INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 4, 557–598 (1984)

A MODIFIED FINITE ELEMENT METHOD FOR SOLVING THE TIME-DEPENDENT, INCOMPRESSIBLE NAVIER–STOKES EQUATIONS. PART 1: THEORY*

PHILIP M. GRESHO, STEVENS T. CHAN, ROBERT L. LEE AND CRAIG D. UPSON

Atmospheric and Geophysical Sciences Division, Lawrence Livermore National Laboratory, Livermore, CA 94550, U.S.A.

SUMMARY

Beginning with the Galerkin finite element method and the simplest appropriate isoparametric element for modelling the Navier–Stokes equations, the spatial approximation is modified in two ways in the interest of cost-effectiveness: the mass matrix is 'lumped' and all coefficient matrices are generated via 1-point quadrature. After appending an hour-glass correction term to the diffusion matrices, the modified semi-discretized equations are integrated in time using the forward (explicit) Euler method in a special way to compensate for that portion of the time truncation error which is intolerable for advection-dominated flows. The scheme is completed by the introduction of a subcycling strategy that permits less frequent updates of the pressure field with little loss of accuracy. These techniques are described and analysed in some detail, and in Part 2 (Applications), the resulting code is demonstrated on three sample problems: steady flow in a lid-driven cavity at $Re \leq 10{,}000$, flow past a circular cylinder at $Re \leq 400$, and the simulation of a heavy gas release over complex topography.

KEY WORDS Navier–Stokes Equations Finite Element Method Incompressible Flow Advection-Diffusion

CONTENTS

* This invited paper is an extended, and refereed version of one presented at the Fourth International Symposium on Finite Elements in Flow Problems held in Tokyo, Japan, 26–29 July, 1982.

0271-2091/84/060557-42$04.20

Received 16 May 1983
Revised 1 December 1983

1. INTRODUCTION

We have developed a numerical method for solving the time-dependent, incompressible Navier–Stokes (NS) equations (or variants, such as the Boussinesq equations or the anelastic equations) and the advection–diffusion (AD) equation in two and three dimensions (2D and 3D). Although the technique was originally derived via the conventional Galerkin finite element method (GFEM), we invoke two ensuing simplifying approximations that generate a scheme which is probably better described as a blend of finite elements and finite differences, i.e. an 'isoparametric element, finite difference method'.

The philosophy guiding the evolution of the techniques is a common one: simplicity and cost-effectiveness. Starting with the simplest GFEM approximation for spatial discretization, we invoke the simplest method for advancing the solution in time. We therefore use multilinear basis functions for the velocity (bilinear in 2D, trilinear in 3D) and (piecewise) constant approximation for the pressure. The explicit (forward) Euler method is then used to integrate the resulting ordinary differential equations (ODEs) in time.

The two *a priori* simplifications to the GFEM are: (1) the 'mass matrix', which couples the time-derivatives in the GFEM, is replaced by a diagonal matrix via 'mass lumping', thus decoupling the time derivatives and paving the way for explicit time integration, and (2) all Galerkin integrals are evaluated approximately by invoking one-point quadrature. (Sometimes this evaluation is exact—see below.) We believe that these two *ad hoc* modifications, when combined with those discussed below, lead to a scheme which is generally more cost-effective (accuracy per unit cost) than that when GFEM is used (consistent mass, higher order quadrature and implicit time integration methods). In 2D, we have ample evidence to support this position since we also have our own GFEM code.[1] We have no such evidence in 3D, however, because we were afraid to extend our GFEM scheme to 3D—not that this cannot be done (e.g. Reference 2); our fear was related to our goal of attaining a real-time predictive capability for the atmospheric boundary layer.

The first simplification (mass lumping) also permits the maximum uncoupling of the NS equations and leads to a discrete Poisson equation for the pressure that is easy to generate and solve. Regarding the second simplification (1-point quadrature), we point out that for large time-dependent problems, especially in 3D (say 10^4 nodes), the use of more accurate (higher-order) quadrature methods (typically Gauss–Legendre) can be quite impractical (cost-ineffective) even though most of the integrations are exact. Although the element matrices are not time-dependent, and could thus be generated once and for all, stored on disk, and retrieved every time they are needed, this proves to be quite expensive in I/O (input/output) cost—at least on a CRAY-1 computer, for which the CPU (central processing unit) is very fast and the transfer rate of data (I/O) is relatively very slow. The alternative of (accurately) recomputing the Galerkin integrals at each time step would probably lead to even higher cost. Even more disappointing is the fact that typically little additional accuracy is gained, even with these heavy computational penalties, as has been recognized in the field of solid mechanics[3], and will be partially demonstrated in this paper for fluid mechanics—especially when mass lumping is invoked. On the other hand, 1-point quadrature permits the rapid (approximate) evaluation of the Galerkin integrals 'on the fly' (whenever needed), which totally eliminates this portion of I/O, leads to efficient code vectorization, and has been found to reduce both I/O and CPU costs (in the 3D cases studied) by about an order of magnitude.

Two additional modifications that provide further increase in computational speed are associated with the time integration of the ODEs. The first of these is called balancing tensor diffusivity (BTD), or in particular, balancing tensor viscosity for the NS equations, and is used to permit larger time steps with no loss of accuracy—indeed, a gain in accuracy is often realized. The second is called subcycling, a procedure which permits less frequent updates of the pressure relative to the stability-limited processes of advection and diffusion.

In the remainder of this paper, these techniques will be described in detail, analysed with respect to accuracy and stability, and (in Part 2) demonstrated via numerical examples.

2. GOVERNING EQUATIONS AND BASIC SPATIAL DISCRETIZATION

The governing equations of interest here are the NS equations for an incompressible, constant property fluid in the Boussinesq approximation. In dimensionless form these are

$$\partial \mathbf{u}/\partial t + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla P + Re^{-1} \nabla^2 \mathbf{u} + Fr^{-1} \mathbf{k} T \tag{1a}$$

$$\partial T/\partial t + \mathbf{u} \cdot \nabla T = Pe^{-1} \nabla^2 T \tag{1b}$$

and

$$\nabla \cdot \mathbf{u} = 0 \tag{1c}$$

where $\mathbf{u} = (u, v)$ or (u, v, w) is the velocity, P is the pressure deviation from hydrostatic at the reference temperature, T is the temperature deviation from the reference temperature, $Re = u_0 L/\nu$ is the Reynolds number, $Fr = u_0^2/(\gamma \Delta T g L)$ is the Froude number (or an inverse Richardson number), $\mathbf{k}$ is the unit vector in the direction of gravity, and $Pe = Re\, Pr = u_0 L/\kappa$ is the Peclet number ($Pr = \nu/\kappa$ is the Prandtl number). Also, u_0 is the reference (characteristic) velocity, L is the characteristic length, $\nu = \mu/\rho$ is the kinematic viscosity, κ is the thermal diffusivity, γ is the volumetric thermal expansion coefficient, ΔT is the characteristic temperature difference, and g is the gravitational acceleration. In (1), the unit of length is L, that of time is L/u_0, that of velocity is u_0, that of pressure is ρu_0^2, and that of temperature is ΔT. Of course for isothermal flows, the buoyancy term in (1a) is dropped and (1b) is omitted.

Finally, for buoyancy-driven flows, the characteristic velocity (usually) is $u_0 = \sqrt{(\gamma g L\, \Delta T)}$, $Re = \sqrt{(Ra/Pr)}$, $Fr = 1$ and $Pe = \sqrt{(RaPr)}$, where $Ra \equiv \gamma g L^3 \Delta T/\kappa\nu$ is the Rayleigh number. Given an initial temperature field, $T_0(\mathbf{x})$, an initial velocity field, $\mathbf{v}(\mathbf{x})$ which satisfies $\nabla \cdot \mathbf{v} = 0$ (see Reference 1) and appropriate boundary conditions (BCs) for $\mathbf{u}$ and T (none are needed for P; see Reference 1), equation (1) can be solved (in principal) for $\mathbf{u}$, P and T as functions of space and time.

