INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 11, 621–659 (1990)

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

PHILIP M. GRESHO AND STEVENS T. CHAN

Lawrence Livermore National Laboratory, University of California, PO Box 808, Livermore, CA 94550, U.S.A.

SUMMARY

Ever since the expansion of the finite element method (FEM) into unsteady fluid mechanics, the 'consistent mass matrix' has been a relevant issue. Applied to the time-dependent incompressible Navier–Stokes equations, it virtually demands the use of implicit time integration methods in which full 'velocity–pressure coupling' is also inherent. The high cost of such (high-quality) FEM calculations led to the development of simpler but *ad hoc* methods in which the 'lumped' mass matrix is employed and the velocity and pressure are uncoupled to the maximum extent possible. Resulting computer codes were less expensive to use but suffered a significant loss of accuracy, caused by lumping the mass when the flow was advection-dominated and accurate transport of 'information' was important. In the second part of this paper we re-introduce the consistent mass matrix into some semi-implicit projection methods in such a way that the cost advantage of lumped mass and the accuracy advantage of consistent mass are simultaneously realized.

KEY WORDS Incompressible flows Navier–Stokes equations Projection methods Consistent mass

4. DISCRETE PROJECTION METHODS*

In this second part of the paper we go from semi-discrete to fully discrete in the form of several finite element methods that we have designed and tested. In addition to implementing some of the projection methods discussed in Part 1, we also introduce a variation on the GFEM that utilizes the consistent mass matrix in a cost-effective way. That is, the object of the second part of this paper, besides implementing some of the techniques derived above, is to put the 'mass' back where it belongs—distributed according to the Galerkin principle—not concentrated at node points as it is in the lumped mass approximation used in our (and many others') most recent FEM schemes for solving the time-dependent incompressible Navier–Stokes equations—see Gresho *et al.*[1] for an explicit time integration method and Gresho and Chan[2] for a semi-implicit one.

* The numbering of sections, tables and figures in this paper follows on from that of Part 1.

0271–2091/90/130621–39$19.50

Received August 1989
Revised January 1990

4.1. The GFEM on the Navier–Stokes equations

We demonstrated in our first publications on the subject of finite elements in fluids[3,4] that advection-dominated flows received a marvelous blessing when linear basis functions (1D) were employed (bilinear on quadrilaterals in 2D, etc.) in the Galerkin finite element method (GFEM). The blessing took the form of an extraordinarily accurate semi-discrete scheme for tracking parcels of 'material' that are carried by the fluid—typically observed as small phase speed errors (or group velocity, when analysed more properly[5]): i.e. consistent mass (CM) simulations (CM is a by-product of GFEM) display much smaller phase speed errors and therefore less of the related and cursed wiggle generator (dispersion) than do their (*ad hoc*—non-GFEM) lumped mass (LM) counterparts. This error is often described in terms of a Taylor series analysis that shows 'superconvergent' behaviour (fourth-order spatial accuracy) for 1D linear elements with CM for pure advection on a uniform mesh and only second-order for LM. While useful, this analysis is somewhat misleading because it also says that the accuracy of *both* (CM and LM) drops to first-order on any non-uniform grid. But the CM solution is *still* quite accurate (and the LM *still* quite inaccurate) on such meshes[6,7] and shows (again) that Taylor series methods, while still useful, do not tell the whole story and can even be misleading. The main point is that CM on advection is a coveted attribute. The above references were focused on the scalar advection–diffusion equation. In Gresho *et al.*[8] we showed the superiority of CM over LM when Karman vortex shedding was simulated by solving the Navier–Stokes equations.

Of course the CM approach practically *demands* the use of implicit time integration methods since it couples the time derivatives in the resulting ordinary differential equations (ODEs), and this is probably the main reason why the *ad hoc* approximation of mass lumping (e.g. by summing the rows of the CM matrix and placing the results on the diagonal) is ever employed. So, if one can 'afford' or justify for *other* reasons implicit time integration methods, CM is fine—the lunch really is free. And there is an interesting paradox here in that CM is *desired* for the hyperbolic end of the spectrum (advection-dominated; large *Pe* or *Re*), yet most common wisdom seems to suggest that *explicit* time integration methods are then more appropriate owing to the notion of 'rate of information transfer' or 'domain of influence' (~ 1 grid point per time step, etc.); at the other end of the spectrum (diffusion-dominated; low *Pe* or *Re*), however, CM is less *needed* (its gains in accuracy are not really so pronounced, though it does have other advantages there such as sending out wiggle signals[9]). Yet *these* are the problems for which the diffusive stability limit of explicit schemes definitely favours *implicit* time integration techniques! *C'est la vie.*

While the discussion above is valid for both advection–diffusion—a scalar equation—and NS (a vector system of equations), it can be especially costly to apply implicit time integration methods to the latter (for *any Re*) since these necessarily couple *all* velocity components *and* the pressure, thus leading to large linear (and *non*-linear) systems to be solved at each time step. For this reason, many NS 'solvers' avoid the use of fully coupled, implicit time integration, such as backward Euler or trapezoid rule (TR), in favour of explicit schemes (such as forward Euler or Adams–Bashforth) or semi-implicit schemes—sometimes couched in phrases like 'splitting schemes' or 'fractional step schemes' or 'projection methods'. The typical semi-implicit scheme will use—consistent with the above remarks—an explicit scheme (e.g. forward Euler or Adams–Bashforth or leap-frog) for the advection terms and an implicit scheme (typically backward Euler or TR) for the viscous term. A key objective of these semi-implicit schemes is to reduce the intense coupling and thus permit the *sequential* solution of *smaller* and linear systems. The pressure—an inherently implicit variable in incompressible flow—is obtained by forming and solving a Poisson equation, which we call the pressure Poisson equation (PPE), *when* in fact it is used to obtain pressure rather than some related potential function.

The objective of the present research is to recover the consistent mass approach in the NS equations in a cost-effective way, a goal that we have already attained for the advection–diffusion equation.[2] However, the semi-implicit approach for the NS equations seems to 'require' lumped mass in order to be even *viable*, let alone cost-effective. And indeed, virtually all NS codes to date have employed the lumped mass approach when using the PPE. But we have discovered and will demonstrate that a partial return to consistent mass—where it is needed the most (the advection process)—is achievable in the context of a cost-effective semi-implicit method employing the same *lumped* mass matrix for the discrete PPE. (It turns out that it is just as easy to use consistent mass on the viscous terms as well, so we do so.) Our first reports on this research effort appeared in Gresho and Chan,[10,11] wherein several of the new schemes are described and demonstrated. This paper presents our 'final' report on our foray into the area of semi-consistent mass (or mixed mass) methods, and we introduce virtually *all* of the schemes that we have seriously considered, mention why we dropped some, focus on the remaining useful schemes, and present some results and offer conclusions. All of our work has thus far been restricted to 2D, so that is all we discuss herein. The extension to 3D is 'straightforward' (as they all say), but we have not yet found the time to do so. We also focus on a single 'element' herein, the 4/1 element, in which bilinear basis functions are used for velocity and piecewise-constant basis functions are used for pressure.

We emphasize at the outset that all of these ideas were originally directed towards *transient* flow simulations, and *especially* those in which some identifiable quantity (such as a vortex or a cold front) is advected (though non-linearly) by the flow, and it is desired to retain the high accuracy of the CM of the GFEM for this process. The techniques are not proposed to solve low-Reynolds-number problems or those for which time-dependent advection accuracy is of less importance. It is *especially* true that the schemes are neither derived nor recommended to solve *steady* flow problems—advection-dominated or not—via time marching. (For these cases we would advocate LM via the semi-implicit Projection 2 method.) Indeed, our new compromise techniques (semi-consistent mass, SCM) could be said to deliver the 'wrong' answers (relative to GFEM and for finite Δx) at steady state (SS)—and at increased cost! However, while they are then (still) not GFEM, they are (still) *consistent* approximations and thus *do* deliver convergent results at SS—and surprisingly good ones, in fact—as we will demonstrate.

If the conventional GFEM (see e.g. Gresho *et al.*[12]) is applied to (1), the results are given by the following system of differential–algebraic equations (DAEs—see e.g. Petzold and Lötstedt[13]):

$$M\dot{u} + N(u)u = f - Ku - CP, \tag{43a}$$

$$C^{\mathrm{T}}u = g, \tag{43b}$$

where now u is a vector containing all of the (unspecified) velocities, P is a vector of pressures, M is the CM matrix ($M_{ij} = \int_\Omega \phi_i \phi_j$, where φ_k is the finite element basis function for node k) and is symmetric positive definite (SPD), C is the gradient matrix (actually the gradient operator is $M^{-1}C$, a fact that will play a major role later), its transpose is the (negative of the) divergence matrix, K is the viscous or Laplacian matrix, which is also SPD, and $N(u)$ is the advection matrix (again it is $M^{-1}K$ that actually represents $-\nu\nabla^2$ and $M^{-1}N(u)$ that actually represents $\mathbf{u}\cdot\nabla$). Finally, f is a vector that includes the BCs on velocity (from both (1c) and, when $F_n \neq 0$ or $F_\tau \neq 0$, from (1d)), as is g (it includes the BC given by (1c) and is generated by transposing those portions of $\tilde{C}^{\mathrm{T}}u = 0$, where $\tilde{C}^{\mathrm{T}}$ is the 'full' divergence matrix—including all boundary nodes—to the RHS when forming the 'condensed' system; see Gresho *et al.*[14] for details). The initial condition and the analogue of the solvability constraints (1f) and (1g) are given by

$$u(0) = u_0 \quad \text{with the constraint } C^{\mathrm{T}}u_0 = g_0. \tag{43c}$$

(The constraint (1h)—when applicable—is carried over to the semi-discrete system as $\sum_j g_j(t)=0$, which represents the discrete version of global mass conservation.)

Just as (1) implies (2), so does (43) imply the discrete analogue of (2), namely

$$(C^{\mathrm{T}}M^{-1}C)P=C^{\mathrm{T}}M^{-1}[f-Ku-N(u)u]-\dot{g}, \tag{44}$$

the discrete PPE, complete with BCs (2b) and (2c) (i.e. they are 'built-in' to the linear algebraic system)—see e.g. GS for details.[36] Again, as in the continuum, the solution of (43) gives the same (u, P) as that of (43) with (43b) replaced by (44)—and again only if due precautions are taken (i.e. all solvability constraints are respected).

We will omit (43b) in the sequel and work with (44) since it leads to the desired uncoupling of the velocities from the pressure.

Remarks

1. Since M^{-1} is dense, the CM approach of GFEM is not very viable if (44) is actually *used*; i.e. the PPE approach seems to *demand* mass lumping—as we and many others have done—with the attendant loss of phase speed accuracy for the advection process.
2. If the LM approximation is used, the sparse banded CM matrix M is replaced by the diagonal matrix ($M_{ij}=\delta_{ij}\int_\Omega \phi_i$, where δ_{ij} is the Kronecker delta) whose inverse is also diagonal.
3. Fully implicit schemes are more appropriately applied to the coupled system of (43), in which CM is quite cost-effective (partly because (43) is already expensive to solve).
4. In many simulations, g is constant in time (e.g. specified and constant inflow velocity) and thus $\dot{g}=0$; we carry the time-dependent Dirichlet BC case for generality.

4.2. The modified FEM equations and semi-implicit time integration

Thus far we have merely set the stage by reviewing the conventional GFEM. Now we take the (bold?) step of *modifying* (43a) in such a way that the advection process (at least) is treated via CM. But we wish also to use (44) instead of (43b), in which CM is 'out'. So we make the *a priori assumption* that a lumped mass matrix in the PPE—on the LHS *only*—is a *viable* approach, whether or not it is cost-effective; i.e. we assume that an LM approximation to the pressure gradient will not seriously degrade accuracy if we can use CM elsewhere. All of these words can be more precisely stated by replacing (43a) by the (admittedly *ad hoc*) momentum equation

$$\dot{u}+M_{\mathrm{L}}^{-1}CP=M^{-1}[f-Ku-N(u)u], \tag{45}$$

wherein M is the CM matrix and M_{L} is the LM matrix, in which the following obvious interpretation is stated for emphasis: the processes of advection and diffusion (and BCs) are treated consistently—note that M^{-1} has the interesting feature of coupling all nodes in the entire grid, since a global approximation to advection and diffusion is implied in the acceleration with CM—while that of the pressure gradient is treated 'inconsistently' (i.e. via LM); the matrix $M_{\mathrm{L}}^{-1}C$ is a *local*, but legitimate, approximation to ∇—almost everywhere; see below.

Remarks

1. $M_{\mathrm{L}}^{-1}C$ is *not* an approximation to ∇ on Γ_2—because of (2c).
2. Rearrangement of (45), solely for suggesting alternative interpretations, to either

$$M\dot{u}+Ku+N(u)u+MM_{\mathrm{L}}^{-1}CP=f$$

or to

$$M_L\dot{u}+CP=M_LM^{-1}[f-Ku-N(u)u]$$

introduces two (unsymmetric) matrices that, curiously, are transposes of 'averaging' matrices. That is, both $M_L^{-1}M\equiv\bar{M}$ and its inverse $M^{-1}M_L=\bar{M}^{-1}$ are averaging matrices in the sense that each row sums to unity. Also, whereas $\bar{M}_{ij}\geqslant 0$ for all i, j, $\bar{M}_{ij}^{-1}$ contains negative entries (because M^{-1} does), although all of the diagonal elements are positive.

3. Further rearrangement gives another equivalent form,

$$\bar{M}\dot{u}+M_L^{-1}[Ku+N(u)u]+\bar{M}(M_L^{-1}CP)=M_L^{-1}f,$$

which, while cumbersome, has the interesting and more useful interpretation that the advection, diffusion and source terms are *pointwise* approximations (*à la* simple finite difference methods), but the acceleration *and* pressure gradient are *averaged* approximations. (For GFEM this interpretation says that only the acceleration is an averaged quantity.)

4. $\bar{M}$, its transpose and its inverse each approximate the identity matrix as $\Delta x\to 0$ in the following sense: The action of $\bar{M}$ on a vector returns that vector to within $O(\Delta x)$—or perhaps $O(\Delta x^2)$ on a uniform mesh. $\bar{M}u=u+O(\Delta x)$ for u 'sufficiently smooth', i.e. for vectors that are sufficiently well approximated by a linear combination of the eigenvectors from the 'low end' of the spectrum of M_L and (especially) of M. (This remark applies to the bilinear elements employed herein; higher-order elements may give higher-order results.)
5. If a steady solution is attained, it too is a function of *both mass matrices* (set $\dot{u}=0$ in (45)), a perhaps suspicious result.
6. Even the Stokes problem ($N=0$) is no longer symmetric, another denigrating observation.
7. It may be worth pointing out that the modified momentum equation idea is—thus far at least—completely independent of the solution method, projection or otherwise.

The effect of this change (i.e. (43a) to (45)) is to change (44), using (43b), to the SCM PPE

$$(C^TM_L^{-1}C)P=C^TM^{-1}[f-Ku-N(u)u]-\dot{g}, \tag{46}$$

in which the 'Laplacian' $C^TM_L^{-1}C\equiv A$ is formed (easily, and only once per problem—as usual) from the LM matrix, while the RHS must (it seems) use *consistent* mass, and one may rightly query: 'Have you really gained much since the RHS is still 'impossible' (i.e. costly) to form?' The answer to this cogent question is: 'We *will* gain something because we will not need to form the RHS of (46) directly'; this issue has in fact been a principal focus of this research. And this leads to a discussion of solution methods for (45) and (46); we have considered several approaches, all based on solving (45) with a semi-implicit technique (implicit for diffusion, explicit for advection). We favour the second-order-accurate and non-dissipative TR and usually use it in our code, and do in the examples to be presented later, but the backward Euler (BE) scheme is simpler to describe (and we also have it in our code), and we will do so here. (The switch to TR is of course effected simply by replacing Ku_{n+1} on the LHS of the momentum equations below by $\frac{1}{2}Ku_{n+1}$ and subtracting $\frac{1}{2}Ku_n$ from the RHS.)

As in the continuum, it is useful, analytically, to rewrite (formally) and interpret (45) and (46) as a *projection*, this time in a finite-dimensional space. To this end we first 'solve' (46) for P and place the result in (45), yielding the equivalent continuous-in-time 'projection' ODE

$$\dot{u}+M_L^{-1}CA^{-1}\{C^TM^{-1}[f-Ku-N(u)u]-\dot{g}\}=M^{-1}[f-Ku-N(u)u]$$

or

$$\dot{u} = \wp M^{-1}[f - Ku - N(u)u] + M_L^{-1} CA^{-1}\dot{g}, \tag{47}$$

and the associated pressure gradient

$$M_L^{-1} CP = QM^{-1}[f - Ku - N(u)u] - M_L^{-1} CA^{-1}\dot{g}, \tag{48}$$

where $\wp \equiv I - M_L^{-1} CA^{-1} C^T$ and $Q \equiv I - \wp$ are now projection *matrices*, $\wp^2 = \wp$ and $Q^2 = Q$. Note that (47)+(48)=(45), analogous to the continuum projections. Again, for a more detailed discussion of the properties of these projections, refer to the Appendix—since most of those details are not actually required in the sequel. All we need to emphasize here is that (45) and (46) $\leftrightarrow$ (47) and (48), with the former pair leading to algorithms and the latter pair used in analysis.

4.2.1. Global energetics and stability. The modified equations modify, not surprisingly, the global conservation laws that are built into the discrete equations. Clearly it is important to show (or at least to know) that (45) and (46) are both consistent and stable. To examine stability in the simplest way, let f and g be zero in (43) and (45); i.e. we are considering a 'spin-down' in a closed container ($\Gamma_2 = \phi$ and $\mathbf{u} = \mathbf{0}$ on Γ). First, take the inner product of (43a) with the velocity vector to get the GFEM version of global energy transfer, i.e. the 'goal':

$$\dot{E} + u^T N(u)u + u^T CP = -u^T Ku, \tag{49a}$$

where $E \equiv u^T Mu/2$ is the kinetic energy. This equation has the following interpretation after we insist that N is a skew-symmetric matrix (for simplicity, if not for necessity; see e.g. Lee *et al.*[15] for a discussion of how to do so): (i) $u^T Nu = 0$ (for any skew-symmetric matrix), which says that the advection process in the closed box properly does not contribute anything to the kinetic energy balance; (ii) $u^T CP = P^T C^T u = 0$, the pressure gradient (also properly) doesn't either; and (iii) since K is SPD, viscosity causes a monotonic decay (via viscous dissipation) of kinetic energy. This behaviour,

$$\dot{E} = -u^T Ku \leqslant 0, \tag{49b}$$

is of course consistent with the continuum equations and also shows that the DAEs of (43) are stable.

