can be rewritten as (add $-DM_{\mathrm{L}}^{-1}G(P_n - P_{n-1})$ to both sides)

$$-(DM_{\mathrm{L}}^{-1}G)P_n = \tilde{D} M_{\mathrm{L}}^{-1} [f_n + Ku_n - N(u_n)u_n] + (g_{n+1} - g_n)/\Delta t - EL(P_n - P_{n-1}). \tag{42}$$

Now for another *big* if: *if* $P_n - P_{n-1} = O(\Delta t)$, (42) will clearly converge to the consistent Q_1Q_1 PPE

$$(DM_{\mathrm{L}}^{-1}G)P = -\tilde{D} M_{\mathrm{L}}^{-1} [f + Ku - N(u)u] - \dot{g}, \tag{43}$$

which is the lumpted mass version of (31) without the E-factor on the RHS after using $-D=\tilde{D}$. Again there seems to be a 'mysterious disappearance' of the L-matrix. Also, if $h \to 0$ at fixed Δt, the L-matrix term vanishes (because E on any vector does). If in fact (43) is true, it follows from (37) for $\Delta t \to 0$ that

$$D\dot{u} = DM_{\mathrm{L}}^{-1}[f + Ku - N(u)u] - DM_{\mathrm{L}}^{-1}GP = \dot{g}, \tag{44}$$

another miracle—mass conservation if $Du_0 = g_0$. Thus, just as with our semi-implicit algorithm, we *need* $P_{n+1} - P_n = O(\Delta t)$ to realize optimal (up to spurious modes) behaviour.

Another analysis, with $g=0$ for simplicity, goes as follows: given u_0 with $Du_0 \neq 0$ but 'small' (with or without iterations), apply inductive analysis starting at $n=0$ and again neglecting the 'boundary term' by assuming $D+\tilde{D}=0$ to obtain

$$u_n = \mathscr{P}_{\mathrm{a}}^{n-1}u_0 + \Delta t \sum_{i=0}^{n-1} \mathscr{P}_{\mathrm{a}}^{n-i} M_{\mathrm{L}}^{-1}[f_i + Ku_i - N(u_i)u_i], \tag{45}$$

$$Du_n = E^{n-1}Du_0 + \Delta t \sum_{i=0}^{n-1} E^{n-i} DM_{\mathrm{L}}^{-1}[f_i + Ku_i - N(u_i)u_i], \tag{46}$$

$$\begin{aligned} L(P_{n+1} - P_n) &= -DM_{\mathrm{L}}^{-1}[f_{n+1} - f_n + K(u_{n+1} - u_n) - N(u_{n+1})u_{n+1} + N(u_n)u_n] \\ &\quad - EDM_{\mathrm{L}}^{-1}[f_n + Ku_n - N(u_n)u_n] - DM_{\mathrm{L}}^{-1}GL^{-1}Du_n/\Delta t \\ &= O(\Delta t) - EDM_{\mathrm{L}}^{-1}[f_n + Ku_n - N(u_n)u_n] - (DM_{\mathrm{L}}^{-1}G)L^{-1}Du_n/\Delta t. \end{aligned} \tag{47}$$

From these results and those presented earlier, we conclude the following (assuming that the neglected boundary terms do not alter the results).

1. If Du_0 is 'small', less than or equal to $O(\Delta t)$, so too will be Du_n and $P_{n+1} - P_n$; the simulation may be good for all $t \geq 0$.
2. If Du_0 is 'large', (46) shows that Du_n will still be reduced at each step, tending towards a constant part, an $O(\Delta t)$ part and an $O(h)$ part caused by the E-term. Also, (47) shows that $P_{n+1} - P_n$ will not be $O(\Delta t)$, at least not initially; it is $O(h) + O(1/\Delta t)$. It may eventually get there or it may not. In this case the early portion of the simulation will generally be devoid of physical meaning—as indeed might the whole thing.
3. It is thus generally advisable to iterate on the initial velocity to try to drive the div down to an acceptable level before beginning the simulation. Better still is to start with $Du_0 = g_0$.

Again the computer code seems to be smarter than its designers in that the algorithm delivers better results than it 'has any right to'.