The finite element spatial discretization of (1) is performed using the Galerkin method via the following expansions in the piecewise polynomial basis functions associated with the FEM,

$$\mathbf{u}^h(\mathbf{x}, t) = \sum_{i=1}^{N} \mathbf{u}_i(t)\varphi_i(\mathbf{x}) \tag{2a}$$

$$T^h(\mathbf{x}, t) = \sum_{i=1}^{N} T_i(t)\varphi_i(\mathbf{x}) \tag{2b}$$

and

$$P^h(\mathbf{x}, t) = \sum_{i=1}^{M} P_i(t)\psi_i(\mathbf{x}) \tag{2c}$$

where, in the discretized domain, there are N nodes for velocity and temperature and M elements. In (2), $\varphi_i(\mathbf{x})$ is a C^0 piecewise multilinear basis function defined on isoparametric 'quadrilateral' elements (bilinear in 2D and trilinear in 3D), $\psi_i(\mathbf{x})$ is a piecewise constant basis function (unity on element i and zero on all other elements), and the superscript h indicates a finite dimensional approximation. Inserting (2) into the weak (Galerkin) form of (1) (see, e.g. reference 1), which permits φ_i to be discontinuous in the first derivatives, ψ_i to be discontinuous, and introduces the natural boundary conditions, leads to the following set of ODEs with concomitant algebraic constraints (a differential-algebraic system[4])—the GFEM equations, written here in compact matrix form:

$$M\dot{u} + K(u)u + CP = f, \qquad u(0) = u_0 \quad \text{where} \quad C^{\mathrm{T}}u_0 = 0 \tag{3a}$$

$$M_s\dot{T} + K_s(u)T = f_s, \qquad T(0) = T_0 \tag{3b}$$

and

$$C^{\mathrm{T}}u = 0 \tag{3c}$$

where now u is a global vector containing all the nodal values of u, v (and w), P is a global pressure vector, and f is a 'force' vector which incorporates the buoyancy term and the natural BCs on velocity. M is the mass matrix (which we henceforth regard as lumped via (essentially) row-sum at element level), $K(u) \equiv N(u) + K$ is the 'advection + diffusion' matrix, where K is the 'viscous' or diffusion matrix and $N(u)$ is the advection matrix, C is the gradient matrix and its transpose, C^{T}, is the divergence matrix. In the AD equation for temperature, the subscript s refers to similar, but smaller sets of coefficients, and T is a global temperature vector.

Remarks

(i) The natural BCs associated with $\nabla^2\mathbf{u}$ are different from those when the viscous terms are expressed in stress-divergence form (cf. Reference 1); they are generally just as useful, however.

(ii) The initial velocity field must satisfy the discretely divergence-free condition rather

than the continuous one; this fact could, for example, preclude the use of the interpolant of a solenoidal field as an initial condition.

To maximize the efficiency of code vectorization, the u and T vectors in (3) contain *all* of the nodal values, including those specified via Dirichlet BCs. To enforce these (essential) BCs, we replace the appropriate diagonal terms in the lumped mass matrices by a large number (say 10^{20}) and replace the corresponding entries in the 'force' vectors with the products of the same large number and the time derivative of the specified variables (typically zero).

In order to integrate (3) with an explicit method, we will need the discretized Poisson equation for the pressure. This is easily derived from (3a) and (3c) by noting that $C^{\mathrm{T}}\dot{u} = 0$ since $C^{\mathrm{T}}u = 0$ for all time, namely,

$$(C^{\mathrm{T}}M^{-1}C)P \equiv AP = C^{\mathrm{T}}M^{-1}[f - K(u)u] \tag{4}$$

where $A \equiv C^{\mathrm{T}}M^{-1}C$ is a discretized approximation to the Laplacian operator into which velocity BCs have been 'automatically' incorporated. It is a global matrix and must be so constructed, rather than being formed from element level matrices. This construction is quite simple and straightforward (more so via 1-point quadrature) since M^{-1} is a diagional matrix; furthermore, it need only be formed 'once per problem' since it is constant, and this is conveniently done in a preprocessor code. Equation (4) is a discrete approximation to the continuum Poisson equation,

$$\nabla^2 P = Fr^{-1}\,\nabla \cdot (\mathbf{k}T) - \nabla \cdot (\mathbf{u} \cdot \nabla \mathbf{u}) \tag{5}$$

which is implied by (1a) and (1c). Just as (5) can be used in place of (1c) in the continuum, so can (4) be used in place of (3c); i.e. (3a) and (4) imply (3c). Since (4) approximates (5), implies (3c), and incorporates (automatically) the appropriate BCs for the pressure (recall that the original equations required BCs only for $\mathbf{u}$ and T), we refer to it as the *consistent* discretized Poisson equation for the pressure; see also Reference 5.

If one or more pressures are to be specified (e.g. to set the hydrostatic pressure level and/or to avoid some of the effects associated with spurious pressure modes—cf. Reference 6), the same technique discussed above is employed; namely replace the appropriate diagonal entries in A by a large number and replace the corresponding entries in the right hand side vector by the products of the same large number and the desired pressure values (again, typically zero).

The final semi-discretized equations considered henceforth are thus (3a), (3b), and (4).

3. ONE-POINT QUADRATURE AND ITS EFFECTS

Before invoking 1-point quadrature, the element matrices associated with (3) and (4) are delineated:

$$M_{ij}^{e} \equiv \delta_{ij} \int_{e} \phi_i \, \mathrm{d}\Omega \tag{6a}$$

where δ_{ij} is the Kronecker delta and the domain of integration is element e,

$$K_{ij}^{e} \equiv Re^{-1} \int_{e} \nabla\varphi_i \cdot \nabla\varphi_j \, \mathrm{d}\Omega \tag{6b}$$

$$N_{ij}^{e}(u) \equiv \int_{e} \varphi_i \mathbf{u}^h \cdot \nabla\varphi_j \, \mathrm{d}\Omega \tag{6c}$$

and

$$C_{ie}^{(k)} \equiv -\int_e \frac{\partial \varphi_i}{\partial x_k} \psi_e \, d\Omega = -\int_e \frac{\partial \varphi_i}{\partial x_k} d\Omega \tag{6d}$$

where Re is replaced by Pe for the diffusion matrix in the temperature equation.

Since $\mathbf{u}^h = \sum_{k=1}^{N} \mathbf{u}_k \varphi_k$ in (6c), it is clear that an integral of triple products is generally required to generate the non-linear advection matrix, $N(u)$; see Reference 1 for details. This costly procedure can be avoided (ostensibly with an additional loss of accuracy) by considering another *ad hoc* (non-Galerkin) modification called 'centroid advection velocity', in which $\mathbf{u}^h$ in (6c) is evaluated only at the element centroid, thus rendering it constant in the integrand and leads to

$$N_{ij}^e(u) = \bar{u}_k \int_e \varphi_i \, \partial \varphi_j / \partial x_k \, d\Omega \tag{6c}'$$

in which $\bar{u}_k$ is the kth component of the average velocity in the element (i.e. the arithmetic average of the nodal values) and the summation convention on repeated indices is in effect.

Remarks

(i) This short cut has the effect of changing the global advection approximation from a complex stencil of 2/3 centred differences, 1/6 upwind differences and 1/6 downwind differences, (at least in 2D on a regular mesh[7]) to a simpler stencil with the same type of weighting. It may, however, introduce some aliasing error in contrast to honest GFEM, which does not.[8]

(ii) When 1-point quadrature is employed, both (6c) and (6c)' lead to identical results; the stencil is then a simple combination of 1/2 centred, 1/4 upwind and 1/4 downwind differences, a result which is easily stated in words (for 2D and 3D): the average (centroid) velocity over an element is multiplied by the average gradient (of the advected variable) within the element and this result is averaged over the number of elements sharing the node in question. On irregular or distorted meshes, the interpretation is similar except that area (or volume)-weighted averages are employed in the final averaging step.

As a potential basis for comparision, Table I presents the GFEM 'required' quadrature rules[9] for the elements under consideration. Using only these results as a yardstick, it would appear that 1-point quadrature is nearly indefensible—especially in 3D. It turns out to be not nearly so bad, however, when the final results are examined from a finite difference

Table I. Number of Gauss points required in each coordinate direction to evaluate various element matrices (and element size, Ω^e) exactly

	Elements			
Matrices	Rectangle and parallelogram	General quadrilateral	Brick and parallelepiped	General hexahedron
M^e	1	2	1	2
K^e	2	>3*	2	>3*
N^e	2	2	2	2†
C^e (and Ω^e)	1	1	1	2

* It is not generally possible to evaluate K^e exactly using Gauss quadrature.
† A 3-point rule is required if (6c) is used rather than (6c)'.

viewpoint, e.g. most of the approximations are 'Taylor-series legitimate'—at least on regular meshes.

The principal arguments in favour of 1 point quadrature are based on the fact that it can be made to perform in a cost-effective manner, i.e. it *does* work.[10] It seems appropriate at this point to present our definition of 1-point quadrature and to explain the computational economies accruing from it. First we compute each element 'size' (area or volume) exactly (which generally requires 2-point quadrature in 3D—see Table I) and store the resulting vector in memory. Then the 1-point element matrices are constructed (in principle, but see below) by evaluating the appropriate integrand at the element centroid and multiplying this result by the element size. However, since the integrands in the advection and diffusion matrices involve first derivatives of the basis functions, they can be obtained from the 1-point quadrature C-matrix, half of the entries of which (owing to the antisymmetric nature of the shape function derivatives at element centroids) are stored in memory. In the following sections, we will discuss the individual 'matrix construction' in more detail, perform some relevant error analyses which, among other things, reveal the major deficiency of 1-point quadrature, and present a simple and effective remedy for this problem.