Next, repeat the procedure with the modified equation (45). We get either

$$\frac{1}{2}\frac{d}{dt} u^T M_L u + u^T M_L M^{-1} N(u)u + u^T CP = -u^T M_L M^{-1} Ku$$

or

$$\frac{1}{2}\frac{d}{dt} u^T Mu + u^T N(u)u + u^T M M_L^{-1} CP = -u^T Ku,$$

and the interpretations of stability and energy transfer/balance are more obscure. For example, we now have two norms to consider: one based on M_L and the other on M, with the latter making more sense in more terms. In neither are we able to be very definite about 'energy statements' since the matrices $M_L M^{-1} N$ and $M_L M^{-1} K$ are *indefinite*. (Even though $M^{-1} K$ has real positive eigenvalues and $M^{-1} N$ has purely imaginary ones, the quadratic forms are still indefinite; perhaps further effort will produce more useful results, but we do not yet have them.). All we can say for sure is this.

(i) $u^T CP = 0$ and thus the pressure gradient plays no role in the M_L-norm.

(ii) The advection term plays no role and the viscous term still dissipates kinetic energy in the M-norm.
(iii) If u is stable in one norm, it is in the other as well.

Thus

$$\dot{E}+u^{\mathrm{T}}MM_{\mathrm{L}}^{-1}CP=-u^{\mathrm{T}}Ku\leqslant 0$$

is the strongest statement that can be made. Similarly, attempts to show stability via the matrix method (eigenvalues $\leqslant 1$) have thus far proven intractable. About the only 'defence' we have at this time is that experimental evidence at least *suggests* stability; i.e. the 'energy' *does* seem to remain bounded. It must be the fact that both $M_{\mathrm{L}}M^{-1}$ and MM_{L}^{-1} (its inverse) tend to approximate (in the appropriate sense) the identity matrix that is saving us; i.e. if $MM_{\mathrm{L}}^{-1}=I+O(\Delta x)$ or better, the spurious 'pressure work' term in the M-norm above becomes $u^{\mathrm{T}}[I+O(\Delta x)]CP=P^{\mathrm{T}}C^{\mathrm{T}}[I+O(\Delta x)]u=O(\Delta x)$ to give $\dot{E}=-u^{\mathrm{T}}Ku+O(\Delta x)$—at least 'fine mesh' (viscous) simulations are likely to be stable. Indeed, it is this property ($\bar{M}\approx I$) that is also needed to permit both *consistency* and meaningful *steady* solutions, for which $M_{\mathrm{L}}^{-1}CP=M^{-1}[f-Ku-N(u)u]$.

So—at least for the time being—we are stymied regarding theoretical stability results for any of the schemes presented below that solve (45) and (46) approximately. We can only report experimental evidence that strongly suggests that the DAEs are stable and that our time integration techniques are at least conditionally stable.

Finally, we *shall* demonstrate consistency for each of the schemes used to 'solve' (45) and (46).

4.2.2. A semi-implicit pressure Poisson equation scheme (PPE). Before actually embarking on fully discrete *projection* methods, it seems appropriate, and useful, to show how to obtain a solution to (45) and (46) using but a small and inexpensive (multiply CP by MM_{L}^{-1}) modification of our earlier LM semi-implicit method that is even somewhat surprising. (Also, we coded and tested it.) This scheme follows almost directly from our earlier one,[2] in which M_{L}^{-1} is indeed (still) used on the RHS of the PPE. Starting with $n=0$ and $C^{\mathrm{T}}u_0=g_0$:

(1) Solve $AP_n=C^{\mathrm{T}}M_{\mathrm{L}}^{-1}[f_n-Ku_n-N(u_n)u_n-(g_{n+1}-C^{\mathrm{T}}u_n)/\Delta t$ for P_n.
(2) Solve $M(u_{n+1}-u_n)/\Delta t+Ku_{n+1}=f_n-N(u_n)u_n-MM_{\mathrm{L}}^{-1}CP_n$ for u_{n+1}; i.e. solve $(M+\Delta tK)u_{n+1}=Mu_n+\Delta t[f_n-N(u_n)u_n-MM_{\mathrm{L}}^{-1}CP_n]$.
(3) Bump n and go to Step (1).

Remarks

1. Even though $C^{\mathrm{T}}u_0=g_0$, $C^{\mathrm{T}}u_n\neq g_n$ for $n>0$. We return to this issue below.
2. We use a direct (skyline/profile) method[1] to solve the PPE (both here and in the schemes to follow) whenever possible, which includes essentially all 2D problems (for 3D we usually use an ICCG solver; see e.g. Gresho[16]).
3. The matrix multiplying u_{n+1} is SPD, and the conjugate gradient method with diagonal scaling (preconditioning) works well (again, both here and in the schemes to follow); e.g. 'convergence' generally occurs in 1–10 iterations using

$$\|\delta u\|/\|u\|\leqslant 10^{-4} \quad \text{and} \quad \|\text{residual}\|/\|\text{RHS}\|<10^{-4}.$$

4. The velocity components are obtained sequentially: first u, then v.
5. The viscous matrix contains a balancing diffusivity tensor (BTD) term $u_iu_j\Delta t/2$, where u_i is the (current time, t_n) ith Cartesian velocity component, which is added to the physical viscosity (ν) in order to balance the destabilizing truncation error incurred when FE (forward Euler) is used on the advection terms. This technique, discussed in Gresho *et al.*[1] and Gresho

and Chan,[2] stabilizes the scheme up to CFL numbers ($u\Delta t/\Delta x$) of about 5–10, and there is no diffusive stability limit.

6. The 'penalizing' divergence term $C^{\mathrm{T}}u_n/\Delta t$, but *not* $g_n/\Delta t$, on the RHS of the PPE in step (1) is *crucial* to the success of the scheme—even more so than it was in the LM version of Gresho and Chan.[2]
7. There is evidence, but no proof, that a projection of u_{n+1} from this scheme is very close (identical?) to u_{n+1} from the Projection 2 scheme described below.

Consistency analysis. Following on from Remarks 1 and 6 above, let us analyse the scheme both to assure consistency with (45) and (46)—which may at this point *appear* to be impossible—and to see how the (discrete) divergence of u behaves. To start, we rewrite step (2) in terms of the projection operator (matrix), after defining $F_n \equiv f_n - N(u_n)u_n$:

$$\frac{u_{n+1}-u_n}{\Delta t} = M^{-1}(F_n - Ku_{n+1}) - Q\left(M_{\mathrm{L}}^{-1}(F_n - Ku_n) + \frac{u_n}{\Delta t}\right) + \frac{M_{\mathrm{L}}^{-1}CA^{-1}g_{n+1}}{\Delta t}.$$

Next, form the divergence, noting that $C^{\mathrm{T}}Q = C^{\mathrm{T}}$:

$$\frac{C^{\mathrm{T}}(u_{n+1}-u_n)}{\Delta t} = C^{\mathrm{T}}[M^{-1}(F_n - Ku_{n+1}) - M_{\mathrm{L}}^{-1}(F_n - Ku_n)] - \frac{C^{\mathrm{T}}u_n}{\Delta t} + \frac{g_{n+1}}{\Delta t},$$

and the 'crucial' cancellation of $C^{\mathrm{T}}u_n/\Delta t$ leads to

$$C^{\mathrm{T}}u_{n+1} = \Delta t C^{\mathrm{T}}[(M^{-1} - M_{\mathrm{L}}^{-1})F_n - (M^{-1}Ku_{n+1} - M_{\mathrm{L}}^{-1}Ku_n)] + g_{n+1},$$

which, assuming $M^{-1} - M_{\mathrm{L}}^{-1} = O(\Delta x)$ and $u_{n+1} = u_n + O(\Delta t)$, gives $C^{\mathrm{T}}u_{n+1} - g_{n+1} = \Delta t[O(\Delta x) + O(\Delta t)]$, where the $O(\Delta x)$ term is a direct result of the mixed mass matrices. This is as close as this scheme can come to satisfying (43b). Some of this divergence error is even present at steady state, again owing to the mass mixing. (For LM, used in our first semi-implicit scheme, $C^{\mathrm{T}}u - g = O(\Delta t^2)$ and vanishes at steady state.)

Next, inserting the divergence result (at t_n) into the PPE (step (1) of the algorithm) gives

$$AP_n = C^{\mathrm{T}}M_{\mathrm{L}}^{-1}(F_n - Ku_n) + C^{\mathrm{T}}[(M^{-1} - M_{\mathrm{L}}^{-1})F_{n-1} - (M^{-1}Ku_n - M_{\mathrm{L}}^{-1}Ku_{n-1})] - (g_{n+1} - g_n)/\Delta t,$$

where it is important to note, and even worth proving, that it is in fact the (non-zero) divergence of $C^{\mathrm{T}}u_n$ in the above divergence equation (i.e. $C^{\mathrm{T}}u_n \neq g_n$) that actually *converts* it from an LM PPE (which is the PPE actually solved by the computer) to an SCM PPE (which is the PPE required by the theory). To see this conversion, rearrange the above PPE to

$$AP_n = C^{\mathrm{T}}M^{-1}(F_{n-1} - Ku_n) + C^{\mathrm{T}}M_{\mathrm{L}}^{-1}[(F_n - F_{n-1}) - K(u_n - u_{n-1})] - (g_{n+1} - g_n)/\Delta t,$$

perform a Taylor series expansion about t_n to obtain

$$AP_n = C^{\mathrm{T}}M^{-1}(F_n - Ku_n) - \Delta t C^{\mathrm{T}}M^{-1}\dot{F}_n + \Delta t C^{\mathrm{T}}M_{\mathrm{L}}^{-1}(\dot{F}_n - K\dot{u}_n) - \dot{g}_n - \frac{\Delta t}{2}\ddot{g}_n + O(\Delta t^2),$$

and pass to the limit to obtain (46).

In a similar way, inserting the above pressure into the momentum equation (i.e. into step (2) of the above algorithm) yields, after some 'algebra', (47), and thus (45), as $\Delta t \to 0$.

Remarks

1. The lack of a divergence-free solution is not an asset—though it isn't either in the LM version (i.e. that of Gresho and Chan[2]) and that scheme usually works quite well—albeit with the

advantage of one order (in Δt) smaller divergence and no SS divergence error. *Both of these schemes 'work' because (i) the PPE is properly penalized—and (ii) the BHM saves the rest.*

2. The start-up of this scheme is not smooth: $P_1 - P_0 = O(1)$ in Δt as $\Delta t \to 0$ because P_0 is an LM pressure and P_1 is an SCM pressure; it is also true though that $P_1 - P_0 = O(\Delta x)$. In fact, $A(P_1 - P_0) = C^{\mathrm{T}}(M^{-1} - M_{\mathrm{L}}^{-1})(F_0 - Ku_0)$ for $\Delta t \to 0$. Ironically, this discontinuity is *caused* by the divergence-free initial condition; i.e. we begin with u_0 satisfying (43c). After step (1), however, the pressure *is* smooth—$P_{n+1} - P_n = O(\Delta t)$ for $n \geqslant 1$—and the velocity is no longer divergence-free; also, P_{n+1} satisfies (46), at least for $\Delta t \to 0$. The appropriate 'remedy' for this inconsistency (if one is really needed) may be to just take one *very* small time step and report the pressure there as the 'initial' pressure. Finally, if one computes the initial pressure using CM on the RHS, via $AP_0 = C^{\mathrm{T}}M^{-1}(F_0 - Ku_0) - (g_1 - g_0)/\Delta t$, which we have tried, the results are worse yet—it then requires *two* time steps to 'recover', the first step (P_1) being an LM result.
3. If SS is attained, it satisfies (45) with $\dot{u} = 0$ as well as (46) with $\dot{g} = 0$. But it is also true that it does *not* satisfy (43b); in fact,

$$(C^{\mathrm{T}}u - g) = \Delta t C^{\mathrm{T}}(M^{-1} - M_{\mathrm{L}}^{-1})[f - Ku - N(u)u] = O(\Delta t)O(\Delta x).$$

4. The semi-discrete version of BTD (i.e. continuous in space, discrete in time) leads to the following result:

$$\nabla \cdot \mathbf{u}_{n+1} = \Delta t \nabla \cdot \left[\left(\nu \nabla^2 + \frac{\Delta t}{2} (\mathbf{u}_n \cdot \nabla)^2 \right) (\mathbf{u}_{n+1} - \mathbf{u}_n) \right],$$

which is $O(\Delta t^2)$ and vanishes at steady state.
5. At the end of the day, we shall reject this scheme.

4.2.3. Projection 1. This scheme is a simple and nearly direct extension of Chorin's famous[17] projection scheme into FEM using mixed mass matrices, and was first introduced in Gresho and Chan.[10] The essence and beauty of this scheme lie in its simplicity. Given a divergence-free velocity u_n, the first-order projection scheme is, starting with $n = 0$:

(1) Solve

$$\frac{M(\tilde{u} - u_n)}{\Delta t} + K\tilde{u} = f_n - N(u_n)u_n$$

for the intermediate velocity field; i.e. solve $(M + \Delta t K)\tilde{u} = Mu_n + \Delta t[f_n - N(u_n)u_n]$ for $\tilde{u}$.

(2) Project $\tilde{u}$ onto the divergence-free subspace and simultaneously obtain the pressure field as follows: solve $\tilde{u} = u_{n+1} + M_{\mathrm{L}}^{-1}C\varphi$ and $C^{\mathrm{T}}u_{n+1} = g_{n+1}$; i.e. first solve $(C^{\mathrm{T}}M_{\mathrm{L}}^{-1}C)\varphi = C^{\mathrm{T}}\tilde{u} - g_{n+1}$ for φ and then compute $u_{n+1} = \tilde{u} - M_{\mathrm{L}}^{-1}C\varphi$.

(3) (Optional) Compute $\tilde{P} = \varphi/\Delta t$.

(4) Bump n and go to Step (1).

Remarks

1. Remarks 2–5 for the PPE scheme also apply here.
2. The scheme is a CM predictor and an LM projector/corrector, i.e. a sort of predictor–corrector where the corrector is a projector. The projection step can be formally expressed as

$$u_{n+1} = \wp\tilde{u} + M_{\mathrm{L}}^{-1}CA^{-1}g_{n+1},$$

which defines u_{n+1} as an affine transformation on $\tilde{u}$. (While $\wp$ is still a valid projection, we are usually—for $g \neq 0$—more interested in the closely related affine transformation.) But since $w \equiv M_L^{-1} C A^{-1} g_{n+1}$ satisfies $\wp w = 0$, this affine transformation could also be called a projection in the sense that $u_{n+1} = \wp \tilde{u} + w$ implies $\wp u_{n+1} + w = u_{n+1}$, a result we shall use below.

3. Unlike the semi-implicit PPE scheme, the velocity is always divergence-free—even if it did not start out that way; see Remark 8 below.
4. Another difference (besides CM) from Chorin's method regards intermediate BCs: whereas he (and others as well, e.g. Kim and Moin[18]) generated special BCs for the intermediate velocity field that produced both penetration and slip at solid walls, we use the same BCs for $\tilde{u}$ as are used for u—i.e. the *physical* BCs—as discussed earlier. Also, as also discussed earlier, we overspecify the tangential velocity during the projection step. But see also the following remark.
5. If $K\tilde{u}$ in the first step were replaced by Ku_n, we would have the simple forward Euler scheme. (The forward Euler method is in fact the simplest projection method; it is also more 'honest' than semi-implicit projection methods in that it more rigorously enforces the no-slip condition.) Such a scheme was in fact introduced by Donea *et al.*[19,20] in the LM approximation, except that they modified the forward Euler method to preclude the spurious pressure mode by permitting some slip—see also Gresho.[21]

   It is interesting to note that Donea *et al.* modified a forward Euler method to eliminate the checkerboard (CB) pressure mode in a way that seems somewhat illegitimate, and that we modify the projection method in an 'equally illegitimate' way (overspecifying the BCs) to save coding and computing but which *retains* the CB mode that we could have *legitimately* precluded! Considering some of the admonitions put forth in Sani *et al.*[22] regarding the CB mode, we should probably change our own codes.
6. $\tilde{P}$ is indexless because it is an 'intermediate' pressure, here in the sense that the time level at which it applies is ambiguous. It can be shown, however, that if the true end-of-step pressure (at t_{n+1}) is computed from (46), then $P_{n+1} - \tilde{P} = O(\Delta t)$. It is also true, however, that $P_n - \tilde{P} = O(\Delta t)$ but with a different constant, where P_n also comes from (46). So we see that $\tilde{P}$ really *is* $O(\Delta t)$ away from any physical pressure of interest, and perhaps that is all that matters. (It is *indeed* an intermediate pressure.) Similar remarks apply to $\tilde{u}$.
7. The *effective* pressure gradient, which would even be *accurate*—to $O(\delta^2)$—on Γ_1, is easily derived to be

$$GP|_{\text{eff}} \equiv M_L^{-1}(M + \Delta t K) M_L^{-1} C \tilde{P},$$

   which is even computable, and is the discrete analogue of $P_{\text{eff}} = (I - \delta^2 \nabla^2) P_{m+1}$ that was presented earlier for the semi-discrete system.
8. It is no longer necessary (but still advisable) to satisfy the original solvability constraint $C^T u_0 = g_0$. This scheme, like the backward Euler scheme (on (43), or on (45) and (46)), is *oblivious* to the well-posedness of the initial data. They each have the (convenient?) property—and one that has probably fooled some CFD practitioners—of converting *any* initial velocity field (well-posed or not) into one that *is* divergence-free at the end of the first time step (and evermore), with the result that the transient flow actually computed may be very different than the one imagined; See Gresho (1991).[37]

   In fact, if $C^T u_0 \neq g_0$ and Δt is small, both Projection 1 and BE would project u_0 to the divergence-free subspace in their respective 'appropriate' manners: $u_1 = \wp u_0 + M_L^{-1} C A^{-1} g_1 + O(\Delta t)$ for the former and $u_1 = \wp_c u_0 + M^{-1} C (C^T M^{-1} C)^{-1} g_1 + O(\Delta t)$ for the latter, where $\wp_c$ is the CM version of the LM projection matrix, $\wp_c \equiv I - M^{-1} C (C^T M^{-1} C)^{-1} C^T$, each giving $C^T u_1 = g_1$ and $u_1 - u_0 = O(1)$ in Δt since u_0 was 'improper'. But each would also deliver

a very large pressure: $P_1 = O(1/\Delta t)$. (For 'large' Δt the projections are not so simple to state.) Clearly a *lumped* mass BE scheme (generally not a suggested approach) would deliver—to $O(\Delta t)$—the *same* u_1 as does Projection 1.