5. NUMERICAL RESULTS

The first thing we test is the ability of the approximate projection to 'preserve the div'—here with the semi-implicit version. Recall that we noted below (13) that methods based on solving the PPE will maintain (pointwise) the initial velocity divergence, a fact demonstrated in Reference 24. A portion of those experiments is repeated here. At $t=0$ we set $u=3$ and $v=2$ at *one* node of a presumably incompressible fluid in the unit box; all other nodes had $\mathbf{u}_0 = \mathbf{0}$. Figure 2 shows the steady solution at $Re=10$. While not a solution of the Navier–Stokes equations, the result does help 'grade' the quality of the approximation projection: $\| \nabla \cdot \mathbf{u}_\infty \|_{\mathrm{RMS}} \approx 0{\cdot}6 \| \nabla \cdot \mathbf{u}_0 \|_{\mathrm{RMS}}$, which is not bad. In another experiment we iterated the projection on the single vector initial condition above (25 iterations) to reduce the initial spurious div, as discussed above. The resulting time integration was successful in that

a meaningful simulation (spin-down) did occur, with results that looked much like those in Figure 16 of Reference 24. Finally, application of the *explicit* Q_1Q_1 approximate projection to the given initial vector *also* resulted in a semi-meaningful spin-down (slowest Stokes decay mode, ultimately), clearly demonstrating that the 'penalty' term on the RHS of (36) is also acting like an 'iterated projection' *during* the time integration; more obfuscation.

However, we have successfully (for the most part) compared Q_1Q_1 against our workhorse Q_1P_0 for lid-driven cavities and vortex shedding past circular cylinders. Here we show a sample of results for flow past an aerofoil (waterfoil, actually) at $Re=10^4$ (based on chord c) at a small (1·2°) angle of attack. The particular aerofoil, 'a NACA 16 thickness form with maximum thickness $t_0/c = 8{\cdot}84\%$ and maximum camber $f_0/c = 2{\cdot}576$ per cent with a bevelled anti-singing trailing edge',[34] was tested in a *water* tunnel (*incompressible* flow!) at MIT's Ocean Engineering Department as part of a Navy/ARPA programme on unsteady fluid dynamics. However, our laminar flow simulations were not meant to be compared with their measured data at $Re > 10^6$ because we do not yet have a believable turbulence model.

We designed a 'truth' mesh of about 40,000 elements and a test mesh of about 6700 elements. The fine mesh was run at $Re=10^3$, 10^4, 10^5 and 10^6 using Q_1P_0 and our 'projection 2' algorithm (an 'exact' projection method).[26] The results were not believable at $Re=10^6$ (chaotic), semi-believable at $Re=10^5$ (nearly periodic) and believable at $Re \leqslant 10^4$. $Re=10^3$ (only) produced a steady state result. The coarse mesh for $Re=10^3$ was then run with Q_1Q_1; all three results were acceptably close to each other.

We now focus on the $Re=10^4$ case and compare Q_1P_0 on the two meshes with Q_1Q_1 on the coarse mesh. Qualitatively the flow is one of periodic vortex formation and shedding of vortices at the trailing edge. Nowhere else is there any interesting dynamics. Figure 3 shows snapshots of vectors, streamfunction and vorticity from Q_1P_0 on the coarse mesh (vectors are interpolated via a coarser-yet mesh graphics package). Figure 4 shows time histories of the x-component of velocity at a node just above the trailing edge (about $0{\cdot}002c$); it is #8101 on the fine mesh and #1361 on the coarse mesh. The agreement of semi-implicit Q_1Q_1 with Q_1P_0 on the coarse mesh is quite close, the range of us during

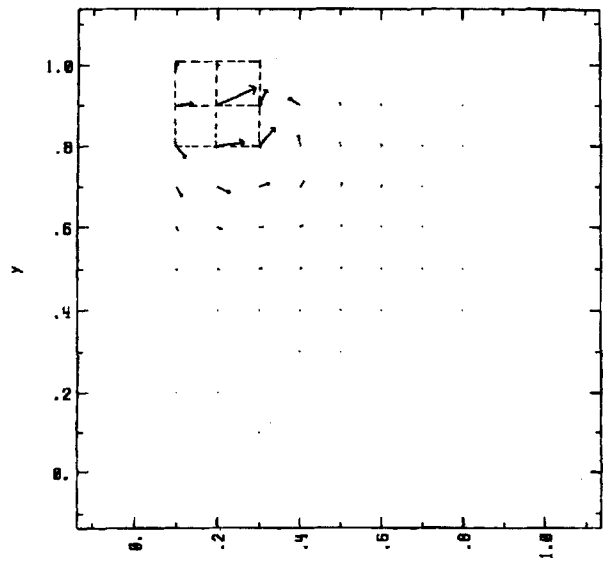

Figure 2. Steady state solution for testing how well the initial divergence is preserved