3.1. Element volume (or area) and mass matrix

Since it has been claimed[11] that convergence of the FEM requires, among other things, an exact integration of element size, and because the extra cost of doing so is negligible, we integrate the element size exactly (à la Table I). This need be done only once, in a preprocessor code, and the results are stored in memory as a single vector (as mentioned above). Each element size is then distributed equally to the associated nodal points to form the global, lumped mass matrix, another vector which is stored in memory.

Remarks

(i) If the mesh is composed of simply-shaped elements, the result is truly the same as that using 'row-sum' lumping. For distorted elements, the results will differ somewhat, to the extent of the distortion—which should be kept as small as possible in general; see also Reference 12.

(ii) The true row-sum technique could be easily employed for any element shape, simply by using a 2-point rule in (6a), at little extra cost (in the preprocessor); we probably should, but have not yet, implemented this option.

(iii) If steady-state solutions are obtained, the results are (necessarily) independent of any mass lumping procedure.

3.2. Divergence and gradient matrices

One-point quadrature on (6d) is exact for 2D general quadrilaterals and for 3D elements of simple shape. In these cases, both element level (à la (3c)) and global mass balances are maintained. For distorted 3D elements, however, element level mass balance is not guaranteed and, as a result, global mass conservation cannot be assured. Nevertheless, it appears (from our experience) that for meshes consisting of only mildly distorted elements, the mass imbalance is probably tolerable in that it does not seem to adversely affect the overall quality of the solution. For a 3D mesh consisting of many highly distorted elements, however, prudence would suggest that 2-point quadrature be employed on the C-matrix. (Prudence would also suggest, however, that such meshes be avoided.) The cost of these better results is

a loss in code efficiency and additional storage (in memory or on disk) is required.

In order to gain further insight into the divergence approximation, we performed a Taylor-series-like analysis to determine how well $\nabla \cdot \mathbf{u}$ is approximated at the element centroid (both 1-point and 2-point quadrature were tested in 3D). By inserting increasingly higher order polynomials into the stencil associated with (3c) and (6d) and dividing the result by the element size, the accuracy of the difference operator is obtained by noting when the result is exact. We found that the accuracy is $O(h^2)$ and $O(h)$ for simply-shaped and distorted elements, respectively, in both 2D and 3D, even though 1-point quadrature is inexact for distorted 3D elements.

The accuracy of the approximation to the pressure gradient, however, is *generally* one order lower than that of the divergence of the velocity field and, significantly, is (in 2D) independent of the Gauss rule used. Also, unlike the divergence operator, analysis of the gradient operator requires a 'patch' of elements, rather than a single element. Accordingly, we have performed Taylor-series analyses on various 4-patches in 2D (wherein 1-point quadrature is exact) with the following results:

(i) The error in the ∇P approximation is $O(h^2)$ on a uniform mesh of equal-sized rectangles (and pure rotation does not degrade the accuracy).
(ii) It is $O(h)$ on a general rectangular mesh.
(iii) It is $O(1)$ on a mesh of distorted elements, a rather disconcerting observation.[13,14] This means, for example (in a Taylor-series sense) that the intended approximation to $\partial P/\partial x$, would actually look more like $[1+O(\varepsilon)]\,\partial P/\partial x + O(\varepsilon)\,\partial P/\partial y$, where ε is a measure of the grid distortion ($\varepsilon = 0$ for parallelograms); this error could be quite serious if the distortion is large and if $\partial P/\partial y \gg \partial P/\partial x$, the latter of which often occurs in buoyancy-affected flows.

Although we have not extended this analysis to 3-D, we believe that there would be no surprises.

These results, most of which relate to the use of piecewise-constant pressures rather than to any additional problems associated with 1-point quadrature, are somewhat ameliorated by the following additional points:

(i) In all numerical studies involving mesh refinement, the pressures (and the velocities, of course) seemed to converge even when the ultimate mesh would involve distorted elements. It is true, however, that a coarse grid of distorted elements can generate large errors in ∇P (and thus in $\mathbf{u}$), a point we will return to when we discuss numerical results.
(ii) Consistency in a Taylor-series sense is a sufficient, but not necessary condition for convergence.[15]

Indeed, one may even argue that Taylor-series analyses are inappropriate for finite element schemes (they are rarely, if ever, used in mathematical analyses of error and/or convergence)—see also Reference 13. We believe, however, that results from *all* types of error analyses are useful and should be taken into account, but that the ultimate test is 'in the field'—how well does the method perform on a wide variety of problems? The foregoing results suggest that simulations in which the elements are of simple shapes (with gradual grading of element size; $\Delta h/h \ll 1$) are likely to be much more accurate than those with distorted elements—and our experience seems to support this viewpoint.

3.3. Advection matrix.

The 1-point quadrature C-matrix is used extensively and effectively to directly construct the advection (and diffusion, discussed later) contributions in (3), i.e. the element level advection (and diffusion) matrices are never explicitly formed. Rather, the entire advection contribution associated with a particular element is computed as shown below. First, however, we must apply 1-point quadrature to the C-matrix in (6d), to obtain

$$C_{ie}^{(k)} \equiv -(\partial \phi_i/\partial x_k)_0 \Omega_e \tag{6d}'$$

where $(\ )_0$ denotes centroid evaluation and Ω_e is the eth element of an M-vector containing the element sizes. Using the temperature equation for demonstration, and invoking the summation convention on repeated indices, 1-point quadrature on (6c)′ leads to (upon multiplication by the T vector)

$$\begin{aligned} N_{ij}^e T_j &= \bar{u}_k T_j \int_e \varphi_i \, \partial \varphi_j/\partial x_k \, \mathrm{d}\Omega \\ &= \varphi_i(0) \bar{u}_k T_j (\partial \varphi_j/\partial x_k)_0 \Omega_e \\ &= \varphi_i(0) \bar{u}_k (\partial T/\partial x_k)_0 \Omega_e \\ &= -\varphi_i(0) \bar{u}_k C_{je}^{(k)} T_j \end{aligned} \tag{6c}''$$

Remarks

(i) Since $\varphi_i(0)$ is a constant for all i (1/4 in 2D and 1/8 in 3D), the final result in (6c)″ is seen to be (effectively) a simple scalar; this result is then distributed to each node in element e.

(ii) The centroid gradient, $(\partial T/\partial x_k)_0$, will be seen to be also useful in the evaluation of the diffusion term.

We now examine the accuracy associated with this advection approximation. A standard way to compare advection schemes is via Fourier analysis of the constant-velocity, pure advection equation ($Pe = \infty$ and $\mathbf{u}$ is constant in (1b)): a given wave (with wave number vector $\mathbf{k} = (k_1, k_2)$ or $\mathbf{k} = (k_1, k_2, k_3)$ with wavelength $\lambda = 2\pi/|\mathbf{k}|$) is placed on the infinite span (or on a finite domain with periodic boundary conditions) and the resulting solution obtained from the approximate scheme is compared with the exact solution. A complete analysis involves the consideration of all possible (i.e. mesh-resolvable) wave numbers and a comparision of both phase and group velocities;[16] herein we will be content with a special case (described below) which is probably (we hope) adequate for the purpose of comparison. If a periodic function of the form

$$T_0(\mathbf{x}) = \exp\left(\sum_j i k_j x_j\right) \tag{7}$$

is taken as an initial condition, the exact solution to the pure advection form of (1b) is

$$T(\mathbf{x}, t) = T_0(\mathbf{x}) e^{-i\omega t} \tag{8a}$$

where

$$\omega = \sum_j u_j k_j \tag{8b}$$

is the frequency. The phase speed, c, is given by

$$c = \omega/|\mathbf{k}| = \omega \Big/ \left(\sum_j k_j^2\right)^{1/2} \tag{9}$$

The phase speed is the projection of $\mathbf{u}$ onto $\mathbf{k}$ and measures the wave speed along the wave direction. Now if the same initial wave is placed on a uniform rectangular mesh (a 4-patch analysis in 2D) over which the advection operator $(\mathbf{u}.\nabla)$ has been approximated (e.g. via FEM or FDM), the resulting frequency (and phase speed, and group velocity) will differ from that given by (8b), owing to numerical dispersion; i.e. the semi-discretized solution will be given by an equation similar to (8a) with ω replaced by $\bar{\omega}$ where, for non-dissipative schemes of the type considered herein, $\bar{\omega}$ is real. We have performed this analysis and present a summary of the results. The ratio of $\bar{\omega}/\omega$ (or $\bar{c}/c$ where $\bar{c}$ is the approximate phase speed) is displayed in functional form in Table II and shown in Figure 1 for the special case wherein the wave number vector is chosen so that $k_j\,\Delta x_j \equiv p$ is constant, where Δx_j is the mesh size in the jth direction and the following nomenclature is adopted: FD is the simplest centred-difference approximation to the advection operator and the others are the four possible combinations of using either consistent mass (C) or lumped mass (L) in conjunction with either 1-point (exact in 1D) or 2-point (always exact) quadrature for the advection matrix. Also shown in Table II is the local truncation error, $(\bar{c}/c - 1)$ as $\Delta x_j \to 0$ for fixed $\mathbf{k}$.