9. While the scheme was designed for time-dependent advection-dominated flows, it is also of interest to see how it behaves for flows that reach steady state (SS). It turns out that if an SS is attained, it is a function of Δt and both mass matrices:

$$[K+N(u)]u+(M+\Delta tK)M_L^{-1}CP=f,$$

vis-à-vis the GFEM result from (45),

$$[K+N(u)]u+CP=f,$$

with both results also satisfying $C^Tu=g$.

Note that the continuum (in space) version of the above Projection 1 result, using $(M^{-1}K)\sim -\nu\nabla^2$, is $\mathbf{u}\cdot\nabla\mathbf{u}+(I-\nu\Delta t\nabla^2)\nabla P=\nu\nabla^2\mathbf{u}$, a result that also applies to the original schemes of Chorin[17] and Temam,[23] and that we earlier called (in the time-dependent version) the modified/perturbed momentum equation. Finally, the divergence of this equation yields $(I-\nu\Delta t\nabla^2)\nabla^2P=-\nabla\cdot(\mathbf{u}\cdot\nabla\mathbf{u})$, the BHE discussed earlier.

Consistency analysis. Since the underlying DAEs are by no means obvious for this scheme, nor is it obvious that they even represent a legitimate approximation to the NS equations, it is important to derive them by analysing the scheme for $\Delta t\to 0$. (It is of course also necessary to test them on the computer—i.e. using finite Δt.) Towards this end we begin by writing the scheme in the equivalent form

$$\tilde{u}=(M+\Delta tK)^{-1}\{Mu_n+\Delta t[f_n-N(u_n)u_n]\},$$
$$u_{n+1}=\wp\tilde{u}+M_L^{-1}CA^{-1}g_{n+1}.$$

Now let $\Delta t\to 0$ to obtain

$$u_{n+1}=\wp\{u_n+\Delta tM^{-1}[f_n-Ku_n-N(u_n)u_n]-\Delta t^2M^{-1}KM^{-1}[f_n-Ku_n-N(u_n)u_n]\}$$
$$+M_L^{-1}CA^{-1}g_{n+1}+O(\Delta t^3),$$

from which it is clear, using $\wp u_n=u_n-M_L^{-1}CA^{-1}g_n$, that (47) is recovered, and thus (45) and (46) are recovered as $\Delta t\to 0$; the scheme is indeed a consistent approximation to the desired DAEs.

4.2.4. Projection 2. A seemingly unattractive aspect of Projection 1 is that the pressure gradient is completely disregarded/discarded in the computation of $\tilde{u}$—a feature that seems to want to send $\tilde{u}$ farther than necessary away from a true (and divergence-free) velocity. So we introduce a discrete version of Projection 2 with nearly consistent mass, a scheme that improves the accuracy by providing a good-yet-inexpensive estimate of the pressure gradient in the computation of $\tilde{u}$.

The Projection 2 algorithm is the following, starting with $n=0$. Given u_n and P_n with $C^Tu_n=g_n$:

(1) Solve

$$\frac{M(\tilde{u}_{n+1}-u_n)}{\Delta t}+K\tilde{u}_{n+1}+MM_L^{-1}CP_n=f_n-N(u_n)u_n$$

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

$$(M+\Delta tK)\tilde{u}_{n+1}=Mu_n+\Delta t[f_n-N(u_n)u_n-MM_L^{-1}CP_n].$$

(2) Project $\tilde{u}_{n+1}$ and update the pressure as follows: solve

$$\tilde{u}_{n+1}=u_{n+1}+M_L^{-1}C\varphi \quad \text{and} \quad C^T u_{n+1}=g_{n+1};$$

i.e. first solve $A\varphi=C^T\tilde{u}_{n+1}-g_{n+1}$ for φ, then compute $u_{n+1}=\tilde{u}_{n+1}-M_L^{-1}C\varphi$; finally, compute $P_{n+1}=P_n+2\varphi/\Delta t$.

(3) Bump n and go to Step (1).

Remarks

1. Remarks 1–4 for Projection 1 also apply here.
2. For start-up ($n=0$), a mass matrix problem must be solved to obtain P_0 from $AP_0=C^TM^{-1}[f_0-Ku_0-N(u_0)u_0]-\dot{g}_0$, where $C^Tu_0=g_0$. This is done in our code as follows: (i) solve $Mx=f_0-Ku_0-N(u_0)u_0$ for x using the diagonally scaled conjugate gradient method; (ii) solve $AP_0=C^Tx-\dot{g}_0$ for P_0. (An alternative start-up procedure is: use Projection 1 and a very small initial time step.) Note that, unlike Projection 1, $C^Tu_0=g_0$ is indeed a firm requirement—as it is for the DAEs given by (43)—thus making it a more 'honest' scheme.
3. The intermediate velocity is closer to u_{n+1} than it is to u_n: $u_{n+1}-\tilde{u}_{n+1}=O(\Delta t^2)$ and $\tilde{u}_{n+1}-u_n=O(\Delta t)$. This is the reason we endow it with a time index, *vis-à-vis* Projection 1.
4. The *effective* pressure gradient for this scheme is

$$\begin{aligned} GP|_{\text{eff}} &\equiv M_L^{-1}MM_L^{-1}CP_n+M_L^{-1}(M+\Delta tK)M_L^{-1}C(P_{n+1}-P_n)/2 \\ &= M_L^{-1}MM_L^{-1}C(P_n+P_{n+1})/2+O(\Delta t^2). \end{aligned}$$

5. If an SS is attained, it too (like both PPE and Projection 1) will be a mixed mass result; but unlike Projection 1 it is independent of Δt, and unlike PPE it is divergence-free. It is given by (45) with $\dot{u}=0$ and by (46) with $\dot{g}=0$, so that it satisfies (43b) as well.

Consistency analysis. Letting $F_n\equiv f_n-N(u_n)u_n$ again, we have

$$\begin{aligned} \tilde{u}_{n+1} &= (I+\Delta tM^{-1}K)^{-1}[u_n+\Delta t(M^{-1}F_n-M_L^{-1}CP_n)], \\ u_{n+1} &= \wp\tilde{u}_{n+1}+M_L^{-1}CA^{-1}g_{n+1}. \end{aligned}$$

For small Δt these yield

$$\tilde{u}_{n+1}=u_n+\Delta t(I-\Delta tM^{-1}K)\tilde{a}_n+O(\Delta t^3),$$

where

$$\tilde{a}_n\equiv M^{-1}(F_n-Ku_n)-M_L^{-1}CP_n,$$

and the tilde on $\tilde{a}_n$ is used to remind us that it is not a divergence-free acceleration. (In fact, $C^T\tilde{a}_n-\dot{g}_n=O(\Delta t)$, as will be seen below.) Finally,

$$u_{n+1}=u_n+M_L^{-1}CA^{-1}(g_{n+1}-g_n)+\Delta t\wp\tilde{a}_n+O(\Delta t^2),$$

where we have (again) used $\wp u_n=u_n-M_L^{-1}CA^{-1}g_n$. But since $\wp M_L^{-1}C=0$, this is equivalent to

$$u_{n+1}=u_n+M_L^{-1}CA^{-1}(g_{n+1}-g_n)+\Delta t\wp M^{-1}(F_n-Ku_n)+O(\Delta t^2),$$

which, upon division by Δt and passage to the limit, yields (47).

While the above is sufficient to also ensure that P_n converges to (46), it may be helpful to demonstrate this more directly. Inserting the $\tilde{u}_{n+1}$ result into the PPE of the projection step yields

$$A(P_{n+1}-P_n)/2=C^T(I-\Delta tM^{-1}K)\tilde{a}_n-(g_{n+1}-g_n)/\Delta t+O(\Delta t^2),$$

which, using $g_{n+1}=g_n+\Delta t\dot{g}_n+O(\Delta t^2)$, yields

$$A(P_{n+1}+P_n)/2=C^{\mathrm{T}}M^{-1}(F_n-Ku_n)-\dot{g}_n+O(\Delta t),$$

and it is clear that (46) is recovered as $\Delta t\to 0$. Finally, it follows that $C^{\mathrm{T}}\tilde{a}_n\equiv C^{\mathrm{T}}M^{-1}(F_n-Ku_n)-AP_n=\dot{g}_n+O(\Delta t)$, as promised. As in Projection 1, this scheme delivers a pressure that is $O(\Delta t)$ away from that given by the PPE—even though we had hoped for more when designing the scheme.

4.2.5. Projection 3. For completeness we present the associated SCM version of Projection 3, but remark that we have not yet tested this scheme in the laboratory.

(0) Given u_0 with $C^{\mathrm{T}}u_0=g_0$, P_0, and $\dot{P}_0$, set $n=0$ and do the following:
(1) Solve

$$M\frac{\tilde{u}_{n+1}-u_n}{\Delta t}+K\tilde{u}_{n+1}+MM_{\mathrm{L}}^{-1}C(P_n+\Delta t\dot{P}_n)=f_n-N(u_n)u_n$$

for the intermediate velocity; i.e. solve

$$(M+\Delta tK)\tilde{u}_{n+1}=Mu_n+\Delta t[f_n-N(u_n)u_n-MM_{\mathrm{L}}^{-1}C(P_n+\Delta t\dot{P}_n)].$$

(2) Project $\tilde{u}_{n+1}$ and update the pressure via $\tilde{u}_{n+1}=u_{n+1}+M_{\mathrm{L}}^{-1}C\varphi$ and $C^{\mathrm{T}}u_{n+1}=g_{n+1}$; i.e. first solve $A\varphi=C^{\mathrm{T}}\tilde{u}_{n+1}-g_{n+1}$ for φ, then compute $u_{n+1}=\tilde{u}_{n+1}-M_{\mathrm{L}}^{-1}C\varphi$, $P_{n+1}=P_n+\Delta t\dot{P}_n+3\varphi/\Delta t$ and $\dot{P}_{n+1}=(P_{n+1}-P_n)/\Delta t$.
(3) Bump n and go to Step (1).

Remarks

1. $\dot{P}_0$ could be obtained by taking one very small time step with, for example, forward Euler, as shown earlier.
2. If this scheme truly has lower-order projection error, it should probably be implemented with higher-order ODE methods (say second-order or better).
3. The effective pressure gradient is

$$\begin{aligned}GP|_{\mathrm{eff}}&=M_{\mathrm{L}}^{-1}MM_{\mathrm{L}}^{-1}C(P_n+\Delta t\dot{P}_n)+M_{\mathrm{L}}^{-1}(M+\Delta tK)M_{\mathrm{L}}^{-1}C(P_{n+1}-P_n-\Delta t\dot{P}_n)/3\\&=M_{\mathrm{L}}^{-1}MM_{\mathrm{L}}^{-1}C[2(P_n+\Delta t\dot{P}_n)+P_{n+1}]/3+O(\Delta t^3).\end{aligned}$$

4. A subcycling strategy (at least when $T_{n+1}>O(\Delta t)$), similar to that proposed earlier for Projection 2, might even be more useful here, in which case the pressure estimate during the intermediate velocity calculation would be changed from $P_n+\Delta t\dot{P}_n$ to $P_n+(t-t_n)\dot{P}_n$—*à la* our old explicit projection method.[1]

4.2.6. Two bad schemes. One thing we would like to point out/admit, in the hope of saving others from wasting time in the future, is that we were actually foolish enough to try a seemingly obvious modification of two of the schemes described earlier, while still deep under the influence of 'try CM for momentum, LM for pressure'. (Also, Fortran is usually easier than mathematics.) One scheme is based on the PPE approach and the other on the Projection 2 method. Both simply omit the MM_{L}^{-1} factor in the discrete pressure gradient of their respective algorithms. The omission is simple and may appear innocuous (perhaps even desirable!), but the resulting schemes, and their analyses, are not. Suffice it to say that the schemes are neither consistent nor stable. A modified momentum equation is the only successful approach that we have found—*and it is significant that this is the same momentum equation that underlies the Projection 1 scheme.*

4.2.7. Global errors in time. While too lengthy to reproduce here (see Gresho *et al.*[14] for details), it can be shown, via global error analysis (local error analysis is not as useful for DAEs as it is for ODEs) applied to the Projection 1 and Projection 2 schemes presented above, that:

(1) The global errors (difference between approximate and exact solutions of the DAEs) in both velocity and pressure are $O(\Delta t)$.
(2) The local error (difference in solutions after one time step, assuming that both started from the exact solution), which is easily obtained from the global error results, is $O(\Delta t^2)$ in velocity and $O(\Delta t)$ in pressure.

Thus local errors in velocity accumulate to give larger global errors (as in most ODE methods), while those in pressure do not. Of course this error analysis, besides demonstrating the important fact of convergence as $\Delta t \to 0$, says no more than 'one scheme is as good as the other'. We must therefore rely on some numerical experiments—at various *finite* Δt—to help select the best and reject the worst.

A final remark on stability: since Stokes flow appears to be (i.e. experimentally) unconditionally stable, it is probably true that the mixed mass projection step does not affect the stability of the intermediate velocity calculation.

4.2.8. Summary comparison of the schemes. Before presenting numerical results and trying to 'pick a winner', it may be useful to summarize the salient features of the three schemes that we have tested. This is done in Table II, in which the descriptors (e.g. least, most) are not intended to imply *large* differences; indeed most of the differences that we have observed have been quite *small*.

Table II. A comparison of three nearly consistent mass schemes

Property	PPE	Projection 1	Projection 2
Div-free?	No	Yes	Yes
Start-up	Not smooth; $P_1 - P_0 = O(1)$ in Δt	No problems	Must solve a CM matrix problem
Steady state	u OK P OK div $u = O(\Delta t)O(\Delta x)$	u and P are in error by $O(\Delta t)$;* div $u = 0$	u OK P OK div $u = 0$
Global accuracy in time	$O(\Delta t)$ in u and P	Ditto	Ditto
Local accuracy in time	$O(\Delta t^2)$ in u $O(\Delta t)$ in P	Ditto	Ditto
Spurious dissipation	Largest	Intermediate	Smallest
Stability	Least	Most	Intermediate

* At least in the boundary layer, $\delta = O[\sqrt{(\nu \Delta t)}]$.

(i) All schemes involve the solution of $(M + \Delta t K)x = b$ for which the diagonally scaled, conjugate gradient method is used (first u, then v) and usually converges in 1–10 iterations.

(ii) All schemes involve the solution of an 'LM' Poisson equation, with $A \equiv C^{\mathrm{T}} M_{\mathrm{L}}^{-1} C$.

(iii) The cost per step of each is about equivalent, though P1 < P2 < PPE.

(iv) The last two entries in the table are based on numerical results.

While Projection 2 *seems* to show more advantages and less disadvantages than the others, it is somewhat disappointing that it shows up to be little better than Projection 1, both in theory (i.e. in the ODE/DAE theory) and in practice. Except for SCM, the PPE method displays only disadvantages and is no longer advocated.

A final remark: in all of the schemes, it is of course permissible to lump the mass everywhere ($M \rightarrow M_L$), in which case they simplify to schemes that are closer to finite difference projection methods.

4.2.9. Final remarks on Projection 2. We have experimentally determined the following behaviour: if the pressure update is computed as $P_{n+1} = P_n + \gamma\varphi/\Delta t$, where $\gamma = 2$ defines the 'conventional' Projection 2 method, then the scheme still works pretty well (especially for the velocity) for any value of $\gamma \leqslant 2$, i.e. $0 \leqslant \gamma \leqslant 2$, and $\gamma > 2$ is actually unstable—except perhaps for Stokes flow. In the results to follow we had $\gamma = 1$; this is because we had not yet discovered that $\gamma = 2$ is theoretically better. We note also that:

(i) $\gamma = 0$ actually converts the scheme to a slightly modified Projection 1 scheme.
(ii) $\gamma = 2$ is optimum from the *local* analyses presented earlier.
(iii) Global effects (i.e. accumulation)—neglected in most of the analysis—are probably the cause of the apparent success for all $\gamma \leqslant 2$, although detailed understanding is still lacking.

We believe that a good subcycling strategy might, probably with $\gamma = 2$, lead to a more cost-effective Projection 2 scheme.

5. NUMERICAL RESULTS

5.1. Lid-driven cavity

Before demonstrating the new schemes on the type of problems for which they were designed, we shall jump directly to a steady state simulation, with a (nearly) known solution, to help remove any doubt that mixed mass 'works'. The test problem is one of the standards: the (non-leaky) lid-driven cavity. Here we use the same graded 50×50 mesh that we used in Gresho *et al.*[1] and pick one Reynolds number: 5000. The time step was 0·025 using TR, five times larger than the CFL stability-limited Δt used in our explicit code, and gives a maximum Courant (or CFL) number of ~ 2.

Figures 1 and 2 show the streamlines and pressure contours from the Projection 2 scheme. Similar results from the Projection 1 scheme are shown in Gresho and Chan,[10] and PPE results are not shown because, in fact, all three schemes delivered virtually identical results—consistent with the theory and showing that the SS Δt-effects from PPE and Projection 1 are small for this problem. The ψ- and P-contours look 'just like' those in Ghia *et al.*[24] for ψ and in Gresho *et al.*[1] for P.