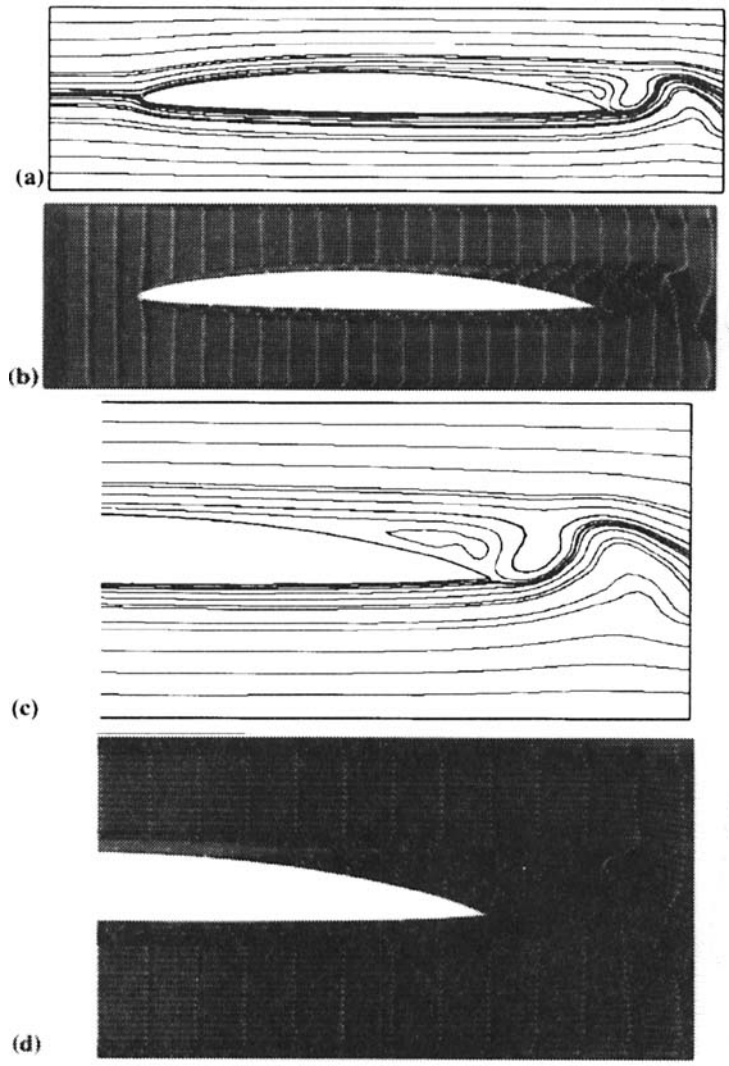

Figure 3. MIT waterfoil—snapshots of results at $Re = 10^4$: (a) streamfunction; (b) vectors and vorticity; (c) and (d) are close-ups of (a) and (b) respectively

vortex shedding being from about $-0{\cdot}07$ to $0{\cdot}16$ for Q_1P_0 and from about $-0{\cdot}08$ to $0{\cdot}15$ for semi-implicit Q_1Q_1. (The fine mesh results ranged from about $-0{\cdot}10$ to $0{\cdot}20$, somewhat stronger, but still reasonably close, while the explicit Q_1Q_1 results (not shown) ranged from about $-0{\cdot}17$ to $0{\cdot}35$, stronger yet.) The pressure histories, shown in Figure 5, are of similar quality; the coarse mesh results ranged from about $-0{\cdot}17$ to $-0{\cdot}31$ for Q_1P_0, from about $-0{\cdot}16$ to $-0{\cdot}27$ for semi-implicit Q_1Q_1 and from about $-0{\cdot}17$ to $-0{\cdot}41$ for explicit Q_1Q_1 (not shown) compared with the 'true' results from about $-0{\cdot}15$ to $-0{\cdot}33$. Note that we selected a particular challenging point in the flow field in order to *emphasize* differences. (In all cases the inlet velocity is $1{\cdot}0$, as is the velocity on the top and bottom boundaries—tow tank BCs). The OBCs were 'natural', i.e. $\nu\partial u/\partial x - P = 0 = \nu\partial v/\partial x$ for Q_1P_0 and $\nu\partial u/\partial x = \nu\partial v/\partial y = P = 0$ for Q_1Q_1, the latter being imposed as an *essential* BC in the PPE.) Finally, Figure 6 presents, for Q_1Q_1 and the approximate projection, the RMS norm of $\nabla \cdot \mathbf{u}$—showing a typical tolerable result.