Remarks

(i) For 1D problems, since 1-point quadrature is exact for the advection matrix, the difference in the FEM phase speeds depends only on the type of mass matrix being used. When consistent mass is used, the finite elment method with linear approximation is fourth-order accurate; however, when mass lumping is employed it degrades to the centred, second-order finite difference scheme (L1 = L2 = FD).

(ii) The GFEM scheme, C2, is clearly the most accurate, and it retains fourth-order accuracy in all space dimensions. This scheme, however, is not considered to be viable using isoparametric elements, at least with the current computer capacity, for solving large time-dependent problems (especially in 3D).

(iii) The lumped mass schemes are inferior to the finite difference scheme, especially in 3D (note that only C2 and FD have phase speeds that are independent of n).

(iv) Mass lumping induces much larger errors than those caused by reduced quadrature.

(v) The popular second-order Arakawa scheme[17] is equivalent, for the case of pure advection, to the L2 scheme.

Table II. Phase speeds and truncation errors associated with various discrete approximations to the pure advection equation in n space dimensions ($p = k_j\,\Delta x_j$)

	Properties			
		Leading truncation error terms		
Approximation	Relative phase speed, $\bar{c}/c$	$n=1$	$n=2$	$n=3$
C2	$[3/(2+\cos p)](\sin p/p)$	$-p^4/180$	$-p^4/180$	$-p^4/180$
C1	$[3/(2+\cos p)]^n[(1+\cos p)/2]^{n-1}(\sin p/p)$	$-p^4/180$	$-p^2/12$	$-p^2/6$
L2	$[(2+\cos p)/3]^{n-1}(\sin p/p)$	$-p^2/6$	$-p^2/3$	$-p^2/2$
L1	$[(1+\cos p)/2]^{n-1}(\sin p/p)$	$-p^2/6$	$-5p^2/12$	$-2p^2/3$
FD	$\sin p/p$	$-p^2/6$	$-p^2/6$	$-p^2/6$

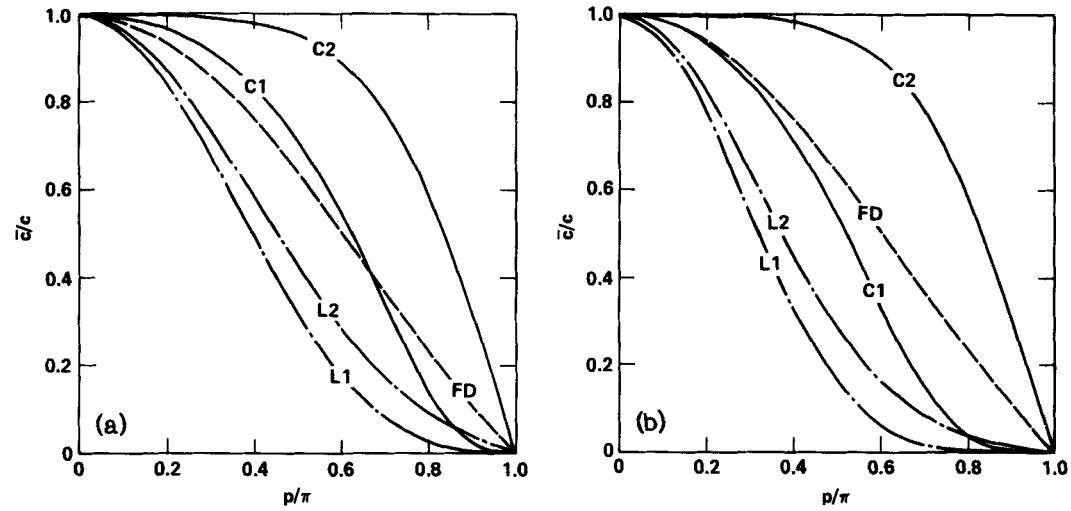

Figure 1. Phase speed vs wave number for various discrete approximations to the pure advection equation in (a) 2D and (b) 3D

(vi) The L1 scheme has recently been derived (by finite difference methods) and advocated by Smolarkiewicz;[18] albeit only in conjunction with the explicit Euler time integration scheme in which the balancing tensor diffusivity (BTD) improvement (see section 4.3) is also employed.

(vii) Our L1 scheme will be shown (section 4.3) to have a more accurate phase speed when a particular time integration scheme is employed.

3.4. *Diffusion matrix*

The diffusion contributions are also efficiently evaluated via the C-matrix in (6d)′; again using the temperature equation as our example, we evaluate (cf. (6b)) $K_{ij}^e T_j$ as follows:

$$\begin{aligned} PeK_{ij}^e T_j &= T_j \int_E (\partial\varphi_i/\partial x_k)(\partial\varphi_j/\partial x_k)\,\mathrm{d}\Omega \\ &= T_j(\partial\varphi_i/\partial x_k)_0(\partial\varphi_j/\partial x_k)_0\Omega_e \\ &= T_j(C_{ie}^{(k)}/\Omega_e)C_{je}^{(k)} \qquad \text{(no sum on } e\text{)} \\ &= -(C_{ie}^{(k)}/\Omega_e)(\partial T/\partial x_k)_0 \quad \text{(no sum on } e\text{)} \end{aligned} \tag{6b}'$$

Remarks

(i) The centroid gradient appears again; it is thus constructed only once and used to compute both the advection and diffusion contributions in each element.

(ii) Owing to antisymmetry in the first derivatives, (6b)′ needs to be evaluated for only half of the nodes in each element.

In order to (partially) assess the performance of 1-point quadrature relative to other schemes in approximating the diffusion (∇^2) operator, we again present some results from Fourier analysis, this time applied to the transient heat equation,

$$\frac{\partial T}{\partial t} = \alpha\,\nabla^2 T \tag{10}$$

If a periodic function of the form given by (7) is again taken as an initial condition, the exact solution to (10) is

$$T(\mathbf{x}, t) = T_0(\mathbf{x})e^{-\beta t} \tag{11}$$

where $\beta = \alpha \sum_j k_j^2$. For a spatially-discretized version of (10), a (uniform) 4-patch analysis will lead to the effective diffusivity, $\bar{\alpha}$, by using the same wave as in (7), but with $\beta = \bar{\alpha} \sum_j k_j^2$ in (11). Unlike the constant true diffusivity, $\bar{\alpha}$ is a function of wave number and is close to α only for long waves (i.e. small $|\mathbf{k}|$). For simplicity, we again take $p = k_j \Delta x_j = \text{constant}$. The results, in the form of $\bar{\alpha}(p)/\alpha$ vs. p, are shown in Table III and Figure 2 (the curves labeled LH will be discussed later). The last three columns display $(\bar{\alpha}/\alpha - 1)$ as $\Delta x_j \to 0$ for fixed $\mathbf{k}$; this is the truncation error.

Remarks

(i) All schemes except one are now second-order accurate. The exception is C1, which is fortuitously fourth-order accurate in 2D. (We include this seemingly impractical case only to further demonstrate the point that mass lumping is generally more deleterious than the use of 1-point quadrature.)

(ii) Again, the GFEM result (C2) is probably the best. It over-diffuses short waves (indeed, all waves), which is probably desirable in most simulations because these waves are more difficult to resolve, are too slowly advected, and presumably (or hopefully) contribute less to the overall solution accuracy.

(iii) Again, the multidimensional lumped mass results are inferior to those from simple finite differences; they are more under-diffusive.

(iv) Again, the depression of the curves from C2 is caused more by mass lumping than by 1-point quadrature.

(v) The above remarks are irrelevant at steady state, for which 1-point quadrature usually performs quite well in practice.

(vi) The biggest problem with 1-point quadrature is revealed at $p = \pi$: the diffusion matrix is singular with respect to '$2\Delta x$' waves, a result which is independent of the mass matrix. Since these waves are neither diffused nor advected, they can cause a serious problem in the form of undamped, spurious spatial oscillations, i.e. wiggles.

In the next section, we describe these 'zero energy modes', which are null vectors of the diffusion matrix, and present a method for overcoming this deficiency.