A more quantitative comparison is offered in Tables III and IV, wherein relevant extrema in streamfunction and velocity are tabulated. Unless otherwise stated, all results are from the new SCM schemes. The key observations seem to be these: (1) all three SCM schemes deliver essentially the same results; (2) all three deliver weaker flows than desired; (3) the LM results from Projection 1 (which would agree with those from PPE and Projection 2) agree well with the SCM results; (4) when Δt is reduced to that used in the explicit (forward Euler with BTD) code, the eddies strengthen and the solutions become closer to the right answer.

Hence we conclude that all schemes 'work', but that semi-implicit time integration with BTD is subject to noticeable 'excess diffusion' at SS when Δt is not small. (As we shall show later, the

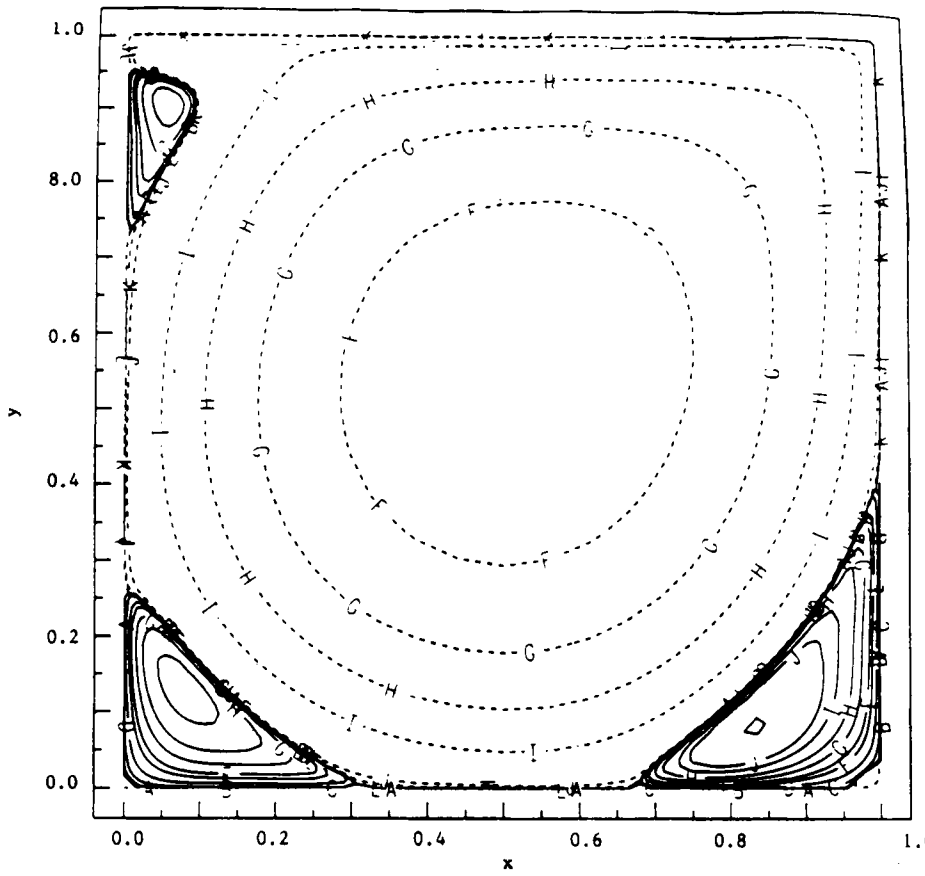

Figure 1. Streamfunction for the lid-driven cavity at $Re = 5000$: Projection 2. Contour values are:

F = −0·07	A = 0·0
G = −0·05	B = 10^{-7}
H = −0·03	C = 10^{-6}
I = −0·01	D = 10^{-5}
J = −0·0001	E = 5×10^{-5}
K = -10^{-5}	F = 0·0001
L = -10^{-7}	G = 0·00025
	H = 0·0005
	I = 0·001
	J = 0·0015
	K = 0·003

excess diffusion is *not* exclusive to SCM *nor* to semi-implicit integration *nor* to SS, but is caused by BTD itself and is not even *diffusion.*) Also, the *ad hoc* procedure of employing mixed mass matrices is seen to be legitimate at steady state; indeed, more of the error is caused by BTD at 'large' Δt than by SCM.

5.2. Vortex shedding

Our first LM failure/disappointment with vortex shedding was presented in Gresho *et al.*:[8] the LM scheme lost the eddies in a Karman vortex street because the graded mesh became too coarse

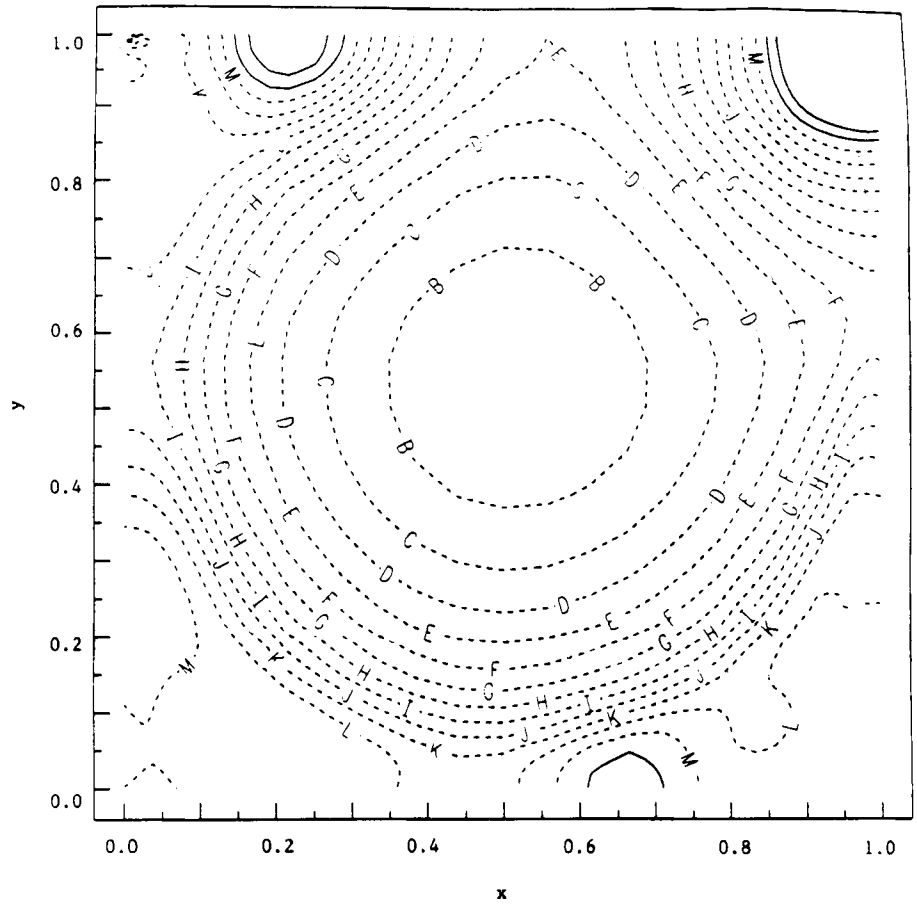

Figure 2. Same as Figure 1 except isobars. The large pressures near the upper right corner are not plotted. Contour values are:

A = −0·060	I = −0·020
B = −0·055	J = −0·015
C = −0·050	K = −0·010
D = −0·045	L = −0·005
E = −0·040	M = 0·000
F = −0·035	A = 0·005
G = −0·030	B = 0·010
H = −0·025	

in the wake, i.e. downstream of the cylinder. We return to that type of problem here to show that SCM 'delivers' where LM (still) fails. The cause of this failure is the inability of the LM four-node element to accurately advect 'short' waves; e.g. for a wave (of a Karman vortex street, say) that has a length (λ) of $4\Delta x$, the LM phase speed (relative to unity) is only $U_p^L = (\sin\theta)/\theta \simeq 0{\cdot}64$, where $\theta \equiv 2\pi\Delta x/\lambda$, whereas for CM it is $U_p^C = U_p^L \times 3/(2+\cos\theta) \simeq 0{\cdot}95$. So, based on previous results, we designed a graded mesh that forces the shed vortices to move from a fine mesh, where they are generated, into an ever coarser mesh, in which transport by advection should prevail but will finally fail; i.e. we now test SCM on a similar problem.

Because our most-used codes are largely restricted to logically regular meshes for efficient vectorization (and because our usual simulations are in the atmospheric boundary layer and do

Table III. Streamfunction extrema

	Primary vortex	Top left vortex	Bottom left	Bottom right (large)	Bottom right (small)
PPE	−0·087	+0·00071	0·0013	0·0031	$-7{\cdot}3\times10^{-7}$
Projection 1	−0·085	+0·00070	0·0013	0·0031	$-9{\cdot}0\times10^{-7}$
Projection 2	−0·088	+0·00072	0·0013	0·0031	$-8{\cdot}0\times10^{-7}$
Projection 1 with LM	−0·085	+0·00072	0·0013	0·0032	$-1{\cdot}3\times10^{-6}$
Projection 1 with LM and $\Delta t=0{\cdot}005$	−0·102	+0·0011	0·0015	0·0037	$-2{\cdot}9\times10^{-6}$
Gresho *et al.*[1]	−0·109	+0·0012	0·0015	0·0039	$-5{\cdot}2\times10^{-8}$
Ghia *et al.*[24]	−0·119	+0·0015	0·0014	0·0031	$-1{\cdot}4\times10^{-6}$

Table IV. Velocity extrema

	$U_{min}(x=0{\cdot}5)$		$V_{min}(y=0{\cdot}5)$		$V_{max}(y=0{\cdot}5)$	
	U_{min}	y	V_{min}	x	V_{max}	x
PPE	−0·376	0·074	−0·504	0·953	0·358	0·074
Projection 1	−0·377	0·074	−0·504	0·953	0·357	0·074
Projection 2	−0·379	0·074	−0·507	0·953	0·360	0·074
Projection 1 with LM	−0·376	0·074	−0·503	0·953	0·357	0·074
Projection 1 with LM and $\Delta t=0{\cdot}005$	−0·425	0·074	−0·560	0·953	0·413	0·074
Gresho *et al.*[1]	−0·426	0·074	−0·563	0·953	0·419	0·074
Ghia *et al.*[24]	−0·436	0·070	−0·554	0·953	0·436	0·078

not need 'helter-skelter' meshes), our current simulation deals with flow past a square cylinder. (All our previous work and most of all other work on vortex shedding has focused on circular cylinders.)

This less studied and probably more difficult simulation was based largely on the work of Davis and Moore,[25] who have performed what seem to be the most thorough studies—both experimental and numerical—to date. (See also Davis *et al.*[26] for related and more recent work using different boundary conditions.) Although warnings by R. W. Davis (personal communication) increased our initial concern regarding the added difficulty of computing flow past sharp corners, we still believed that such a simulation would be the easiest way to make our point, which we repeat for emphasis: we desired to see if the new method could better advect the shed vortices; we did *not* desire to embark on a detailed study of the (interesting) physics of vortex shedding past a square—and we do not.

A sequence of mesh designs—including a finer mesh than that to be shown, in which both LM and CM did a fairly good job—evolved to that shown (in part) in Figure 3. The key features of this mesh are: (1) the grid design near the (unit) cylinder is rather fine (e.g. a factor of at least two finer than that of Davis and Moore; we used 20 elements to their 10 on each face, and our minimum $\Delta x(\Delta y)$ was 0·02 *vis-à-vis* theirs of $\sim$0·10) to hopefully at least *generate* vorticity accurately—not to imply that Davis and Moore did not; (2) the elements are graded away from the square in such a way that $\lambda/\Delta x$ (where λ is measured as the 'length' of a vortex pair) would be $\sim$4 at the exit of the

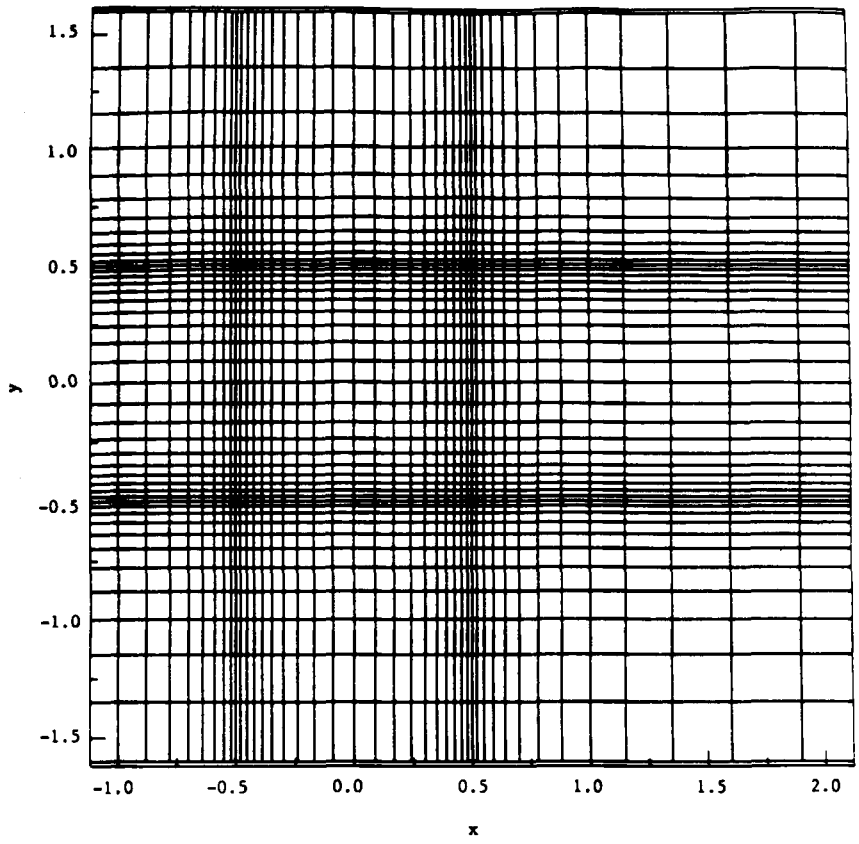

Figure 3. Details of mesh design near cylinder. The 400 (of 4898) elements shown inside the square are 'inactive' in that **u** and P are set to zero there

computational domain (in fact, $\Delta x_{\max} \simeq 1{\cdot}8$ at $x=25$); and (3) the logically regular mesh restriction is obviously not optimal for this problem since a 'too fine' mesh exists in too much of the domain.

The cylinder is centered at (0, 0) and is modelled as 'unconfined' (tow-tank simulation) with the following BCs: $\mathbf{u}=\mathbf{0}$ on the cylinder; $u=1$, $v=0$ at the inlet and at $y=\pm 6$; and $P=v\partial u/\partial x$, $\partial v/\partial x=0$ at the outlet, *á la* (1d). The latter are the natural BCs associated with our weak form of the NS equations and, for high Reynolds numbers, the first of these is nearly equivalent to setting $P=0$ at the outlet since $Re=1/v \gg 1$ in the non-dimensional case. It is to be emphasized, however, that $P=v\partial u/\partial x$ is enforced *weakly* so that the (more important) divergence-free constraint can be enforced strongly—a nice feature of the FEM, since the continuity equation is of paramount importance for incompressible flows (see GS[36] and Gresho (1991)[37]).

We remark/admit that the simulations below did not use the technique (on Γ_2) presented in Section 3.1.4 for OBCs, since we only recently derived it. Rather, the following simpler—yet still effective—procedures were employed:

(i) $\varphi=0$ on Γ_2 for Projection 1 rather than $\varphi=-TF_n(T)$ *á la* (22)
(ii) $\varphi=0$ on Γ_2 for Projection 2 rather than $\varphi=-(T/2)[F_n(T)+P_0]$ *á la* (23).

The main reason that these BCs 'worked'—i.e. did not cause us to be suspicious of the quality of the results—is that the use of $F_n=0$ saved us. Either OBC will then give $P\simeq 0$ at the outlet, and

that is what we observed. In the general case, however, with $F_n \neq 0$, the simpler BCs necessarily rely on the BHM to give decent results, whereas the preferred OBCs would give more accurate results because the BHM is then hardly needed. (For example, if Projection 1 is employed with the '$P=0$' BC in a situation where P should be 'large' (i.e. far from zero) on Γ, the BHM causes the pressure to rise from zero on Γ to the proper value 'just outside' δ.)

The Reynolds number here is 250 and the time step is 0·05 using the trapezoid rule for diffusion. This Δt gives a maximum Courant number of about 4^+ (just above the leading corner) and a maximum grid Reynolds number ($u\Delta x/\nu$) of 400–500 (near the exit).

The principal and self-explanatory result is shown in Figure 4: even *semi*-consistent mass is capable of transporting vortices through a mesh that is too coarse for the lumped mass approximation. The results for all three SCM schemes were virtually identical, and thus Figure 4 is our last pictorial comparison of all three. These and subsequent snapshots are taken at the same time, i.e. when the vertical velocity on the centreline at $x = 1{\cdot}0$ passes through zero from above. The relative streamlines (observer moving at speed 1) in Figure 5 show the individual eddy pairs and their graceful exit from the domain.