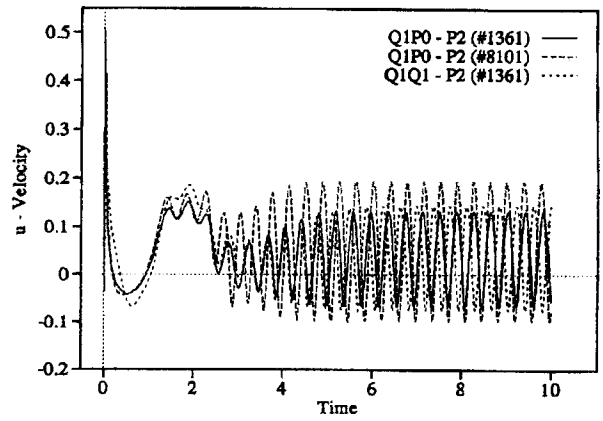

Figure 4. Horizontal velocity near trailing edge from three simulations—all 'projection 2'

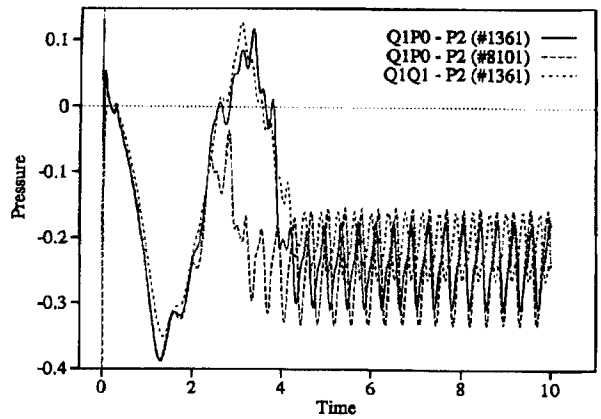

Figure 5. Pressures at same locations as in Figure 4

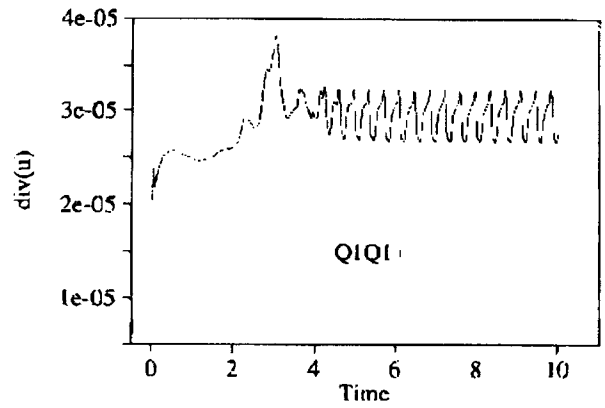

Figure 6. RMS norm of divergence

6. DISCUSSION

One of the numerical experiments not yet discussed relates to the 'impulsive start' (potential flow and no-slip BC; see e.g. Reference 21) of flow past a 'geometric singularity'—a convex boundary with a sharp (discontinuous) change in slope. This experiment helped to open our eyes to a 'down-side' of projection methods, both approximate (Q_1Q_1) *and* exact (Q_1P_0): projection methods are inherently *time-accurate* methods—a statement that tends to make them seem more like explicit methods than the implicit ones that they (usually) allege to be. This remark and those to follow do not apply to the stability-limited explicit method discussed above.