3.4.1. *Definition and control of* $2\Delta x$ *waves.* Since the 'patch job' for these undiffused waves was transferred from the solid mechanics community, it may be appropriate to also

Table III. Summary of effective diffusivity for various approximation schemes ($n = 1$, 2 or 3 is the number of space dimensions)

Approximation	Properties			
		Leading truncation error terms		
	$\bar{\alpha}/\alpha$	$n=1$	$n=2$	$n=3$
C2	$[3/(2+\cos p)]2(1-\cos p)/p^2$	$p^2/12$	$p^2/12$	$p^2/12$
C1	$[3/(2+\cos p)]^n[(1+\cos p)/2]^{n-1}2(1-\cos p)/p^2$	$p^2/12$	$-p^4/90$	$-p^2/12$
L2	$[(2+\cos p)/3]^{n-1}2(1-\cos p)/p^2$	$-p^2/12$	$-p^2/4$	$-5p^2/12$
L1	$[(1+\cos p)/2]^{n-1}2(1-\cos p)/p^2$	$-p^2/12$	$-p^2/3$	$-7p^2/12$
FD	$2(1-\cos p)/p^2$	$-p^2/12$	$-p^2/12$	$-p^2/12$

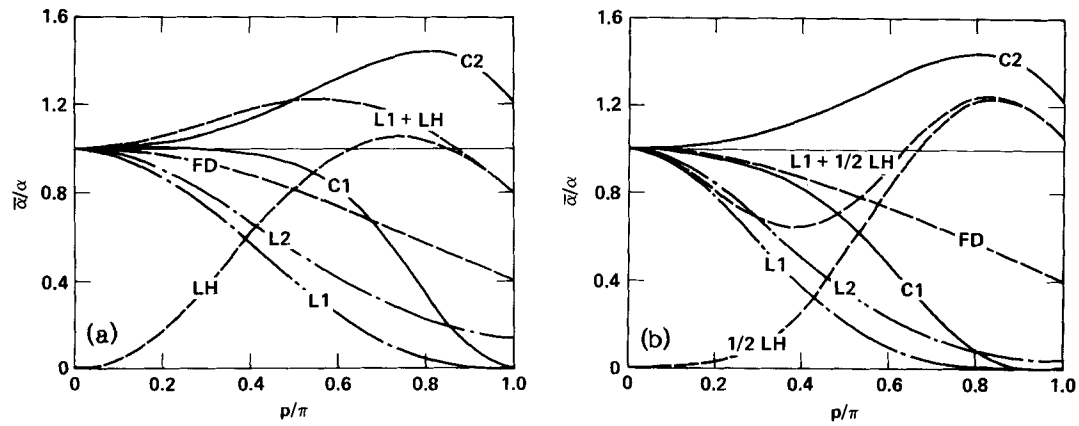

Figure 2. Effective diffusivity vs wave number for various discrete approximations to the transient heat equation in (a) 2D and (b) 3D

first describe them in this language: in the Lagrangian FEM codes of solid mechanics, the problem is described in terms of the so-called hour-glass patterns and is interpreted in terms of zero energy mode shapes.[3,19,20] In fluid mechanics, using an Eulerian reference frame, however, it seems more appropriate to describe the problem in terms of '$2\Delta x$' waves, although it is indeed the same 'problem'.

In 2D, there is only one $2\Delta x$ wave (more properly and more generally, it is a $2\Delta x$ by $2\Delta y$ wave), which has alternating nodal values of ± 1. The 1-point diffusion matrix annihilates this 'oscillating' vector, which is of course just another way to say that the matrix is singular and that ± 1 is a vector in the null space. We also remark that this mode has much in common with the spurious chequerboard pressure mode discussed by Sani *et al.*[6]

In 3D, however, there are four null vector waves, one of which is $2\Delta x$ by $2\Delta y$ by $2\Delta z$ (fully three-dimensional with nodal values of ± 1); the other three are two-dimensional, one in each plane. These four vectors, which are orthogonal to each other, are shown (schematically) in Figure 3. As in 2D, it is a simple matter to verify that the 1-point element-level diffusion matrix annihilates each of these vectors, *regardless* of the shape of the elements.

Actually, total (global) annihilation of these vectors in *bounded* domains occurs only in the case of Neumann (natural) or periodic boundary conditions. However, even in the more common case wherein Dirichlet conditions are applied on at least part of the boundary, local '$2\Delta x$ waves' (whose effects are deleterious) can be generated. Thus, it is often desirable to provide some sort of control in the form of damping.

The technique we employ to deal with these problems was (we believe) devised by Goudreau and Hallquist[21] and is first described for equal-sized elements: simply form the outer product of each null space vector with itself at element level ($\mathbf{x}\mathbf{x}^{\mathrm{T}}$ where, e.g. $\mathbf{x}^{\mathrm{T}} = (1\ -1\ 1\ -1)$ for 2D), multiply the resulting hour-glass matrix (or matrices) by the diffusivity and, if necessary, by a tuning constant (scalar) which is dimensionless in 2D but must have units of length in 3D, and add the result to the 1-point quadrature diffusion matrix. The additional 'diffusivities' provided by the hour-glass matrix are shown in Figure 2, both before and after being added to the 1-point diffusion matrix, as LH—lumped mass+hour-glass, in 2D (or $\frac{1}{2}$LH in 3D, where LH is the 3D hour-glass matrix)—and L1+LH (or L1+$\frac{1}{2}$LH). In 2D the 'tuning constant' is 1·0 in the figure (which is near-optimum, but may be a little too high—by 10–20 per cent), whereas in 3D the constant has been taken as $h/2$ for the 3D wave, where h is a linear measure of the element size. In both

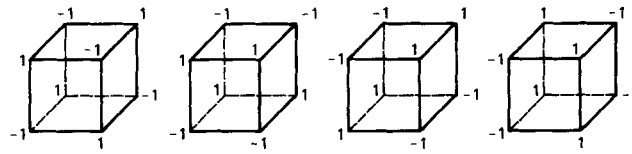

Figure 3. The four '$2\Delta x$' wave patterns on a single element in 3D

cases the total behaviour of the effective diffusivity is clearly much improved, especially for short waves (which, of course, was the goal).

It is also instructive to perform Taylor-series analysis on the various discrete approximations to the diffusion term in (10). When this is applied to an appropriately assembled patch of elements, the following results are obtained for 2D:

$$K_{\mathrm{fd}} \rightarrow -\nabla^2 T - \frac{h^2}{12}(T_{xxxx} + T_{yyyy}) + O(h^4) \tag{12a}$$

$$K_{1\mathrm{p}} \rightarrow -\nabla^2 T - \frac{h^2}{12}(T_{xxxx} + T_{yyyy} + 6T_{xxyy}) + O(h^4) \tag{12b}$$

$$K_{2\mathrm{p}} \rightarrow -\nabla^2 T - \frac{h^2}{12}(T_{xxxx} + T_{yyyy} + 4T_{xxyy}) + O(h^4) \tag{12c}$$

$$H \rightarrow h^2 T_{xxyy} + O(h^4) \tag{12d}$$

and

$$K_{1\mathrm{p}} + H \rightarrow -\nabla^2 T - \frac{h^2}{12}(T_{xxxx} - 6T_{xxyy} + T_{yyyy}) + O(h^4) \tag{12e}$$

In the above, K_{fd}, $K_{1\mathrm{p}}$, $K_{2\mathrm{p}}$ and H represent, respectively, the stencils associated with centred second-order finite difference, 1-point finite element, 2-point finite element, and the 2-D hour-glass matrix. The first three are consistent second-order approximations to the Laplacian, but all are under-diffusive. The hour-glass matrix, on the other hand, leads only to $O(h^2)$ terms and provides some positive 'diffusion' to compensate for the spatial discretization errors of the approximating schemes. It is clearly a higher-order diffusion term, acts like a (balancing) truncation error, and becomes inoperative as $h \rightarrow 0$. It is interesting to note that the results via Fourier analysis are consistent with those of the Taylor-series analysis, i.e. from (12) we have

$$K_{1\mathrm{p}} + \tfrac{1}{6}H = K_{2\mathrm{p}}, \qquad K_{1\mathrm{p}} + \tfrac{1}{2}H = K_{\mathrm{fd}}, \qquad K_{\mathrm{fd}} + \tfrac{1}{2}H = K_{1\mathrm{p}} + H \tag{13}$$

as also suggested in Figure 2(a); perhaps K_{fd} could also benefit from this trick.