Figure 6 shows a zoom picture of the streamlines close to the box, and the corresponding velocity vectors and pressure are shown in Figure 7. It is interesting to note that at some times during the shedding cycle, fluid is actually sucked forwards along the top or bottom surface, and

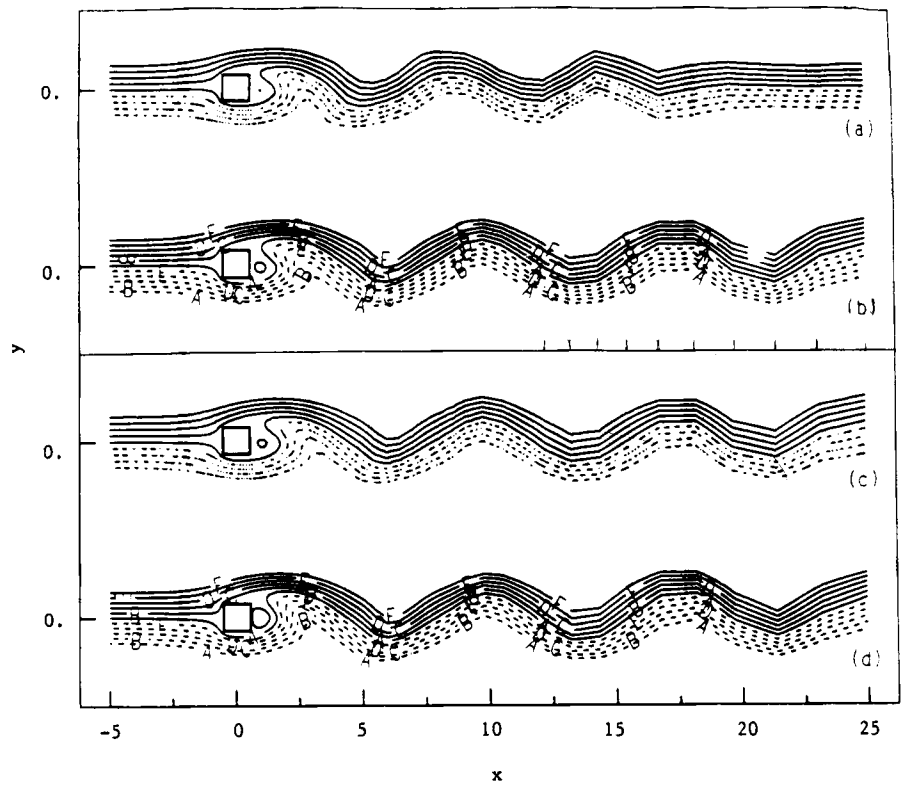

Figure 4. Snapshot of streamlines at the 'same' time for four schemes: (a) PPE with LM; (b) PPE with SCM; (c) Projection 1; (d) Projection 2. The tick marks show the lengths of the last nine elements

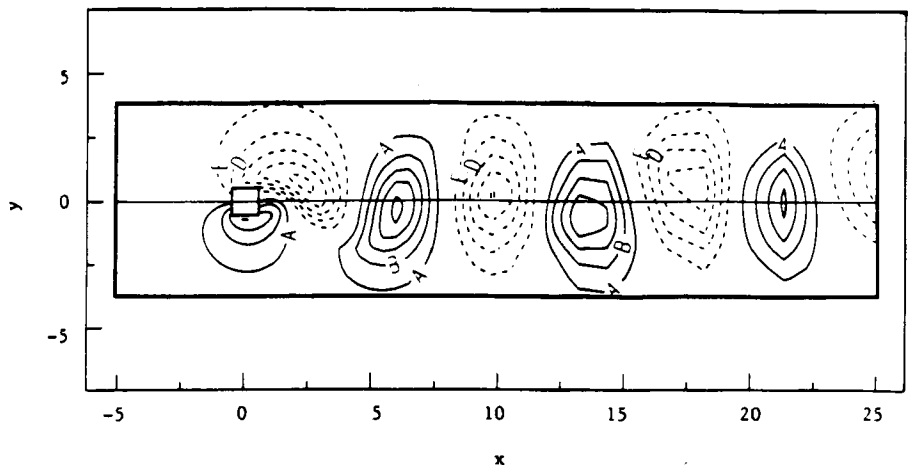

Figure 5. Relative streamlines at the same time as in Figure 4 (Projection 2). Contour values are:

A = −1·0	A = 0·2
B = −0·8	B = 0·4
C = −0·6	C = 0·6
D = −0·4	D = 0·8
E = −0·2	E = 1·0

flow separation occurs in many places. This behaviour is of course related to the extra 'difficulties' of sharp corners mentioned above.

Vorticity is shown in Figure 8 (Projection 2), where the 'shedding process' from the two leading corners is particularly clear. The maximum value of ξ is ~ 43 and occurs on the surface at the first node up from the lower left corner. The minimum value is ~ -30 and occurs at the first node down from the upper left corner. (During the course of a full cycle, of course, the maximum and minimum values of vorticity will be the same (in magnitude)—and probably occur at the same locations as above, suggesting that the locations of the singularities are on the upstream edges of the box.)

The variation of u and v with y at the upstream face of the square is shown in Figure 9. The variation of pressure with y at both upstream and downstream faces is shown in Figure 10. Both the nature of the pressure singularities and the close agreement between the two mass matrix approximations (LM and SCM) are evident in these figures. (When the mesh is fine enough, all 'legitimate' schemes 'work'. Alternatively and preferably, these results again demonstrate, or at least strongly suggest, consistency of the new mixed mass methods.)

In Table V we list a few other relevant quantities and compare them with others. The shedding period (inverse of Strouhal number) and the wavelength are based on the passing of two vortices—one shed from above and one from below the cylinder. The drag is virtually all 'pressure drag', while viscous shear stress contributes 5%–6% to the total lift. Davis and Moore[25] ignored viscous effects. Perhaps the most disturbing feature of these results is the seemingly very large lift coefficient and its wide variation from scheme to scheme, a result we have seen before for a circular cylinder.[1,8]

To help reconcile this issue, a brief independent check was made. Another FEM code (FIDAP[28]) was run to near quasi-steady conditions (wavelength of $\sim 7 \pm 1$ and still changing) using a similar mesh and one of the best 2D elements for incompressible flow: nine-node

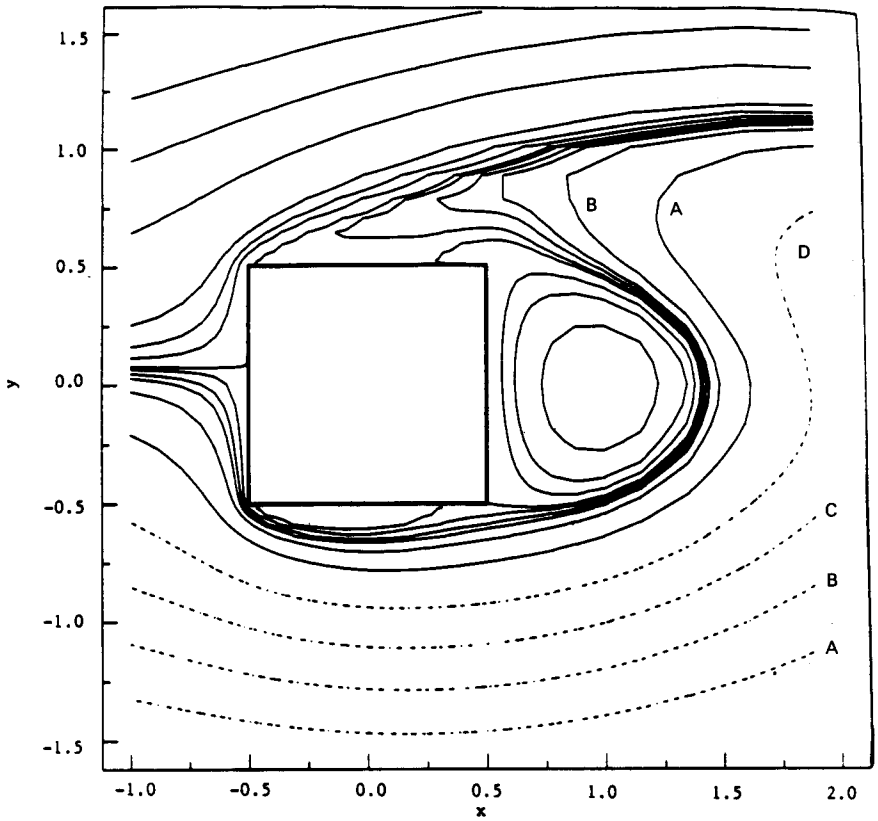

Figure 6. Streamlines close to cylinder (Projection 2). The separation streamline is *f*. The contour levels are:

A = −1·00	E = 0·15
B = −0·75	F = 0·1556
C = −0·50	G = 0·175
D = −0·25	H = 0·20
A = 0·00	I = 0·25
B = 0·10	J = 0·50
C = 0·13	K = 0·75
D = 0·14	L = 1·00

biquadratic velocity with three-node linear pressure (and consistent mass). The results were rather comforting even though not really conclusive; a spot check (at one particular time during the shedding cycle) gave a lift coefficient of about 1·3. It is also worth mentioning that Davis and Moore[25] found that the lift coefficient varied markedly with Reynolds number. Incidentally, the FIDAP code predicted a shedding period of ~7.

A brief cost summary is as follows: for one shedding cycle (~150–160 time steps at $\Delta t = 0{\cdot}05$) the CPU cost was ~35 s and the I/O cost was ~6 s on a CRAY-1S.

The differences in results from ostensibly equivalent SCM schemes and from the two 'equivalent' LM schemes are not easy to explain, but we believe they are related to the overall difficulty of this simulation and suggest that asymptotic analysis does not yet apply—we do not have converged solutions in either space or time. Also, and unfortunately, the extreme sensitivity of the results obtained to the schemes used, combined with a lack of knowledge of the right answer,

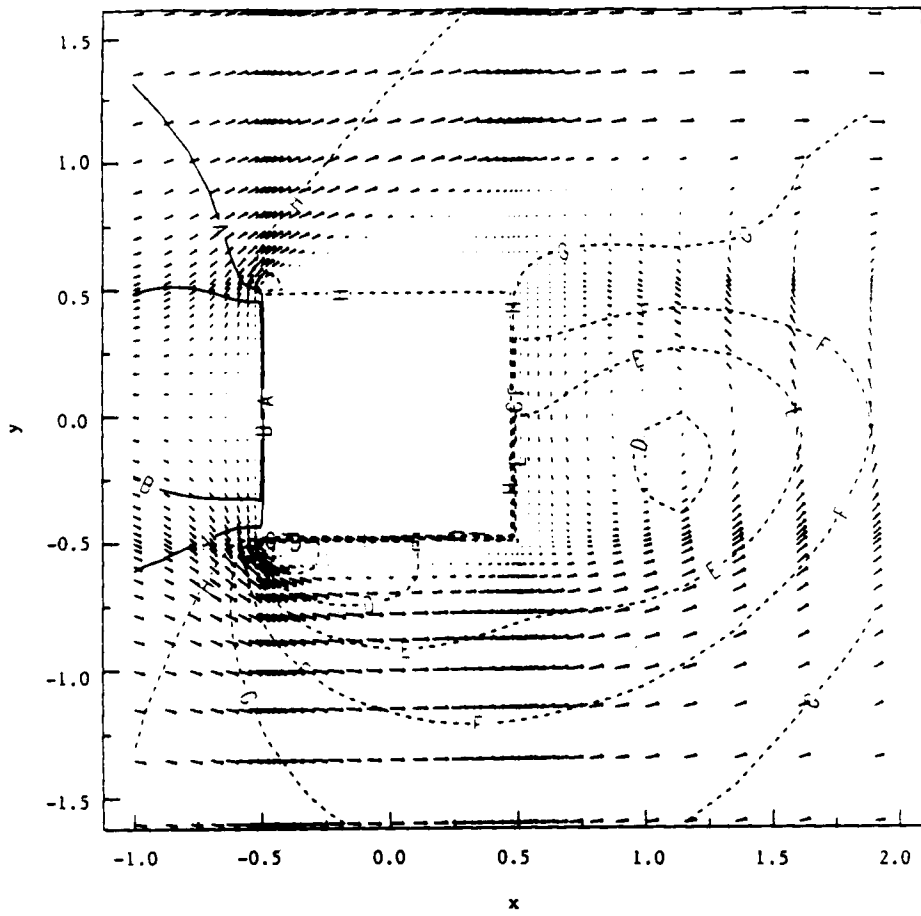

Figure 7. Vectors and isobars near cylinder. The contour levels are:

A = −1·650	G = −0·2527
B = −1·417	H = −0·0197
C = −1·184	A = 0·2132
D = −0·9514	B = 0·4461
E = −0·7185	C = 0·6790
F = −0·4856	

leaves us unable to even estimate the relative accuracy of the methods. The next test problem was designed to help fill in these (and other) gaps regarding comparison of the schemes.

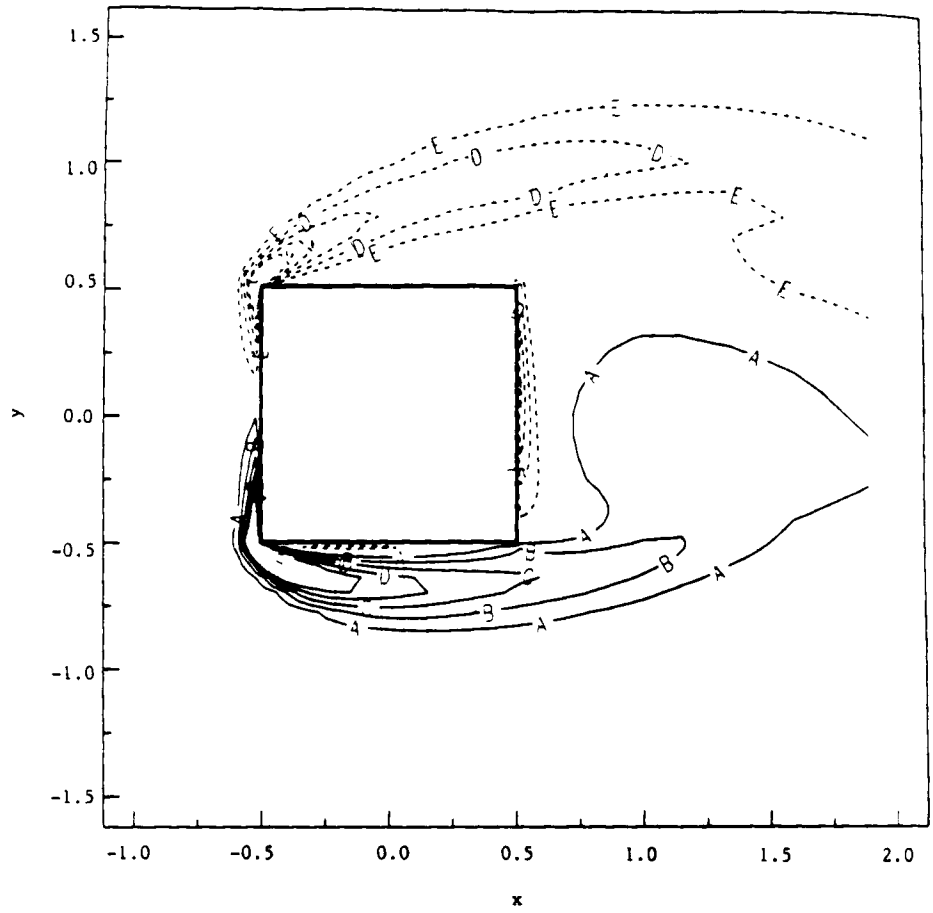

Figure 8. Vorticity contours close to cylinder. Extreme values near the leading corners are not plotted. The contour levels are:

A = −10·0	A = 2·0
B = −8·0	B = 4·0
C = −6·0	C = 6·0
D = −4·0	D = 8·0
E = −2·0	E = 10·0

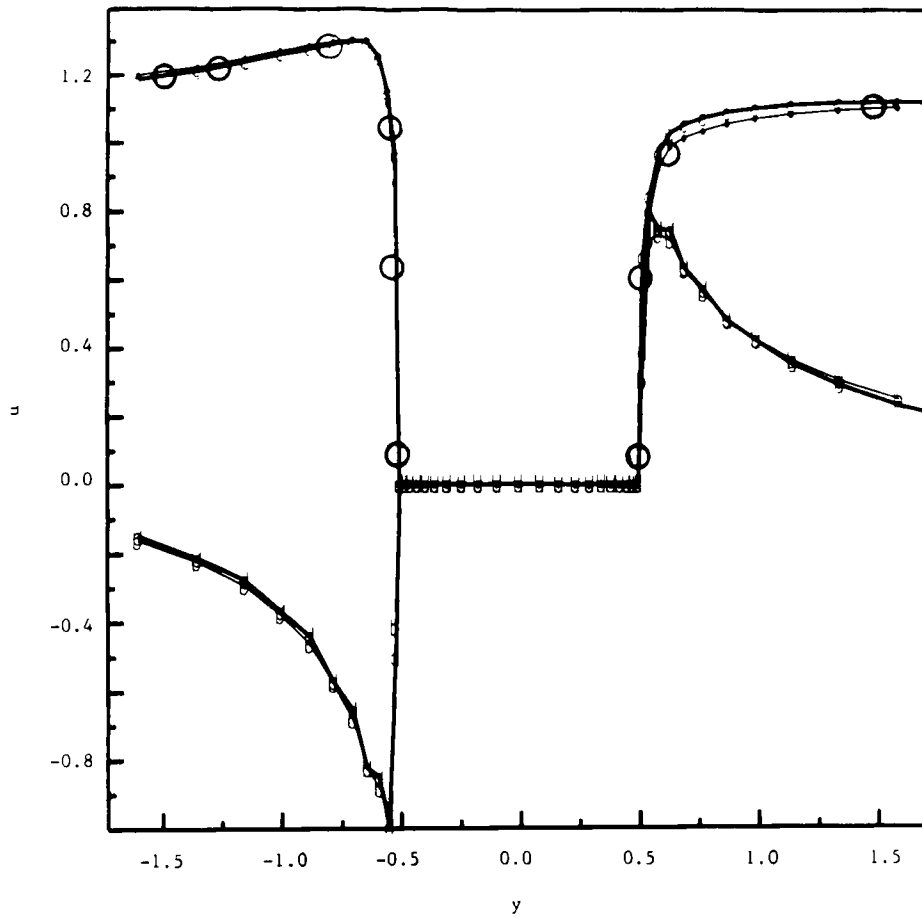

Figure 9. Horizontal (circles) and vertical velocity versus y at $x=-0{\cdot}5$, from Projection 1. Both SCM and LM (heavy lines) results are shown

wherein a Burgers' vortex (related to a Lamb vortex) is actually used as a model of a dust devil), we settled on a similar and *perhaps* simpler (but more discontinuous) vortex of compact support that we call a triangle vortex. Like a Rankine vortex, it has a core of solid body rotation (constant vorticity). But at $r=R$ we switch to a decreasing linear function of r until $r=2R$, where the tangential velocity returns to zero. Specifically, our vortex IC is (before adding the translational velocity)

$$u_\phi(r)=u_0 r/R \quad \text{for } 0\leqslant r\leqslant R,$$
$$u_\phi(r)=u_0(2-r/R) \quad \text{for } R\leqslant r\leqslant 2R,$$
$$u_\phi=0 \quad \text{for } r>2R,$$

where $u_0=1$.