It turns out that 'large'-Δt simulation via Q_1Q_1 and the approximate projection advertises its 'time-accurate' syndrome rather sooner than does Q_1P_0 and the exact projection; namely, if Δt is not small enough, the div-error becomes large enough to be noticeable. However, Q_1P_0 is always exactly div-free (discretely), regardless of step size, and this turns out to be the more 'interesting' case. We go right to the 'limiting and simple case' to state our major (negative) result: if implicit Euler is used to time march the transient Stokes equations and Δt is allowed to become arbitrarily large, not only is all the physics lost, but—rather than obtaining the steady solution in a *single* time step, one of the well-known characteristics of backward Euler applied to ODEs—the 'integration' requires an infinite number of steps! Yes—the *larger* the step size, the *more* of them are required to find the steady solution and the 'transient' is of course meaningless. In the limit the true steady state is not even *attainable*; rather, the algorithm has *changed the problem* by changing the body force—and the pressure *never* changes from its *initial* value. These assertions, which may raise some eyebrows, are actually quite simple to prove, both for the continuum equations and for their spatially discretized approximations; we choose the latter.

We present the analysis in the simplest way and with 'condensed' terminology, i.e.

$$\dot{u} + Ku + GP = f \tag{48}$$

and

$$Du = g, \tag{49}$$

where $D = G^{\mathrm{T}}$, are the transient semidiscretized Stokes equations and f and g are independent of time (and we have changed the sign of K). The backward Euler (exact) projection algorithm is then as follows. Given P_n and u_n with $Du_n = g$ for $n = 0, 1, \ldots$:

1. Solve

$$(\tilde{u}_{n+1} - u_n)/\Delta t + K\tilde{u}_{n+1} + GP_n = f \tag{50}$$

   for $\tilde{u}_{n+1}$.
2. Solve

$$DG(P_{n+1} - P_n) = (D\tilde{u}_{n+1} - g)/\Delta t \tag{51}$$

   for $(P_{n+1} - P_n)\Delta t$.
3. Compute

$$u_{n+1} = \tilde{u}_{n+1} - G(P_{n+1} - P_n)\Delta t. \tag{52}$$

4. Bump n and go to step 1. (Note that $Du_{n+1} = g$, as desired.)

Now consider what happens if the first step is a big one. Take $n = 0$ and $\Delta t = \infty$ to obtain:

1.

$$\tilde{u}_1 = K^{-1}(f - GP_0), \tag{53}$$

where P_0 came from

$$(DG)P_0 = D(f - Ku_0). \tag{54}$$

2. Solve for $\lambda \equiv \Delta t(P_1 - P_0)$ from

$$DG\lambda = D\tilde{u}_1 - g, \tag{55}$$

where it is important to note that $\Delta t\Delta P$ is perfectly well-defined, from (51), even for $\Delta t \to \infty$, i.e. λ (the Lagrange multiplier) is finite even though $P_1 = P_0$.

3. Compute

$$\begin{aligned} u_1 &= \tilde{u}_1 - G\lambda \\ &= [I - G(DG)^{-1}D]\tilde{u}_1 + G(DG)^{-1}g \\ &= \mathscr{P}\tilde{u}_1 + G(DG)^{-1}g \\ &= \mathscr{P}K^{-1}(f - GP_0) + G(DG)^{-1}g. \end{aligned} \tag{56}$$

To show that we are 'finished', consider taking a *second* big step:

$$\tilde{u}_2 = K^{-1}(f - GP_1) = K^{-1}(f - GP_0), \tag{57}$$

so that $\tilde{u}_2 = \tilde{u}_1$ and thus $u_2 = u_1 (= u_3 = \ldots \equiv u)$. What has happened is this: we have converted what *was*, for Δt sufficiently small, a legitimate approximation method for solving the transient Stokes equations to a useless mathematical algorithm that is stable but irrelevant.

That we indeed have obtained a *spurious* solution to the steady Stokes equations follows by realizing that in order for u and P_0 to solve the steady version of (48) and (49), it is necessary to change the *data*, i.e. the body force for the steady Stokes solution is no longer f, but is instead, from (48),

$$\tilde{f} = Ku + GP_0 \tag{58}$$

$$= K\mathscr{P}K^{-1}(f - GP_0) + GP_0 + KG(DG)^{-1}g \tag{59}$$

$$= (I - K\mathscr{P}K^{-1})GP_0 + K\mathscr{P}K^{-1}f + KG(DG)^{-1}g, \tag{60}$$

which has introduced another 'obscure' projection matrix, $K\mathscr{P}K^{-1}$—for what it is worth.