The Taylor series results in 3D are

$$K_{\mathrm{fd}} \rightarrow -\nabla^2 T - \frac{h^2}{12}(T_{xxxx} + T_{yyyy} + T_{zzzz}) + O(h^4) \tag{14a}$$

$$K_{1\mathrm{p}} \rightarrow -\nabla^2 T - \frac{h^2}{12}[T_{xxxx} + T_{yyyy} + T_{zzzz} + 6(T_{xxyy} + T_{xxzz} + T_{yyzz})] + O(h^4) \tag{14b}$$

$$K_{2\mathrm{p}} \rightarrow -\nabla^2 T - \frac{h^2}{12}[T_{xxxx} + T_{yyyy} + T_{zzzz} + 4(T_{xxyy} + T_{xxzz} + T_{yyzz})] + O(h^4) \tag{14c}$$

$$H_{xy} \rightarrow 4h^2 T_{xxyy} + O(h^4) \tag{14d}$$

with analogous expressions for the other 2D modes (H_{xz} and H_{yz}), and

$$H \rightarrow O(h^4) \tag{14e}$$

for the 3D wave. Thus H is a presumably less significant (sixth-order) diffusive correction term in 3D. Since the truncation errors for K_{1pt} are virtually unchanged from 2D to 3D except for the additional terms introduced by the third dimension, and the three 2D hour-glass matrices in 3D are all four times as large as they are in 2D, we selected $h/4$ as the tuning parameter for controlling the 2D waves in 3D problems. (As noted earlier, we use $h/2$ as the multiplier for the 3D wave.)

Returning to 2D, we present for further clarity the element level matrices for K_{1p} and H, and their sum, along with the associated 4-patch stencils:

$$K_{1p} = \tfrac{1}{2}\begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \rightarrow \tfrac{1}{2}$$

$$\begin{array}{ccc} -1 & & -1 \\ & 4 & \\ -1 & & -1 \end{array}$$

which clearly annihilates the $2\Delta x$ by $2\Delta y$ wave,

$$H = \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix}(1\ -1\ 1\ -1) = \begin{bmatrix} 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \end{bmatrix} \rightarrow$$

$$\begin{array}{ccc} 1 & -2 & 1 \\ & & \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{array}$$

and

$$K_{1p} + H = \tfrac{1}{2}\begin{bmatrix} 3 & -2 & 1 & -2 \\ -2 & 3 & -2 & 1 \\ 1 & -2 & 3 & -2 \\ -2 & 1 & -2 & 3 \end{bmatrix} \rightarrow \tfrac{1}{2}$$

$$\begin{array}{ccc} 1 & -4 & 1 \\ & & \\ -4 & 12 & -4 \\ 1 & -4 & 1 \end{array}$$

Thus far, the discussion has been restricted to uniform grids and isotropic diffusion, whereas in practice one is often dealing with anisotropic diffusion and non-uniform, distorted elements. In these cases, the selection of the appropriate scalar multipliers for the hour-glass correction matrices is not nearly so straightforward, especially in 3D. Our approach thus far has been the following:

(1) In 2D we use the same hour-glass coefficient discussed above, and treat anisotropy simply by using the average value of the diagonal diffusivities.
(2) In 3D, the same coefficients ($h/2$ etc.) are still used, where h is now an estimate of the average element length and is used for all elements. Anisotropy is treated similarly as in 2D.

Further remarks

(i) These *ad hoc* corrective measures are surely not optimal, and further effort in this area is probably warranted (especially when strong anisotropy is present in the diffusion tensor and/or when large element aspect ratios are employed).
(ii) Similar corrections may also be beneficial in those finite difference methods which are also under-diffusive for short waves.

(iii) Sometimes we do not even use the hour-glass correction, which may be useful as another 'wiggle signal' à la Gresho and Lee;[7] it is only needed when there is significant energy in the short waves (i.e. in 'tough' problems).

4. TIME INTEGRATION

4.1. The basic algorithm

Application of the explicit Euler method to the differential-algebraic system given by (3a), (3b), and (4) leads to

$$u_{n+1} = u_n + \Delta t M^{-1}[f_n - K(u_n)u_n - CP_n] \tag{15}$$

$$T_{n+1} = T_n + \Delta t M_s^{-1}[f_{sn} - K_s(u_n)T_n] \tag{16}$$

$$AP_n = C^T M^{-1}[f_n - K(u_n)u_n] \tag{17}$$

where u_n (satisfying $C^T u_n = 0$) and T_n are available. Clearly (17) must first be solved for P_n before the velocity can be advanced in time. This fact, plus the common vectors appearing on the right-hand sides of (15) and (17), leads to the following algorithmic implementation:

(1) Form part of the acceleration vector (sans pressure gradient):

$$a_n = M^{-1}[f_n - K(u_n)u_n]$$

(2) Form the 'divergence' of this acceleration and solve the consistent discretized Poisson equation for the pressure:

$$AP_n = C^T a_n$$

(3) Update the velocity by integrating the total acceleration:

$$u_{n+1} = u_n + \Delta t(a_n - M^{-1}CP_n),$$

(4) Finally in an uncoupled step, update the temperature field via (16).

This is the basic method. Before embellishing it with two cost-effective modifications, we make the following

Remarks

(i) Since this element is endowed with two 'pressure modes' in 2D and 'many' in 3D (non-trivial vectors, P_m, such that $CP_m = 0$; see Reference 6), some care must be used when solving (17) since A is singular when one or more of these modes is present. The simplest technique is to (properly—see Reference 6) specify a pressure for each mode that exists (*after* verifying that the algebraic system is well-posed, i.e. that the 'consistency constraints' on the velocity BC's are satisfied, à la Sani *et al.*[6], thus rendering A positive-definite. We assume henceforth that pressure modes have been properly disposed of.

(ii) Since the A matrix is symmetric and invariant with time, we have (thus far) used a direct method (Gaussian elimination via the profile, or skyline, method) to solve the discrete pressure Poisson equation. We use an efficient, highly-vectorized code developed by Taylor *et al.*[22] to factor the A matrix in a preprocessor code ($A = LDL^T$, where L is a lower triangular and D a diagonal matrix). The factored matrix is stored in memory (for small 3D problems and essentially all 2D problems) or on disk (for

most 3D problems) for later use by the main code. During the time integration, each pressure update is obtained by reading the disk file (when necessary) and performing one forward reduction and back substitution.

(iii) The pressure obtained from (17) ensures that $C^{\mathrm{T}}U_{n+1}=0$, independently of the magnitude of Δt, another (essential) attribute of the consistent Poisson equation. (The proof of this assertion is direct: insert $P_n = A^{-1}C^{\mathrm{T}}a_n$ from (17) into (15) and multiply the result by C^{T}.)

4.2. The key problem with forward Euler

Many practical solutions of the NS and AD equations are in the regime called advection-dominated: $Re \gg 1$ and/or $Pe \gg 1$. It is in this important regime that the explcit Euler method is at its worst: the stability limit on Δt is so stringent that the basic method could hardly be called viable. In this section we will explain the cause of this instability and in the next, present an effective remedy which we believe is crucial to the cost-effective implementation of this explicit scheme. The entire presentation will be centred around the AD equation, which is (at least for this purpose) a valid prototype of the NS equations.

4.2.1. *Forward Euler and negative diffusivity.* In Appendix I, we review the necessary and sufficient conditions that the following generalized AD equation be well-posed:

$$\frac{\partial T}{\partial t} = \nabla \cdot [(\mathbf{K} \cdot \nabla - \mathbf{u})T] \tag{18}$$

where $\mathbf{K}$ is a symmetric but (generally) anisotropic diffusivity tensor and $\nabla \cdot \mathbf{u} = 0$. The results presented there will be needed below. The next step is to determine the *effective* spatial operator when the forward Euler method is used to integrate (18) in time. Defining the spatial operator

$$L \equiv \nabla \cdot (\mathbf{K} \cdot \nabla - \mathbf{u}) \tag{19}$$

for convenience, (18) becomes

$$\frac{\partial T}{\partial t} = LT \tag{20}$$

Given the solution T_n, at time t_n, the exact solution, $T(t_{n+1})$, at time $t_{n+1} = t_n + \Delta t$ is given (formally) by

$$T(t_{n+1}) = T_n + \Delta t \frac{\partial T}{\partial t}\bigg|_n + \frac{\Delta t^2}{2}\frac{\partial^2 T}{\partial t^2}\bigg|_n + O(\Delta t^3) \tag{21}$$

which, using (20), can be written as

$$T(t_{n+1}) = T_n + \Delta t(LT)_n + \frac{\Delta t^2}{2}\left[(L^2T)_n + \left(\frac{\partial L}{\partial t}T\right)_n\right] + O(\Delta t^3) \tag{22}$$