Figure 11 shows this initial velocity field as well as the corresponding vorticity, $\zeta\equiv(1/r)(\partial/\partial r)(ru_\phi)$, and pressure, $P(r)=\int_0^r u_\phi^2\,\mathrm{d}z/z+\text{constant}$, in which the constant was selected to give $P=0$ for $r\geqslant 2R$. The two discontinuities in $\zeta(r)$ make the problem 'interesting'. We used a

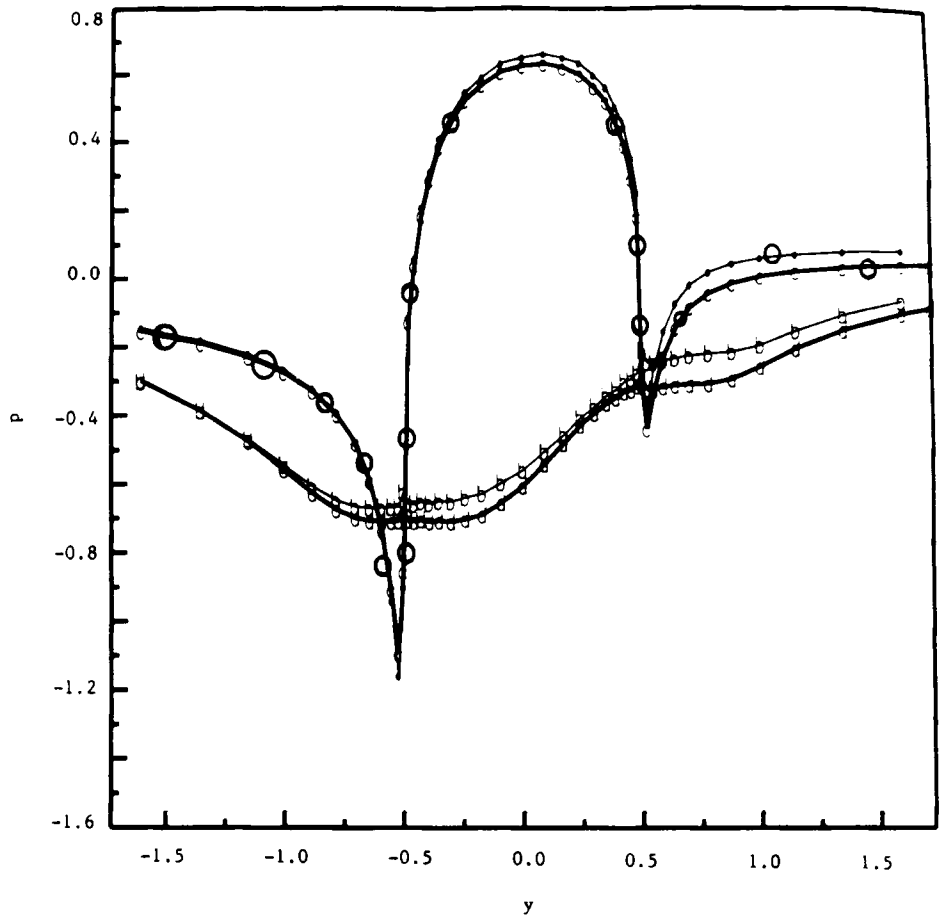

Figure 10. Pressure versus y at $x=-0{\cdot}5$ (circles) and $x=+0{\cdot}5$. Again, SCM and LM (heavy lines) are plotted

Table V. Some global quantities associated with vortex shedding

Source	Shedding period	Wavelength	Drag coefficient	Lift coefficient
PPE	7·8	7·7	1·85–2·12	−1·40 to 1·40
Projection 1	7·7	7·7	1·79–2·05	−1·32 to 1·32
Projection 2	7·9	7·5	1·95–2·39	−1·75 to 1·75
Projection 1 with LM	7·2	5·6[a]	1·86–2·08	−1·20 to 1·20
PPE with LM	7·7	7·1[a]	1·86–2·26	−1·50 to 1·50
Davis and Moore[25]	5·9–6·3[b]	5·9	1·73–1·81	−0·50 to 0·50
Okajima[27]	7·1[c]	6·2[d]	—	—

[a] Rather variable owing to poor grid resolution.
[b] Numerical (several meshes) and experimental.
[c] Experimental value.
[d] A crude estimate from his Figure 10(b).

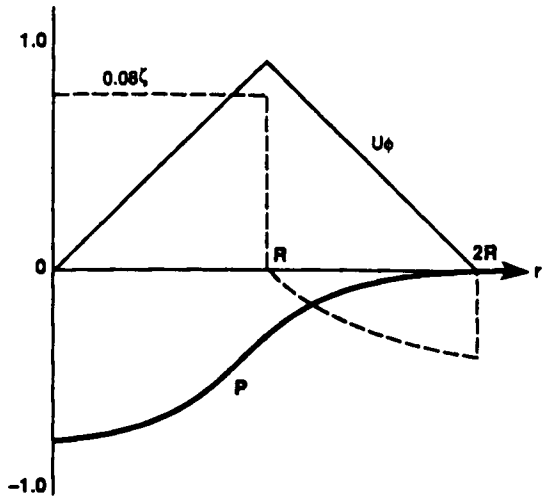

Figure 11. Initial conditions for the triangle vortex problem

uniform mesh of 80 x 20 elements to cover a domain that is one unit high and four units long. The (full) radius of the triangle vortex is $8\Delta x$, i.e. $4\Delta x(=R)$ for u_ϕ to rise from 0 to 1 and $4\Delta x$ to come back down. The initial velocity is shown again in Figure 12, centred at $x=y=0{\cdot}5$, as a vector field in our (x, y) co-ordinate system. (In fact, however, the initial velocity shown in Figure 12 is the discrete *projection* of the analytical $u_\phi(r)$, so that (43c) is satisfied.)

We 'solved' this problem using four methods (PPE, Projection 1, Projection 2 and PPE/LM) many times at many different Δt's. The principal results are these: (1) for $C \leqslant \sim 0{\cdot}1$ ($C \equiv u\Delta t/\Delta x$ with $u=1$ and $\Delta x = 0{\cdot}05$, and we point out that $C_{\max} \simeq 2C$ since the maximum speed is ~ 2), the three SCM schemes delivered results that were so close to each other that they appeared graphically to be the same (Projection 2 results are shown below); (2) Projection 1 and Projection 2 agreed even more closely; (3) the LM results are not very accurate; (4) the spurious damping caused by BTD is 'too large'—a point we shall return to towards the end.

Figure 13 shows the relative streamlines as the vortex moves through the mesh for $C=0{\cdot}1$ (the results for smaller values of C look essentially the same, whereas those at larger values show additional phase lag and distortion owing to time truncation error.)

Virtually no phase lag is present using SCM, and both results (especially LM) show a 'falling' vortex, a result that seems spurious and whose cause is not obvious—it is probably related to the asymmetry in initial conditions (reversal of the initial circulation direction causes the vortex to rise) combined with truncation error (a larger Δt accentuates the effect), perhaps explainable by appeal to group velocity analysis/argument (in which LM evolves as particularly poor). The last snapshot ($t=3{\cdot}5$) is shown to permit some assessment of the utility of the (natural) outflow boundary conditions.

The (theoretically discontinuous) vorticity field evolution is shown in Figure 14. Whereas the SCM schemes capture and translate the initial field reasonably well, that from the LM method is noticeably degraded even by $t=1$. For $t>1$, the spurious trajectory, with resulting boundary interaction, further distorts the vorticity field, although the trailing wiggles, as well as the phase lag, are caused mostly by dispersion error.

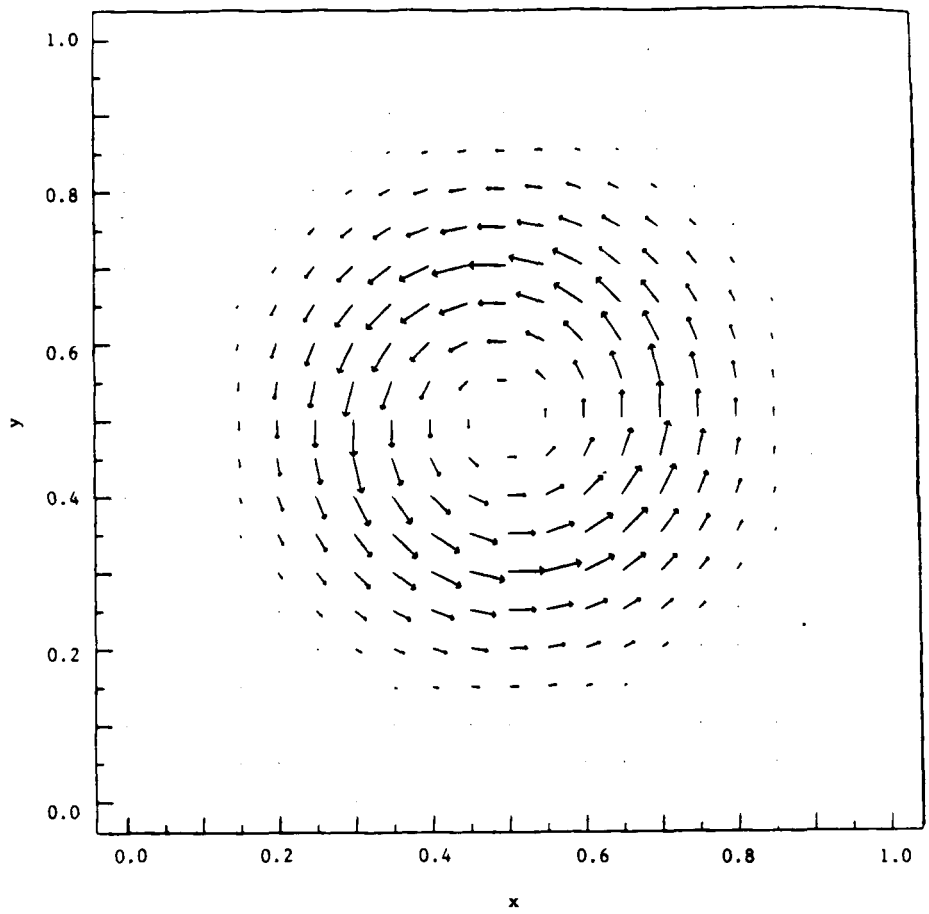

Figure 12. Initial triangle vortex velocity field

The corresponding pressure fields are shown in Figure 15. Again, the SCM has done a markedly better job of preserving the size and shape of the pressure 'low'. The interaction of the isobars with the outflow boundary is interesting, since the boundary conditions are trying to set $P=0$ there. A 'totally unbelievable' pressure field has a surprisingly small effect on the velocity field—especially for SCM.

Finally, Figures 16 and 17 provide a different perspective on the flow field and reveal the asymmetry mentioned above. The vortex now clearly appears as a perturbation to a uniform flow.

Having demonstrated the main point (the 'good news') to be made regarding SCM *vis-à-vis* LM—and we believe but did not demonstrate that a fully GFEM (CM) simulation would not differ much from SCM, at least with respect to vortex size, shape and location—we now turn to the 'bad news' portion of this research project alluded to earlier: the BTD technique for the (vector) equations of motion seems to be definitely inferior in its performance to that for which the method was originally derived—the (scalar) advection–diffusion equation.[1,2] And it took some years and a special and difficult test problem to bring this deficiency to light; i.e. numerical solutions to the incompressible Euler equations, in which the initial vorticity field is discontinuous, would be

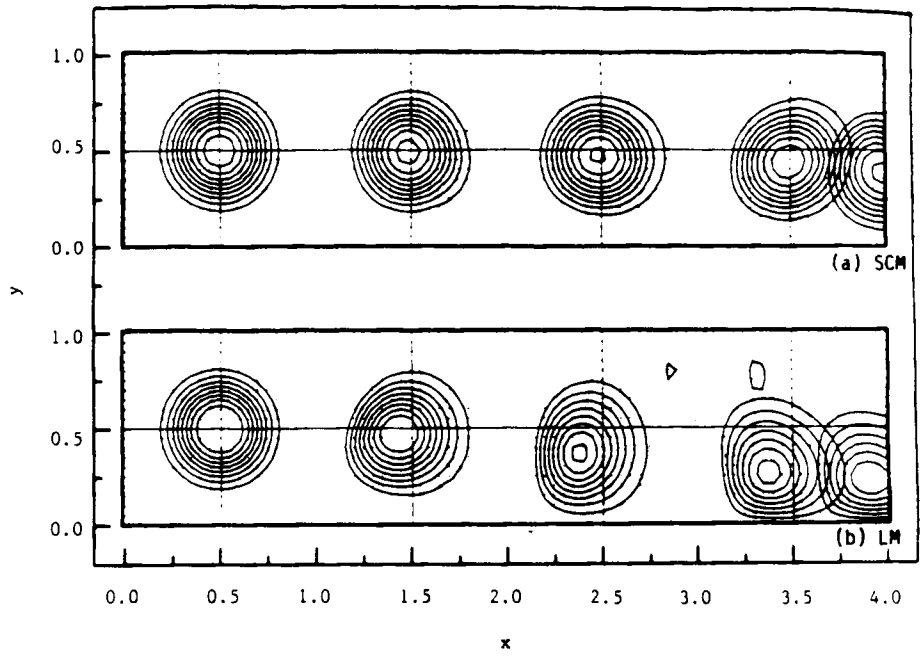

Figure 13. Relative streamlines at $t=0$, 1, 2, 3, 3·5; $\psi=0{\cdot}02(0{\cdot}02)0{\cdot}18$

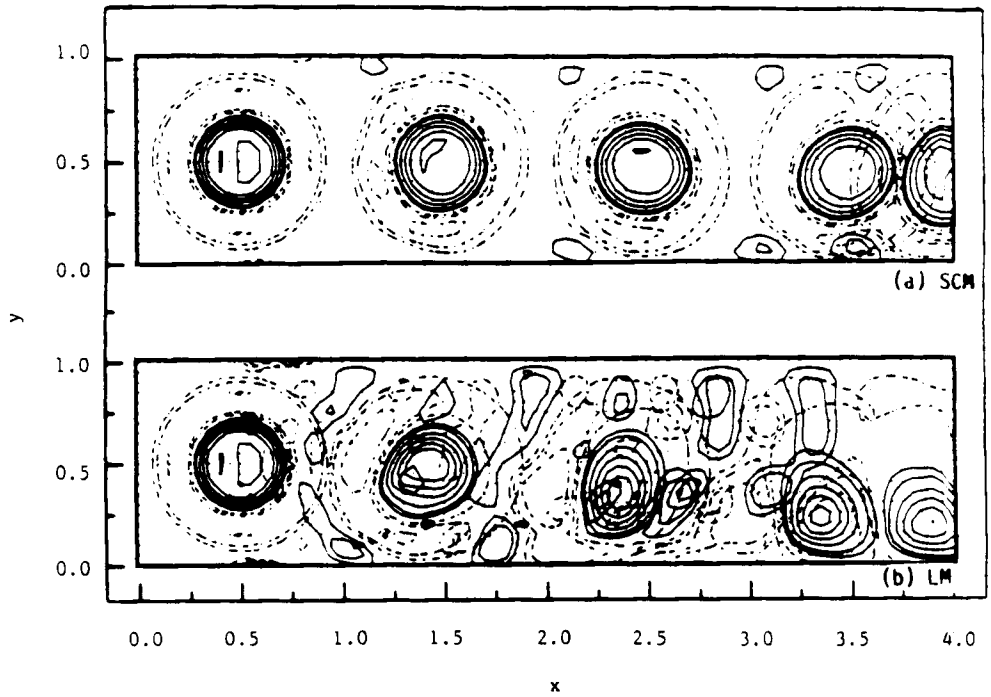

Figure 14. Vorticity contours at the same times: $\xi=-4$, -2, -1, 1, 2, 4, 8, 10

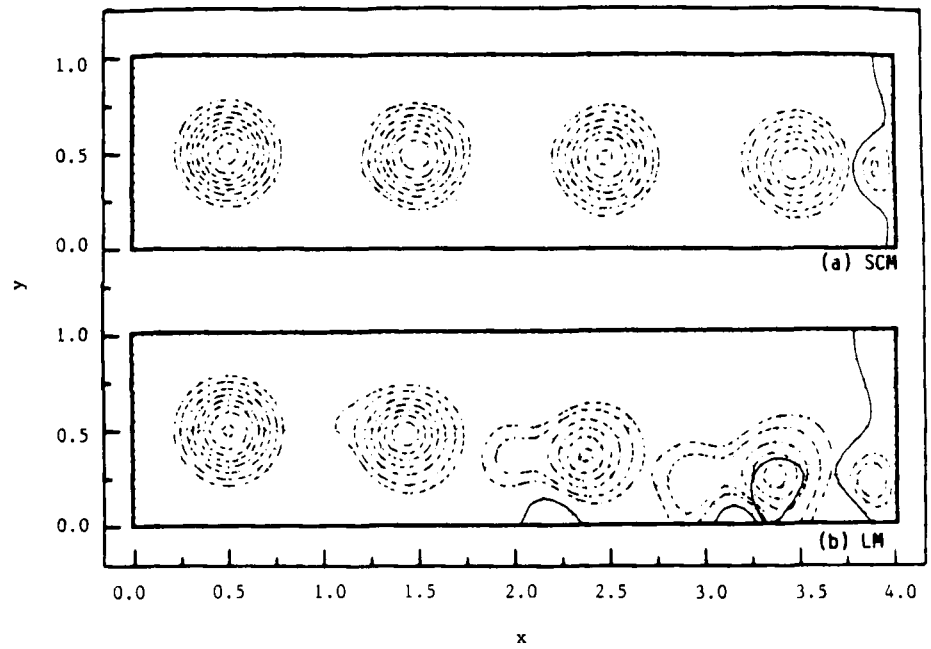

Figure 15. Pressure contours at the same times; $P = -0{\cdot}7(0{\cdot}1) - 0{\cdot}1, \pm 0{\cdot}05$