This bizarre behaviour is depicted schematically in Figure 7, in which the horizontal line represents the manifold of all div-free velocities ($g = 0$ for convenience and to obtain orthogonal projections). Any velocity not on this manifold has a non-zero divergence and any solution of the Stokes (or Navier–Stokes) equations must lie on this line. For example, if u_0 is the IC and u_s is the corresponding steady state Stokes solution, then the solution for $0 < t < \infty$ moves along the line from (say) left to right. The projection method, on the other hand, first moves the velocity *off* this line ($\tilde{u}$) and then projects it back down to it—once per time step. If Δt is sufficiently small (say Δt_1 in the figure), the resulting projected velocities will be close to 'correct.' The larger Δt becomes, however, the further is $\tilde{u}$ taken from the div-free manifold and the less likely is the projection method to be useful. The $\Delta t \to \infty$ situation is depicted by $\tilde{u}_\infty$ and a purely 'vertical' oscillation between $\tilde{u}_\infty$ and u.

The analogous schematic behaviour of the trapezoid rule integrator is shown in Figure 8 for Δt_1, $\Delta t_2 > \Delta t_1$ and $\Delta t = \infty$, the analysis of which we leave to the reader, as we have already digressed too far away from fluid mechanics and into interesting but not useful mathematics.

7. CONCLUSIONS

We have introduced and demonstrated two ways to stabilize the Q_1Q_1 element using approximate projections. While we have been more successful at coding than analysis, we are also somewhat undermotivated to push on, because the new techniques seem to introduce more problems/questions