Now consider integrating (20) with the forward Euler method and ask: 'for what approximation to L, say $\tilde{L}$, is the forward Euler integration of (20), with $\tilde{L}$ in place of L, equivalent, in some sense, to the exact result given by (22)?' The approximate problem is thus

$$T_{n+1} = T_n + \Delta t(\tilde{L}T)_n \tag{23}$$

and we can obtain $\tilde{L}$ to within $O(\Delta t^2)$ by equating T_{n+1} to $T(t_{n+1})$; the result is

$$\tilde{L} = L + \frac{\Delta t}{2}\left(L^2 + \frac{\partial L}{\partial t}\right) \tag{24}$$

where

$$L^2 T \equiv L(LT)$$

and

$$\frac{\partial L}{\partial t} T = \nabla \cdot [((\partial \mathbf{K}/\partial t) \cdot \nabla - \partial \mathbf{u}/\partial t) T]$$

To reveal the basic problem with forward Euler, we now focus on those portions of $\tilde{L}$ which would display second-order dissipative behaviour and assume that $\mathbf{K}$ is independent of time. This leads to

$$\begin{aligned} \tilde{L}T &= LT + \frac{\Delta t}{2} \nabla \cdot [\mathbf{u} \nabla \cdot (\mathbf{u} T)] + O(\Delta t) \\ &= LT + \frac{\Delta t}{2} \nabla \cdot (\mathbf{u}\mathbf{u} \cdot \nabla T) + O(\Delta t) \end{aligned} \tag{25}$$

since $\nabla \cdot \mathbf{u} = 0$. Now let

$$\tau_{ij} \equiv u_i u_j; \tag{26}$$

$\boldsymbol{\tau}$ is a symmetric, positive semi-definite and *singular* matrix, which need not deter us at this point (and will even be used to our advantage later).

Thus we have, finally,

$$\tilde{L}T \simeq \nabla \cdot \left[\left(\mathbf{K} + \frac{\Delta t}{2}\boldsymbol{\tau}\right) \cdot \nabla T - \mathbf{u} T\right] \tag{27}$$

i.e. we see that the exact integration of (18) is 'equivalent', in the sense described above, to $O(\Delta t)$, to the forward Euler time integration of

$$\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T = \nabla \cdot \left(\mathbf{K} + \frac{\Delta t}{2}\boldsymbol{\tau}\right) \cdot \nabla T \tag{28}$$

It also follows that if forward Euler is applied to the *true* spatial operator (19) then the result is equivalent (in the same sense) to the exact integration of

$$\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T = \nabla \cdot \left(\mathbf{K} - \frac{\Delta t}{2}\boldsymbol{\tau}\right) \cdot \nabla T, \tag{29}$$

at least through second-order diffusion terms.

We have thus shown that the straightforward application of forward Euler to the AD equation reduces the effective diffusivity from K_{ij} to $K_{ij} - u_i u_j \Delta t/2$ and that this reduction is ostensibly completely independent of spatial discretization. This, we claim, is the key problem when explicit Euler is used to integrate the AD and/or NS equations.

Remarks

(i) Note that the problem is, as usual, caused by the advection term, i.e. since $\nabla \cdot \mathbf{u} = 0$, $\nabla \cdot \boldsymbol{\tau} \cdot \nabla = (\mathbf{u} \cdot \nabla)^2$.

(ii) A similar reduction in diffusivity is ostensibly inherent in other (but surely not all) explicit schemes.
(iii) If the implicit analogue of forward Euler (backward Euler) is considered instead, the same analysis carries through but with opposite sign; backward Euler increases the effective diffusivity by the same amount that forward Euler reduces it. (This helps explain the well-known 'extreme stability'—via excessive damping—of this scheme.)
(iv) In Section 4.3 we shall show (in the obvious way) how to fix this problem.

We are now ready to address the stability limits associated with explicit Euler.

4.2.2. *Necessary conditions for stability.* Since the effective diffusivity associated with forward Euler is

$$\bar{K}_{ij} \equiv K_{ij} - u_i u_j \, \Delta t/2 \tag{30}$$

it is possible (given that K_{ij} is positive definite) to obtain some *a priori* (i.e. before spatial discretization) stability limits simply by requiring that $\bar{K}_{ij}$ also be positive definite. We thus apply the results of Appendix I to the matrix in (30) to obtain

(i) $$\Delta t < 2K_1/u^2 \tag{31a}$$

(ii) $$\Delta t < 2K_2/v^2 \tag{31b}$$

(iii) $$\Delta t < \frac{K_1 K_2 - K_{12}^2}{K_2 u^2/2 + K_1 v^2/2 - K_{12} uv} \tag{31c}$$

in 2D (3D results are quite lengthy, and seem not to serve any additional useful purpose), which we claim are necessary conditions for the stability of any spatial discretization scheme that does not effectively increase the physical diffusivity via the advection terms. (E.g. this result does not apply to many upwind schemes.)

Remarks

(i) The third inequality in (31) is the limiting one (it includes the first two). The denominator in this inequality is always positive since K_{ij} is positive definite.
(ii) If $K_{12} = 0$, the stability requirement is

$$\Delta t < 1/(u^2/2K_1 + v^2/2K_2) \tag{32}$$

(iii) The allowable Δt decreases rapidly as the velocity increases.
(iv) Explicit Euler is unconditionally unstable in the absence of diffusion (i.e. for pure advection), a well-known fact.

4.2.3. *Necessary and sufficient conditions for FTCS.* Some stronger stability results are available for a special case of spatial discretization using the simplest centred difference approximations (FTCS; forward time, centred space) and with the following additional restrictions:

(i) $\mathbf{u}$ is constant.
(ii) $\mathbf{K}$ is diagonal and constant.
(iii) The mesh is uniform.

Under these conditions we have the following necessary *and* sufficient conditions for the

stability of explicit Euler:[23]

$$\Delta t \leq 1 \Big/ \sum_{j=1}^{n} 2K_j/\Delta x_j^2 \tag{33}$$

and

$$\Delta t \leq 1 \Big/ \sum_{j=1}^{n} u_j^2/2K_j \tag{34}$$

where $n = 1$, 2 or 3 describes the spatial dimensionality.

Remarks

(i) Inequality (34) is a generalization of (32).
(ii) Introducing grid Peclet numbers, $P_j \equiv u_j\, \Delta x_j/2K_j$, it follows that (33) prevails (is more restrictive) when all $P_j < 1$, (34) prevails when all $P_j > 1$, and both inequalities are required otherwise.
(iii) Inequality (34) is especially restrictive when any (or all) $P_j \gg 1$; e.g. in 1D, it can be expressed as $c \leq 1/P$, where $c = u\, \Delta t/\Delta x$ is the Courant number.
(iv) Although the corresponding results are not yet available for the 9-point stencil (2D; 27-point in 3D) associated with our modified FEM scheme, we believe that they will be found to be not too different from these.

4.3. *Balancing tensor diffusivity*

Since the basic problem with forward Euler integration is the reduction in the effective diffusivity, and the concomitant stringent stability limit, it seems natural to consider an *a priori* modification which, theoretically at least, will cancel this deleterious portion of the truncation error. To this end, the technique we have adopted is simply to augment the physical diffusivity by 'exactly' that amount (implicitly) subtracted via explicit Euler, i.e. we use $(K_{ij} + u_i u_j\, \Delta t/2)$ as the diffusivity (or viscosity, in NS) where (for our purposes) u_i is taken to be the (spatially and temporally varying) centroid advection velocity referred to earlier (see (6c)') and the correction is applied element-wise at each time step. For AD, it simply means that we solve (28) rather than (18). We call this simple trick 'balancing tensor diffusivity' (BTD) and note that:

(1) It is not new. BTD has been previously discussed and applied to NS by Dukowicz and Ramshaw[24] using a different spatial discretization (recall that our derivation suggests that its utility is essentially independent of spatial discretization in that it was derived from the continuum equation in space).
(2) Since cross-derivatives must be approximated, it cannot be used on the simplest (FTCS) finite difference mesh (5-point stencil in 2D, 7-point in 3D); the spatial discretization necessarily involves at least a 9-point stencil in 2D and a 19-point stencil in 3D. (The FEM scheme uses a 9-point stencil in 2D and a 27-point stencil in 3D, the additional 8 nodes representing the corners of a 'brick' composed of 8 elements.)
(3) In the hyperbolic limit ($K_{ij} = 0$), the scheme becomes a member of the popular family known as Lax–Wendroff methods,[25] the 1D version of which is also called Leith's method.[26]
(4) The implementation of BTD via 1-point quadrature is conveniently accomplished via the C-matrix, e.g. with $\tau_{ij} \equiv \bar{u}_i \bar{u}_j$,

$$\int_E \frac{\partial \varphi_i}{\partial x_k}\, \tau_{kl}\, \frac{\partial \varphi_j}{\partial x_l}\, T_j\, \mathrm{d}\Omega = \bar{u}_k \bar{u}_l C_{ie}^{(k)} C_{je}^{(l)} T_j/\Omega_e \quad \text{(no sum on } e\text{)}.$$

(5) If $\mathbf{K}$ is time-dependent (perhaps implicitly), the BTD term would presumably be augmented by $(\Delta t/2)\,\partial\mathbf{K}/\partial t \simeq (\mathbf{K}_n - \mathbf{K}_{n-1})/2$. (We have not explored this aspect of the 'problem' with forward Euler.)