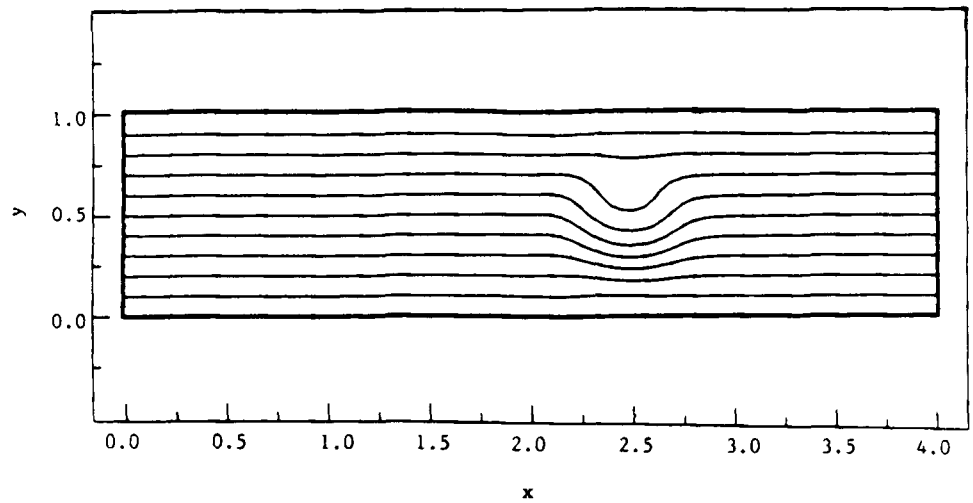

Figure 16. Absolute streamlines at $t = 2$: SCM

expected by few to be easy. Previous simulation of 'more reasonable' problems showed much smaller (and usually tolerable) deleterious effects from spurious 'diffusion'. The problem is perhaps best revealed in Figure 18, showing the serious lack of conservation of vortex kinetic energy (defined here as $E_L = u^T M_L u/2$). The most striking and most disappointing result is the rapid decay of E_L for all but very small values of C (except for LM at $C = 0{\cdot}01$, an anomalous result that we cannot explain, although the indefinite advection operator combined with group velocity errors is

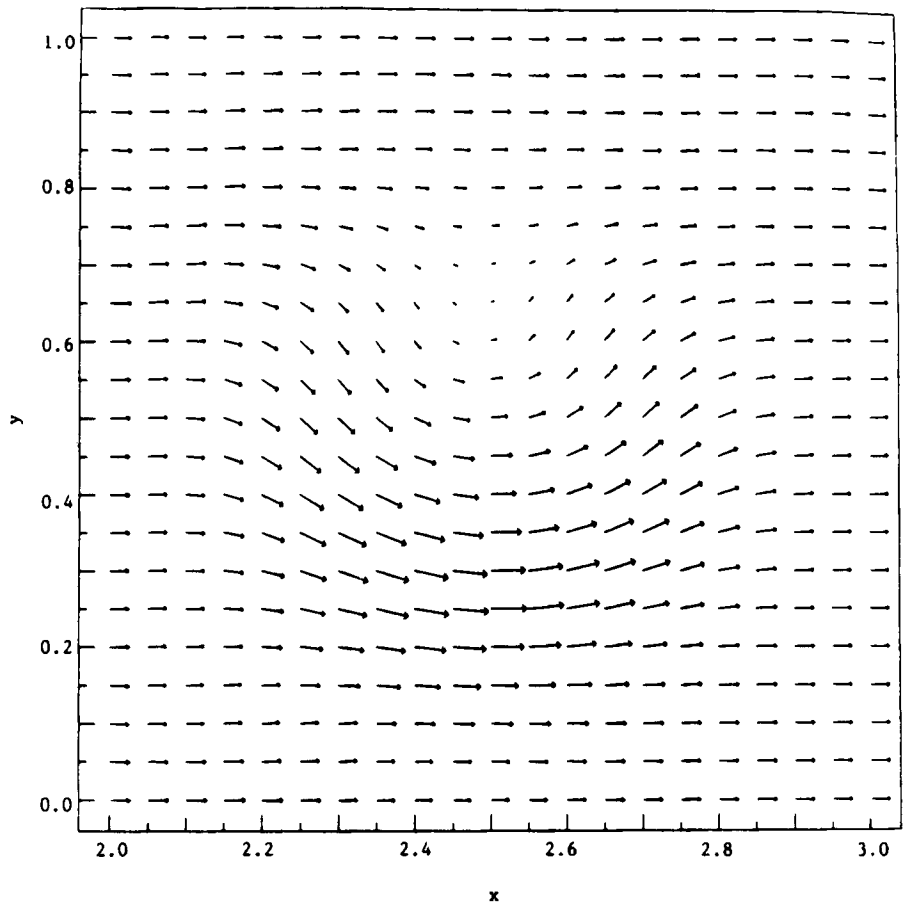

Figure 17. Velocity vectors at $t=2$: SCM

a likely candidate); indeed, the decay of E_L using SCM is *monotonic* at all values of C studied. In the earlier figures the decay is present though not so apparent; e.g. the maximum in ψ at $t=3$ in Figure 13(a) is only 89% of that at $t=0$, whereas for $C=0{\cdot}01$ it is 95%, a mere 6% gain for a factor-of-10 increase in cost. (We need not be reminded that any scheme that requires $C \ll 1$ to deliver acceptable results is not a very good scheme.)

Before expanding further on this negative virtue of BTD, let us review and summarize the additional salient tests that we made before becoming fully convinced of the severity and veracity of the problem.

(1) Two independent GFEM codes—FETISH (our own) and FIDAP (FDI's)—were run on a closely related problem (as was our new code and our old explicit code *à la* Gresho *et al.*:[1] *pure rotation* (circular streamlines) of the same vortex (in a unit square domain) with, of course, free-slip BCs). The results, even using the dissipative backward Euler (BE) scheme for time integration (and wiggle suppressant), showed that BTD is relatively quite 'dissipative'; i.e. BE alone ($C=1{\cdot}0$) was *much* less dissipative. In one experiment (courtesy of M. S. Engelman of FDI) using FIDAP—which has the flexibility of testing all sorts of combinations—the GFEM equations were integrated via BE, *and* the BTD 'flag' was activated (a

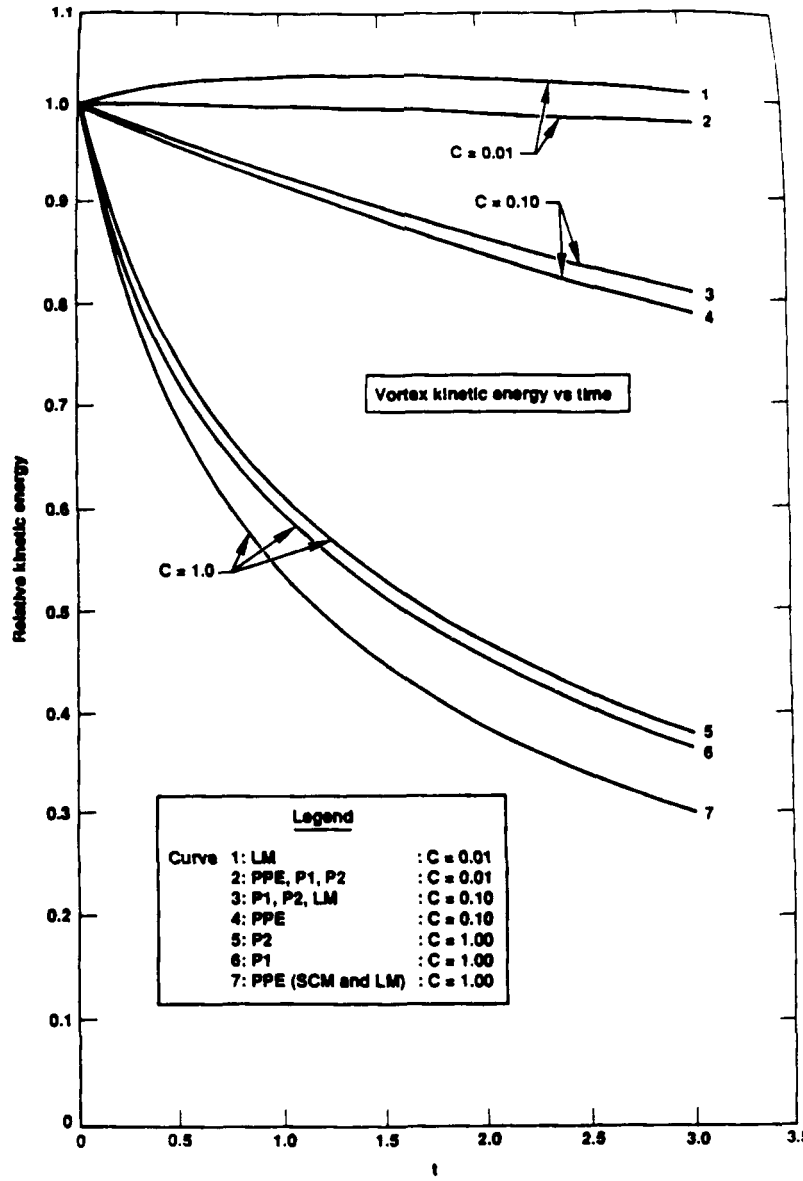

Figure 18. Vortex kinetic energy versus time

procedure that adds the tensor viscosity, $\Delta t u_i u_j/2$, to the physical value and is theoretically justified/required *only* for FE). The results were *nearly the same* as those using explicit Euler and BTD (wherein BTD is required to stabilize the time integration). Since we also obtained similar results using our FE/BTD code, we concluded that the key problem is the *use* of BTD—not in the time integration scheme, since explicit, semi-implicit and implicit all behave similarly if BTD is included. In fact, another BE FIDAP experiment was performed (by M. S. Engelman) in which BTD was *subtracted*—a result that should 'theoretically' make

the scheme less dissipative and neutrally stable: for $C=1{\cdot}0$ the calculation was unstable, whereas for $C=0{\cdot}1$ it was both stable and only very slightly dissipative.

(2) We solved several pure advection (scalar transport) problems using the given (triangle vortex) velocity field and learned that the original BTD theory is then vindicated: spurious dissipation is nearly absent if streamline gradients are too (as in the pure rotation case), even when SS conditions are attained. Simulations in which streamline gradients occur (the general case) are, however, slightly dissipative—in accordance with 'BTD theory' presented in Gresho *et al.*[1] and Gresho and Chan.[2] The decay rate is, however, *much* less than that for the velocity field when Euler solutions are attempted.

(3) We solved the advection equation simultaneously with the NS equations for the pure rotation case. The IC was a top-hat profile: constant from $r=0$ to $r=4\Delta x$, zero thereafter. Results showed good conservation of the passive scalar while the advecting velocity field was weakening quickly and monotonically (and would, it seem, eventually come to rest).

(4) Finally, Figure 19 shows the results when a passive scalar (say S) is added to the inviscid translating vortex problem—using CM in the scalar advection equation. The IC was the same as that given for u_ϕ described earlier—think of a ring of ink in the (frictionless) water with a peak concentration at the annulus centre (i.e. at $r=R=4\Delta x$)—the peak of the concentration triangle coincides with the peak rotational speed. While not a particularly exciting (i.e. accurate) result—the sharp peak has levelled off and the contour lines are rather distorted—it *is* true that the 'energy' of S (i.e. the quadratic form $S^{\mathrm{T}}M_{\mathrm{L}}S$) decays slowly ($<3\%$ loss for $0<t<3$ with $C=0{\cdot}1$) during the simulation, in marked contrast to the kinetic energy of the fluid, which (Figure 18) dropped by almost 20% at the same value of C. The distorted S-contours relative to the better-*looking* contours of ψ (Figure 13(a)) and ζ (Figure 14(a)) seem to be a direct consequence of the extra 'diffusivity' (and wiggle suppressant) that BTD bestows upon the water but not upon the ink; i.e. the Euler simulation *should* be even more difficult than portrayed by the BTD schemes.

The inevitable conclusion from these experiments and further analysis, summarized below, is that the BTD 'correction', derived and justified for advection–diffusion, does not *simply*

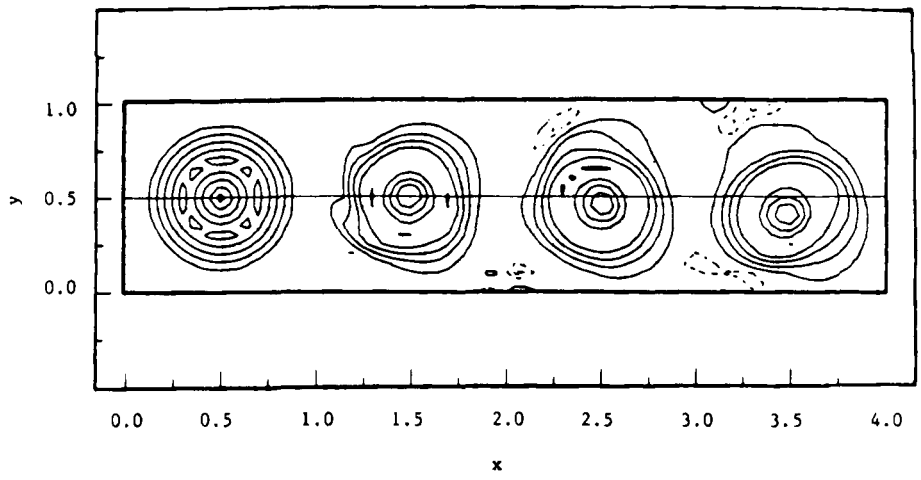

Figure 19. Contours of a passive scalar, corresponding to the flow field given in Figure 13(a). Contour values: $-0{\cdot}3$, $-0{\cdot}2$, $-0{\cdot}1$, $0{\cdot}1(0{\cdot}2)0{\cdot}9$

'transplant' to the NS equations as we (and others) have naively assumed. The explanation of this theoretical *faux pas*, while not yet fully verified, seems to be the following: whereas the (unmodified) explicit Euler scheme does *indeed* introduce a negative (and destabilizing) but known tensor diffusivity directed along streamlines for the (scalar) AD equation, which *needs* to be 'balanced' via BTD, such is *not* the case for the (vector) NS equations; in fact, the portion of the local error from the advection term that 'looks like' (streamline) diffusion may even be cancelled by a similar truncation error related to the pressure gradient. Another portion of the explanation, and one that accounts for the monotonic decay of kinetic energy in the case of a purely rotational inviscid vortex, is this: since the BTD term is added *separately* and *independently* to each (Cartesian) component of the NS equations (u and v), it follows that dissipation *must* occur unless *both* u *and* v are constant along streamlines.

To obtain further insight, let us transform the 2D Euler equations, with BTD,

$$\frac{\partial \mathbf{u}}{\partial t}+\mathbf{u}\cdot\nabla\mathbf{u}+\nabla P=\frac{\Delta t}{2}\nabla\cdot\mathbf{u}\mathbf{u}\cdot\nabla\mathbf{u}=\frac{\Delta t}{2}(\mathbf{u}\cdot\nabla)^2\mathbf{u},$$

to intrinsic co-ordinates—i.e. to a co-ordinate system that follows the (instantaneous) streamlines, s representing the co-ordinate along a streamline and n the (crosswind) co-ordinate normal to it the result is, in the streamline direction,

$$\frac{\partial q}{\partial t}+q\frac{\partial q}{\partial s}+\frac{\partial P}{\partial s}=\frac{q^2\Delta t}{2}\left(\frac{\partial^2 q}{\partial s^2}-\kappa^2 q\right),$$

where $q\equiv|\mathbf{u}|$ and κ is the principal curvature of the streamline. Since $(q^2\Delta t/2)$ looks like a diffusion coefficient, the term $(q^2\Delta t/2)\partial^2 q/\partial s^2$ *does* look like streamline diffusion (and would vanish if q were constant along a streamline). But the remaining term, $-(q^2\Delta t/2)\kappa^2 q$, represents pure *damping* (not via diffusion) and, for $\kappa\neq 0$, would cause a *monotonic decay* of q. For the circular streamlines of the pure rotation case this term is $-\Delta t q^3/2r^2$, where r is the radial distance from the vortex centre. Since $q=|u_\varphi|$ for this (cylindrical co-ordinate) case, and $\partial q/\partial s=\partial^2 q/\partial s^2=\partial\rho/\partial s=0$, we get the simple result

$$\partial u_\varphi/\partial t=-\Delta t u_\varphi^3/2r^2,$$

showing *monotonic decay* of streamline velocity. If we can be permitted to integrate this equation in time for a fixed r, we obtain

$$u_\varphi(r,t)=u_\varphi(r,0)/\sqrt{[1+\Delta t u_\varphi^2(r,0)t/r^2]},$$

a result that, unfortunately (!), describes rather well the observed behaviour in the case of pure rotation.

This result, which is the *cause* of the KE degradation, is actually a consequence of losing tensorial invariance when transforming the equations: the 'BTD' term in the ϕ-equation is a direct consequence of a misrepresentation of the curvature terms (or Christoffel symbols) caused by $(\mathbf{u}\cdot\nabla)^2 u$ and $(\mathbf{u}\cdot\nabla)^2 v$ during the transformation (it is the u_ϕ^2/r curvature term which 'carries through' the transformation process and pollutes the final result). Thus the BTD problem is a *curvature* error resulting from the way it was applied (or mis-applied) to the vector momentum equation.

In any case, the *obvious* 'fix' seems, just as obviously, too expensive (but further effort may prove otherwise): i.e. (i) rotate the $x-y$ equations, at each node and at each time step, to streamwise and crosswind co-ordinates ($\tan\theta=v/u$, etc.), (ii) add $\Delta t|\mathbf{u}|^2/2$ to the *streamwise* momentum equation (only) and (iii) transform back to the Cartesian equations. This procedure—or one equivalent thereto—seems to be the only way to preclude the spurious effect in the general case. And, as of this writing, it appears to be too expensive to contemplate seriously. One should probably either

accept the BTD curvature crisis or design a better scheme—perhaps based on leap-frog for advection, or one of the Taylor–Galerkin methods proposed by Donea *et al.*,[32] or on one of the TWS (Taylor weak statement) schemes of Baker and Kim.[33] A final remark: the omission of BTD renders the Euler equations unconditionally unstable.