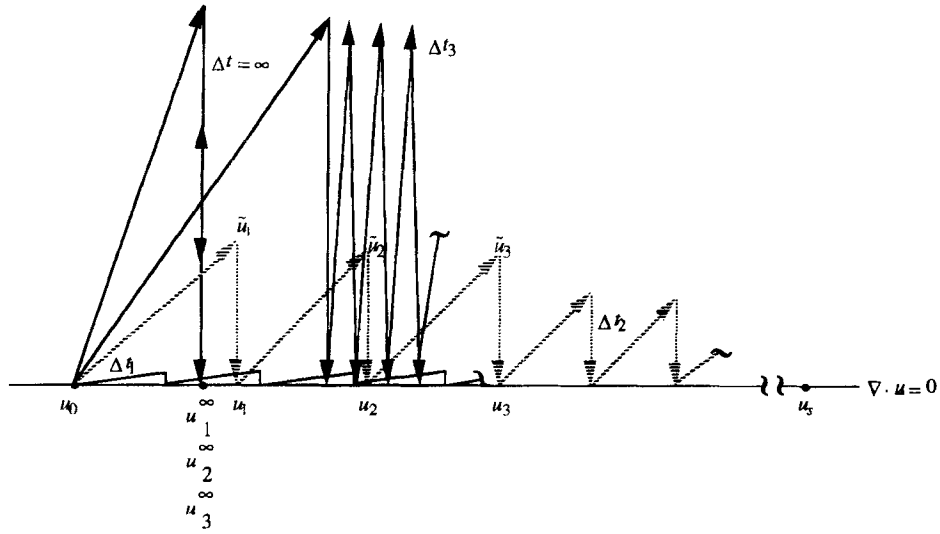

Figure 7. Pictorial of backward Euler projection method

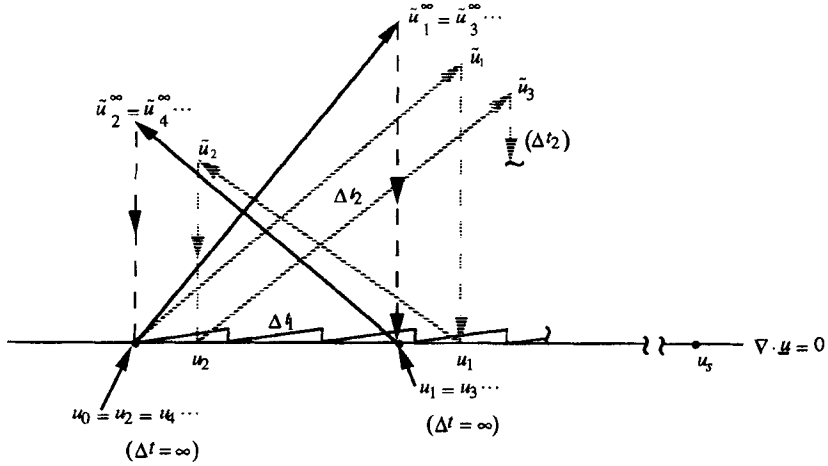

Figure 8. As in Figure 7 but using trapezoid rule

than solutions/answers. They also insidiously reintroduce the spurious null space of the gradient matrix. Finally, the code yielded much less speed-up, say 15 per cent or so, when solving the PPE via DSCG (diagonally scaled conjugate gradient) than we had hoped for.

We have also demonstrated, albeit perhaps not for the first time (see Reference 35) but surely quite clearly, that projection methods of the pressure correction type are *inherently* time-accurate methods that should not be employed with any but sufficiently small time steps so that the time integration is accurate, even if a steady state is finally attained, i.e. the projection method is not a good steady state seeker.

ACKNOWLEDGEMENTS

Discussions with J. Bell and L. Howell (LLNL) have again proven fruitful. This work was performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under contract W-7405-Eng-48.

REFERENCES

1. P. M. Gresho, R. Lee and R. Sani, 'Further studies on equal-order interpolation for Navier–Stokes', *Proc. 5th Int. Symp. on Finite Elements in Flow Problems*, Austin, TX, January 84, pp. 143–148.
2. P. M. Gresho and Leone, 'Another attempt to overcome the bent element blues', *Proc. 5th Int. Conf. on FEM in Water Resources*, Burlington, VT, June 1984, Springer, New York, 1984, pp. 667–683.
3. G. E. Schneider, G. D. Raithby and M. M. Yovanovich, 'Finite-element solution procedures for solving the incompressible, Navier–Stokes equations using equal order variable interpolation', *Numer. Heat Transfer*, **1**, 443–451 (1978).
4. M. Kawahara and K. Ohmiya, 'Finite element analysis of density flow using the velocity correction method', *Int. j. numer. methods fluids*, **5**, 981–993 (1985).
5. J. G. Rice and R. J. Schnipke, 'An equal-order velocity–pressure formulation that does not exhibit spurious pressure modes', *Comput. Methods Appl. Mech. Eng.*, **58**, 135–149 (1986).
6. T. J. R. Hughes and L. P. Franca, 'A new finite element formulation for computational fluid dynamics: VII. The Stokes problem with various well-posed boundary conditions: symmetric formulations that converge for all velocity/pressure spaces', *Comput. Methods Appl. Mech. Eng.*, **65**, 85–96 (1987).
7. D. W. Pepper and A. P. Singer, 'Calculation of convective flow on the personal computer using a modified finite-element method', *Numer. Heat Transfer A*, **17**, 379–400 (1990).
8. J. L. Sohn and J. C. Heinrich, 'A Poisson equation formulation for pressure calculations in penalty finite element models for viscous incompressible flows', *Int. j. numer. methods eng.*, **30**, 349– 361 (1990).
9. C. T. Shaw, 'Using a segregated finite element scheme to solve the incompressible Navier–Stokes equations', *Int. j. numer. methods fluids*, **12**, 81–92 (1991).
10. C. T. Shaw, 'Adding the time-dependent terms to a segregated finite element solution of the incompressible Navier–Stokes equations', *Eng. Comput.*, **8**, 305–316 (1991).
11. O. C. Zienkiewicz and J. Wu, 'Incompressibility without tears—how to avoid restrictions of mixed formulation', *Int. j. numer. methods eng.*, **32**, 1189–1203 (1991).
12. A. Kovacs and M. Kawahara, 'A finite element scheme based on the velocity correction method for the solution of the time-dependent incompressible Navier–Stokes equations', *Int. j. numer. methods fluids*, **13**, 403–423 (1991).
13. O. C. Zienkiewicz and J. Wu, 'A general explicit or semi-explicit algorithm for compressible and incompressible flows', *Int. j. numer. methods eng.*, **35**, 457–479 (1992).
14. D. W. Pepper, K. L. Burton, F. P. Brueckner and B. F. Blackwell, 'Numerical simulation of laminar flow with heat transfer over a backward facing step', in B. Blackwell and D. Pepper (eds), *Benchmark Problems for Heat Transfer Codes*, HTD Vol. 222, ASME, New York, 1992.
15. L. P. Franca, T. J. R. Hughes and R. Stenberg, 'Stabilized finite element methods', in M. D. Gunzburger and R. A. Nicolaides (eds), *Incompressible Computational Fluid Dynamics, Trends and Advances*, Cambridge University Press, Cambridge, 1993.
16. A. S. Almgren, J. B. Bell and W. G. Szymczak, 'A numerical method for the incompressible Navier–Stokes equations based on an approximate projection', *SIAM J. Sci. Comput.*, in press.
17. M. Kawahara, A. Anju and A. Maruoka, 'A fractional step finite element analysis of incompressible Navier–Stokes equation', *Proc. 5th Int. Symp. on Computational Fluid Dynamics*, Sendai, 1993, Vol. 1, p. 19.
18. A. S. Dvinsky and J. K. Dukowicz, 'Null-space-free methods for the incompressible Navier–Stokes equations on non-staggered curvilinear grids', *Comput. Fluids*, **22**, 685–696 (1993).
19. S. Hassanzadeh, V. Sonnad and S. Foresti, 'Finite element implementation of boundary conditions for the pressure Poisson equation of incompressible flow', *Int. j. numer. methods fluids*, **18**, 1009–1019 (1993).
20. P. M. Gresho and R. Sani, 'On pressure boundary conditions for the incompressible Navier–Stokes equations', *Int. j. numer. methods fluids*, **7**, 1111 (1987).
21. P. M. Gresho, 'Incompressible fluid dynamics: some fundamental formulation issues', *Ann. Rev. Fluid Mech.*, **23**, 413–453 (1990).
22. P. M. Gresho, 'Some interesting issues in incompressible fluid dynamics, both in the continuum and in numerical simulation', *Adv. Appl. Mech.*, **28**, 46–140 (1990).
23. G. Strang, *Introduction to Applied Mathematics*, Wellesley–Cambridge University Press, Cambridge, 1986.
24. P. M. Gresho, 'Some current CFD issues relevant to the incompressible Navier–Stokes equations', *Comput. Methods Appl. Mech. Eng.*, **87**, 201–252 (1991).
25. P. M. Gresho, '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 1: Theory', *Int. j. numer. methods fluids*, **11**, 587–620 (1990).