Just prior to mailing this paper, we received a report by Tezduyar *et al.*[34] in which the standing inviscid triangle vortex was computed in several ways, old and new, with the following results.

(1) The KE is conserved using a streamfunction–vorticity formulation—thus adding further credibility to our explanation since the ψ–ω method involves a scalar transport equation.
(2) The KE degradation of the SUPG (streamline-upwind Petrov–Galerkin) scheme of Hughes and Brooks is quite similar to that from our BTD scheme.[35]
(3) By designing special (and in our view, expensive-looking) three-and six-step schemes in which the 'SUPG supplement' is judiciously applied, the former was a noticeable improvement but still rather dissipative, and the latter retained the KE with virtually no loss.
(4) A 'Galerkin formulation' was unsuccessful (wiggly), in contrast to the results that we obtained with the FETISH code and that M. Engelman obtained with the FIDAP code.

Finally, returning to Figure 18, one method of ranking the three SCM schemes presents itself: Projection 2 decays (slightly) more slowly than Projection 1, which decays more slowly than PPE. Another ranking, not obtained from Figure 18, relates to stability and is this—for the same translating vortex problem: Projection 1 is (very slightly) more stable than Projection 2, and both are more stable than PPE—by a factor of several. For example, $C \leqslant \sim 4$ is stable for the first two, but PPE required $C \leqslant \sim 1$. (LM (PPE) is stable for $C \leqslant \sim 3$.)

6. CONCLUSIONS

In the first part of the paper we have derived, reasonably carefully, several types of projection method approximations to the solution of the time-dependent incompressible Navier–Stokes equations. Hopefully the presentation is useful—if for no other reason—by virtue of its attempt to remove the veil of mystery that has always enshrouded these methods. Careful attention to boundary conditions and other details has led to methods that are optimal in the sense of achieving low slip velocity (vortex sheet generation) and higher regularity near boundaries. Simpler and probably more cost-effective schemes were also proposed and justified practically via numerical results, and theoretically via showing the inherent satisfaction of higher-order equations that exhibit boundary layer behaviour in such a way that correct results are realized everywhere except within this (spurious) boundary layer, whose thickness can and should be kept small via $\Delta t \ll l^2/\nu$, where l is any relevant physical length scale. These simpler schemes inject wall vorticity in a different but legitimate manner.

All projection methods studied herein generate a spurious boundary layer in which some aspects of the (inner) solution deviate substantially from those of the Navier–Stokes equations. Away from this boundary layer, however, the (outer) solution from these methods will usually agree well with that from the Navier–Stokes equations.

We make no claim regarding elegant mathematics; quite the converse in many respects. What we *do* claim is to have evoked a new *understanding*, hopefully sufficiently well communicated and with few enough errors that (i) these methods will be further developed and improved and (ii) better mathematics—by others—will follow.

Based on current knowledge, the recommended scheme (at least via finite element implementation) is the simpler version of Projection 2, via a second-order-accurate time integration scheme, in which *all* Dirichlet boundary conditions—intermediate and final (projected)—are those that apply to the true Navier–Stokes equations. (We also derived appropriate—and

simple—boundary conditions for outflow boundaries.) If, however, a good approximation of the time rate of change of the tangential pressure gradient on Γ were somehow made available, then a mixed BC approach may be better; i.e. use the simple BC in the normal direction and the optimal BC in the tangential direction(s). On the other hand, the sheer simplicity of the simple version of Projection 1, in which a first-order time integration scheme is probably a proper match, still has much to recommend it, or at least to not condemn it—although the time step should probably be smaller than that for Projection 2.

If projection error could be separated from time integration error and a proper subcycling strategy designed, it is possible that a version of the (simpler) Projection 3 method would be even better than Projection 2, which would of course also benefit from these improvements. Effort in this area is recommended, including more work on the accumulated effects of pressure updates—i.e. global analysis.

In the actual computer implementation of projection methods via temporal and spatial approximations, semi-implicit techniques were developed and are advocated in which the implicit treatment of the viscous term entails the appeal to a biharmonic equation in order to describe how and why these techniques actually work in practice. (If explicit methods are employed for the viscous term, there is no mystery (or should not be), no spurious boundary layer and no higher-order equations to be satisfied, only a (frequently severe) stability restriction on the time step—and a large vortex sheet (plus other errors, not limited to explicit methods) if the projection is made too infrequently; e.g. via subcycling.)

In the 'finite element' portion of the paper, Part 2, we have described, partially analysed, and demonstrated three schemes that retain the consistent mass matrix in the Navier–Stokes equations in a cost-effective way relative to using consistent mass everywhere; i.e. by lumping the mass only in the pressure gradient term and solving sequential and uncoupled systems in the context of semi-implicit time integration (implicit for diffusion, explicit for advection) and a Poisson equation. All three delivered nearly the same results and for nearly the same cost as a semi-implicit lumped mass scheme. For the problems studied, all three have been shown to be superior to lumped mass methods for flows in which advection accuracy is important, and to produce only small errors at steady state. While a relative ranking is difficult, the first of these, PPE, seems to be the worst in that it shows more disadvantages than advantages: (i) it displays an inconsistency at $t = 0^+$; (ii) it delivers non-divergence-free solutions that are both larger than those from the equivalent lumped mass scheme and non-vanishing at steady state; (iii) it is slightly less stable; (iv) it is slightly more expensive and slightly more dissipative. Projection 1 has the disadvantage that even SS results depend on Δt. While Projection 2 has clear theoretical advantages over Projection 1, in practice the two schemes have thus far been so close that the simplicity and (very slightly) reduced cost of Projection 1 would seem to favour it. Nevertheless, we still prefer and recommend Projection 2, using the mixed-mass matrices when advection dominates and good phase speed is required. (The lumped mass Projection 2 scheme may be better than mixed mass under certain conditions, such as Stokes flow or for simulations wherein only the steady state results are of interest.) Projection 3 is probably worthy of further development, and both higher-order methods would benefit from a good subcycling strategy.

The concept of BTD (balancing (?) tensor diffusivity) has been shown to be less righteous for the incompressible flow equations—a vector system—than for the scalar transport equation for which it was originally derived (and still seems well justified). This problem, uncovered quite by accident during this research, appears to be manifested via spurious but non-diffusive damping of the streamwise velocity component, owing to a misrepresentation of the curvature (centrifugal force) term that is proportional to Δt and does not seem easy to remove. This deficiency would seem also to be present in at least some streamline-upwind methods currently in vogue.

If BTD is retained as is, rather small time steps would seem to be required in order to reduce the spurious dissipation to acceptable levels—at least for very high Reynolds number flows. Better schemes should probably be used. It should also be emphasized that the BTD curvature crisis in no way affects the other conclusions reached in this study.

Finally, the mixed mass matrix concept, 'demonstrated' experimentally to *be* legitimate, must be so because MM_L^{-1} is close to the identity matrix in the appropriate sense and on adequate meshes. While it is never *spectrally* equivalent for *any* Δx, it seems to be so over the *relevant* (lower) portion of the spectrum, i.e. in the subspace in which most of the approximation process effectively takes place. Further *theoretical* efforts regarding stability are recommended.

ACKNOWLEDGEMENTS

Circa 1981, Dr. Michael Cullen (British Meteorological Office) suggested that we lump the mass for pressure but not for velocity. We next thank Dr. Alan Hindmarsh (LLNL) whose assistance in the design, analysis and testing/debugging of the new algorithms was vitally important—especially that of global error analysis. Dr. Peter Brown (LLNL) was good enough to help us in certain linear algebra issues. Next, a very special thanks to Dr. John Leone (LLNL) for his assistance in algorithm design, computer simulations and (especially) for his persistent suspicion (which no doubt still, and justifiably, remains to some degree) and resulting questions regarding half-baked 'explanations' that provided continual impetus to push on. Several conversations with Prof. Alex Chorin (UC Berkeley) and Dr. John Bell (LLNL) were very useful and are gratefully acknowledged. Prof. Robert Sani (University of Colorado, Boulder) provided his usual useful input of guidance, encouragement and analytical support. We also thank Prof. R. Kinney (University of Arizona) for helping us to better understand vorticity generation. Finally, the following people helped us to check and understand our new code by running other codes as cross-checks/verifications: Dr. Michael Engelman (FDI) and Prof. Jeffrey Derby (University of Minnesota) who used the FIDAP code, and Dr. Robert Lee (LLNL) who used the FETISH code.

We are also grateful to Linda Kennedy for her competence and tolerance while typing the manuscript.

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.

APPENDIX: SUMMARY OF PROPERTIES OF PROJECTION OPERATORS

General (continuum and discrete)	*Continuum*	*Discrete*
Projection to div-free subspace: $\wp^2=\wp$, div $\wp=0$	$\wp\equiv I-\nabla(\nabla^2)^{-1}\nabla\cdot$ curl $\wp=$curl	$\wp\equiv I-M_L^{-1}CA^{-1}C^T$ —
$Q\equiv I-\wp$, $Q^2=Q$; div $Q=$div	$Q\equiv\nabla(\nabla^2)^{-1}\nabla\cdot$	$Q\equiv M_L^{-1}CA^{-1}C^T$
$Q\wp=0$: $w\equiv\wp v\rightarrow Qw=0$; $\wp$ projects onto the null space of Q, i.e. onto the space of div-free vectors	$\nabla\cdot w=0$	$C^Tw=0$
$\wp Q=0$: $w\equiv Qv\rightarrow\wp w=0$; Q projects onto the null space of $\wp$, i.e. onto the space of vectors that are gradients of scalars	$w=\nabla\phi$ where $\nabla^2\phi=\nabla\cdot v$ (also, curl $w=0$)	$w=M_L^{-1}C\phi$ where $A\phi=C^Tv$ —
If $u\equiv\wp v$ and $w\equiv Qv$, then $u+w=v$. Also, $u\perp w$, i.e. $\|u\|^2+\|w\|^2=\|v\|^2$, but only if $n\cdot w=0$ on Γ (continuum)	$(a,b)\equiv\int_\Omega ab$, and $\|v\|^2=(v,v)$	$(a,b)\equiv a^TM_Lb$ and $\|v\|^2=(v,v)$
The eigenvalues of $\wp$ are either 0 or 1 (ditto Q); $\|\wp\|=1$ (ditto Q), so that the projections are norm-reducing: $w=\wp v\rightarrow\|w\|\leq\|v\|$ (ditto Q)		(A symmetric projection $\wp_2$, can also be constructed: $\wp_2\equiv I-M_L^{-1/2}CA^{-1}C^TM_L^{-1/2}$, which gives orthogonality in L_2: $(a,b)\equiv a^Tb$)

REFERENCES

1. P. M. Gresho, S. Chan, C. 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 (1984).
2. P. M. Gresho and S. Chan, 'A new semi-implicit method for solving the time-dependent conservation equations for incompressible flow', *Proc. Numerical Methods in Laminar and Turbulent Flow*, Pineridge Press, Swansea, 1985, pp. 3–21.
3. P. Gresho, R. Lee and R. Sani, 'Advection-dominated flows, with emphasis on the consequences of mass lumping', *Proc. Second Int. Symp. on Finite Element Methods in Flow Problems*, Int. Center for Computer-Aided Design, Santa Margherita, Italy, 1976, pp. 743–756.
4. P. M. Gresho, R. L. Lee and R. Sani, 'Advection-dominated flows, with emphasis on the consequences of mass lumping', *Finite Elements in Fluids, Vol. 3*, Wiley, Chichester, 1978, pp. 335–350.
5. L. N. Treffethen, 'Group velocity in finite difference schemes', *SIAM Rev.*, **24**, 113 (1982).
6. R. Lee, P. Gresho and R. Sani, 'A comparative study of certain finite-element and finite-difference methods in advection–diffusion simulations', *Proc. Summer Computer Simulation Conf.*, Washington, DC, 12–14 July 1976, pp. 37–42.
7. J. Rowley and P. M. Gresho, 'Some new results using quadratic finite elements for pure advection', *Proc. IMACS Symp. on Numerical Solutions of PDE's*, Int. Center for Computer-Aided Design, Leheigh University, June 1987, pp. 202–209.
8. P. M. Gresho, R. L. Lee and C. Upson, 'FEM solution of the Navier–Stokes equations for vortex shedding behind a cylinder: experiments with the four-node element', *Adv. Water Resources*, **4**, 175–184 (1981).
9. P. M. Gresho and R. Lee, 'Don't suppress the wiggles—they're telling you something!', *Comput. Fluids*, **9**, 223–255 (1981).
10. P. M. Gresho and S. Chan, 'Solving the incompressible Navier–Stokes equations using consistent mass and a pressure Poisson equation', *Proc. ASME Symp. on Recent Advances in Computational Fluid Dynamics*, Chicago, IL, 28 November–2 December 1988, pp. 51–75.
11. P. M. Gresho and S. Chan, 'Semi-consistent mass matrix techniques for solving the incompressible Navier–Stokes equations', *LLNL Report UCRL-99503*, 1988.
12. P. M. Gresho, R. Lee and R. Sani, 'On the time-dependent solution of the incompressible Navier–Stokes equations in two- and three-dimensions', *Recent Advances in Numerical Methods in Fluids, Vol. 1*, Pineridge Press, Swansea, 1980, pp. 27–81.
13. L. Petzold and P. Lötstedt, 'Numerical solution of nonlinear differential equations with algebraic constraints II: Practical implications', *SIAM J. Sci. Stat. Comput.*, **7**, 720 (1986).
14. P. M. Gresho, R. Sani and M. Engelman, *Incompressible Flows and the Finite Element Method*, Wiley, Chichester, 1991 (in preparation).
15. R. Lee, P. Gresho, S. Chan, R. Sani and M. Cullen, 'Conservation laws for primitive variable formulations of the incompressible flow equations using the Galerkin finite element method', in R. Gallagher *et al.* (eds), *Finite Elements in Fluids, Vol. 4*, Wiley, Chichester, 1982, pp. 21–45.
16. P. M. Gresho, 'Time integration and conjugate gradient methods for the incompressible Navier–Stokes equation', *Proc. 6th Int. Conf. on Finite Elements in Water Resources*, Springer–Verlag, Lisbon, June 1986, pp. 3–29.
17. A. Chorin, 'Numerical solution of the Navier–Stokes equations', *Math. Comput.*, **22**, 745 (1968).
18. J. Kim and P. Moin, 'Application of a fractional-step method to incompressible Navier–Stokes equations', *J. Comput. Phys.*, **59**, 308 (1985).
19. J. Donea, S. Giuliani and H. Laval, 'Accurate explicit finite element schemes for convective–conductive heat transfer problem', *AMD Vol. 34, Finite Element Methods for Convection-Dominated Flows*, ASME, 1979, pp. 149–166.
20. J. Donea, S. Giuliani, H. Laval and L. Quartapelle, 'Finite element solutions of the unsteady Navier–Stokes equations by a fractional-step method', *Comput. Methods Appl. Mech. Eng.*, **30**, 53 (1982).
21. P. M. Gresho, 'Comments on: The significance of chequerboarding in a Galerkin finite element solution of the Navier–Stokes equation', *Int. j. numer. methods Eng.*, **17**, 790 (1982).
22. R. Sani, P. 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, Part 2', *Int. j. numer. methods fluids*, **1**, 171–204 (1981).
23. 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**, 135 (1969).
24. U. Ghia, K. Ghia and C. Shinn, 'High-*Re* solutions for incompressible flow using the Navier–Stokes equations and a multi-grid method', *J. Comput. Phys.*, **48**, 387 (1982).
25. P. M. Davis and E. Moore, 'A numerical study of vortex shedding from rectangles', *J. Fluid Mech.*, **116**, 475 (1982).
26. R. W. Davis, E. Moore and L. Purtell, 'A numerical–experimental study of confined flow around rectangular cylinders', *Phys. Fluids*, **27**, 46 (1984).
27. A. Okajima, 'Strouhal numbers of rectangular cylinders', *J. Fluid Mech.*, **123**, 379 (1982).
28. FIDAP, *FIDAP Users Manual, Version 4.0*, Fluid Dynamics International, Evanston, IL, 1987.
29. A. J. Lugt, *Vortex Flow in Nature and Technology*, Wiley, New York, 1983.
30. J. Steger and P. Kutler, 'Implicit finite difference procedures for the computation of vortex wakes', *AIAA J.*, **15**, 581 (1977).

31. S. E. Logan, 'An approach to the dust devil vortex', *AIAA J.*, **9**, 660 (1971).
32. J. Donea, L. Quartapelle and V. Selmin, 'Analysis of time discretization in the finite element solution of hyperbolic equations', *J. Comput. Phys.*, **27**, 463–499 (1987).
33. A. J. Baker and J. Kim, 'A Taylor weak-statement algorithm for hyperbolic conservation laws', *Int. j. numer. methods fluids*, **7**, 489 (1987).
34. T. Tezduyar, J. Liou and D. Ganjoo, 'Incompressible flow computations based on the vorticity–stream function and velocity–pressure formulations', *UMSI 89/86*, University of Minnesota, Supercomputer Institute, Minneapolis, MN, May 1989.
35. T. J. R. Hughes, 'Recent progress in the development and understanding of SUPG methods with special reference to the compressible Euler and Navier–Stokes equations', *Int. j. numer. methods fluids*, **7**, 1261 (1987).
36. P. M. Gresho and R. Sani, 'On pressure boundary conditions for the incompressible Navier–Stokes equations', *Int. j. numer. methods fluids*, **7**, 1111 (1987).
37. P. M. Gresho, 'Incompressible fluid dynamics: some fundamental formulation issues', in Van Dyke *et al.*, (eds), *Annual Review of Fluid Mechanics*, Annual Reviews, Inc., Palo Alto, Calif., U.S.A., 1991.