26. P. M. Gresho and S. 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**, 587–620 (1990).
27. R. Sani, P. M. Gresho, R. Lee, D. Griffiths and M. Engelman, 'The cause and cure(?) of the spurious pressures generated by certain FEM solutions of the incompressible Navier–Stokes equations', *Int. j. numer. methods fluids*, **1**, 17–43, 171–204 (1981).
28. P. M. Gresho, R. Sani and M. Engelman, *Incompressible Flow and the Finite Element Method*, Wiley, Chichester, 1996.
29. R. Sani and P. M. Gresho, 'Resumé and remarks on the Open Boundary Condition Minisymposium', *Int. j. numer. methods fluids*, **18**, 983–1008 (1994).
30. J. Shen, 'On error estimates of some higher order projection and penalty-projection methods for Navier–Stokes equations', *Numer. Math.*, **62**, 49–73 (1992).
31. P. M. Gresho, S. Chan, Upson and R. Lee, 'A modified finite element method for solving the time-dependent, incompressible Navier–Stokes equations. Part 1: Theory, Part 2: Applications', *Int. j. numer. methods fluids*, **4**, 557–598, 619–640 (1984).
32. G. Strang and G. J. Fix, *An Analysis of the Finite Element Method*, Prentice-Hal, Englewood Cliffs, NJ, 1973.
33. A. J. Chorin and J. E. Marsden, *A Mathematical Introduction to Fluid Mechanics*, Springer-Verlag, 1990.
34. E. H. Lurie, 'Unsteady response of a two-dimensional hydrofoil subject to high reduced frequency gust loading', *MIT Department of Ocean Engineering, Rep. 93-5*, 1993.
35. J. C. Simo and F. Armero, 'Unconditional stability and long-term behaviour of transient algorithms for the incompressible Navier–Stokes and Euler equations', *Comput. Methods Appl. Mech. Eng.*, **111**, 111–154 (1994).