Further discussion and analysis of BTD and its effects will be presented in subsequent sections.

4.3.1. *Stability of the improved scheme.* Since the negative diffusivity of explicit Euler is 'cancelled' via BTD, it follows that the stability limits should be more generous, at least for advection-dominated flows, and this in fact occurs. In 1D, the analysis is simple and the results precise; in multi-dimensions, we have not yet succeeded in completing the stability analysis, although we have a large database of numerical experience which have proved useful. The bottom line, for advection-dominated flows, will be seen to be basically this: the stability limit is set by the well-known CFL condition for hyperbolic problems (advection-dominated flow is 'nearly' hyperbolic): the fluid should not move more than one grid point per time step. For diffusion-dominated situations, ($Re \ll 1$, $Pe \ll 1$) the stability limit is believed to be similar to that given by (33); but see Reference 23.

For the constant coefficient AD equation in 1D, spatially discretized via lumped mass FEM and linear basis functions (or, equivalently, FTCS), the (uniform mesh) stability limit is[23]

$$\Delta t \leq \Delta x^2/\{K(1+\sqrt{[1+(u\Delta x/K)^2]}\} \tag{35a}$$

or, in terms of the Courant number and the grid Peclet number

$$c \leq 2P/(1+\sqrt{[1+4P^2]}) \tag{35b}$$

For $P \to 0$, this yields the standard diffusion-limit ($\Delta t \leq \Delta x^2/2K$), whereas, if $K \to 0$ ($P \to \infty$), we have $c \leq 1$ as the stability limit, where we recall (cf. (34)) that the unmodified scheme is unconditionally unstable in this hyperbolic limit.

In 2D and 3D, the detailed stability analysis has thus far proved intractable (with or without BTD) and we have typically had to be content with applying (33) in general and (35) in regions where the flow is approximately 'locally one-dimensional', although we do have the following necessary and sufficient condition for pure advection: $\sum_i c_i^2 \leq 1$. Fortunately, this approximate technique usually works reasonably well (of course the grid Peclet number is replaced by the grid Reynolds number for NS), although some trial and error is sometimes required.

4.3.2. *Accuracy.* Consider the constant coefficient AD equation with BTD in 1D,

$$\frac{\partial T}{\partial t} + u\frac{\partial T}{\partial x} = (K + u^2\Delta t/2)\,\partial^2 T/\partial x^2 \tag{36}$$

which is to be integrated in time via explicit Euler and discretized in space via FEM using linear basis functions (with lumped mass). If we define the local truncation error (LTE) as $T_j^{n+1} - T(j\Delta x, (n+1)\Delta t)$, where j is the grid point index, n is the time level, and $T(x, t)$ denotes the exact solution, a Taylor series analysis in space and time leads to

$$\text{LTE} = \frac{\Delta t\,\Delta x^2}{12}(KT_j'''' - 2uT_j''') + \frac{u^2\Delta t^2\Delta x^2}{24}T_j'''' - \frac{K\Delta t^2}{2}(KT_j'''' - 2uT_j''') + O(\Delta t\,\Delta x^4) + O(\Delta t^3) \tag{37}$$

where $T_j' \equiv \partial T(x_j, n\Delta t)/\partial x$, etc.

Remarks

(1) The local error is $O[\Delta t(\Delta x^2+K\,\Delta t+\Delta t^2)]$. The global error (in time) is obtained by omitting the Δt factor; thus,
(2) If $K=0$, the global error is $O(\Delta x^2+\Delta t^2)$, *à* la Lax–Wendroff.[27]
(3) If $K\neq 0$ and fixed, the global error is also $O(\Delta x^2+\Delta t^2)$ when stable, since then $K\Delta t\leq O(\Delta x^2)$.
(4) In both cases, this is usually regarded as second order in space and time, although the definition of 'order' may be somewhat ambiguous.
(5) The performance of the scheme away from these asymptotic limits (especially $P\gg 1$) is probably more important, in a practical sense.
(6) Similar remarks apply (we hope) to multi-dimensions.

4.3.3. *Damping and phase speed.* In this section we summarize the performance of the pure advection scheme (with BTD) in one and two dimensions.

4.3.3.1. One dimension

The analysis begins with the pure advection form of (36) for the continuum,

$$\frac{\partial T}{\partial t}+u\frac{\partial T}{\partial x}=0 \tag{38}$$

and its discretized equivalent which, using BTD (otherwise it is unstable) on a uniform mesh, is

$$\frac{T_j^{(n+1)}-T_j^{(n)}}{\Delta t}+\frac{u}{2\Delta x}(T_{j+1}^{(n)}-T_{j-1}^{(n)})=\frac{u^2\Delta t}{2\Delta x^2}(T_{j-1}^{(n)}-2T_j^{(n)}+T_{j+1}^{(n)}) \tag{39}$$

If a single Fourier mode, e^{ikx}, is used as an initial condition, the exact solution of (38) is

$$T(x,t)=e^{ik(x-ut)} \tag{40}$$

a pure translation at speed u and unit amplitude (the true solution has no damping). If the same initial condition, expressed now as $e^{ikx_j}=e^{ikj\Delta x}$, is used in (39), the solution can be expressed as

$$T_j^{(n)}=r^n e^{ik(j\Delta x-u_p n\Delta t)} \tag{41}$$

where r (the amplitude factor) and u_p (the approximation to the phase speed, u) are obtained by inserting (41) into (39); the results are

$$r=[1-c^2(1-c^2)(1-\cos\beta)^2]^{1/2} \tag{42a}$$

and

$$u_p/u=\frac{1}{\beta c}\tan^{-1}\left[\frac{c\sin\beta}{1-c^2(1-\cos\beta)}\right] \tag{42b}$$

where $\beta\equiv k\Delta x$ and $c\equiv u\Delta t/\Delta x$ are the dimensionless wave number and the Courant number, respectively. We first examine the asymptotic behaviour by fixing k and letting $\Delta t\to 0$ and $\Delta x\to 0$ in such a way that the computation would remain stable (i.e. we require $c\leq 1$). The results are

$$r\simeq 1-c^2(1-c^2)\beta^4/8 \tag{43a}$$

and

$$u_p/u\simeq 1-(1-c^2)\beta^2/6 \tag{43b}$$

where higher order terms have been neglected.

Remarks

(i) Since $r \sim (k\Delta x)^4$, the scheme is spatially fourth-order dissipative (this result is also available from (37)).
(ii) The scheme is second-order accurate in phase speed. As $\Delta t \to 0$, the phase speed is always lagging ($u_p < u$); it is given by $u_p/u = \sin\beta/\beta$.
(iii) Finite Δt helps compensate for this lagging phase speed by introducing leading phase speed ($c^2\beta^2/6$).
(iv) Damping is absent and the phase speed is exact if $c = 1$ (from (42)).

Figures 4 and 5 show how r and u_p/u vary over the entire range of β and c, with the key observation being that, over all resolvable wavelengths ($0 \le k \le \pi/\Delta x$) time integration error ($0 < c < 1$) increases both damping and phase speed, the latter helping to compensate for the lagging error caused by spatial discretization. Also, for $1/\sqrt{2} \le c \le 1$ and $k\Delta x = \pi$, $u_p/u = 1/c$ or $u_p = \Delta x/\Delta t$; the $2\Delta x$ wave moves one grid point per time step. Not only does BTD cause a previously unstable method to be stable (for $c \le 1$), it also enhances the accuracy as Δt is increased toward the (Courant) stability limit.

These desirable effects are further demonstrated in Figure 6 which shows some results from the AD equation solved on the unit span with 100 (lumped) linear elements (FTCS) and periodic boundary conditions. The initial condition is a Gaussian wave with $\sigma = 2\Delta x = 0{\cdot}02$ centred at $x = 0{\cdot}5$. The velocity is $1{\cdot}0$ and the diffusion coefficient is $0{\cdot}001$ which gives a Peclet number, $Pe \equiv u\sigma/\kappa$, of 20 and a grid Peclet number of 5. The exact solution, $T(x, t) = e^{-(x-x_0-ut)^2/(2\sigma^2+4\kappa t)}/\sqrt{(1+2\kappa t/\sigma^2)}$, is shown at $t = 1{\cdot}0$ (dotted) along with three numerical results: (1) FTCS using a small Δt to (nearly) eliminate time truncation error

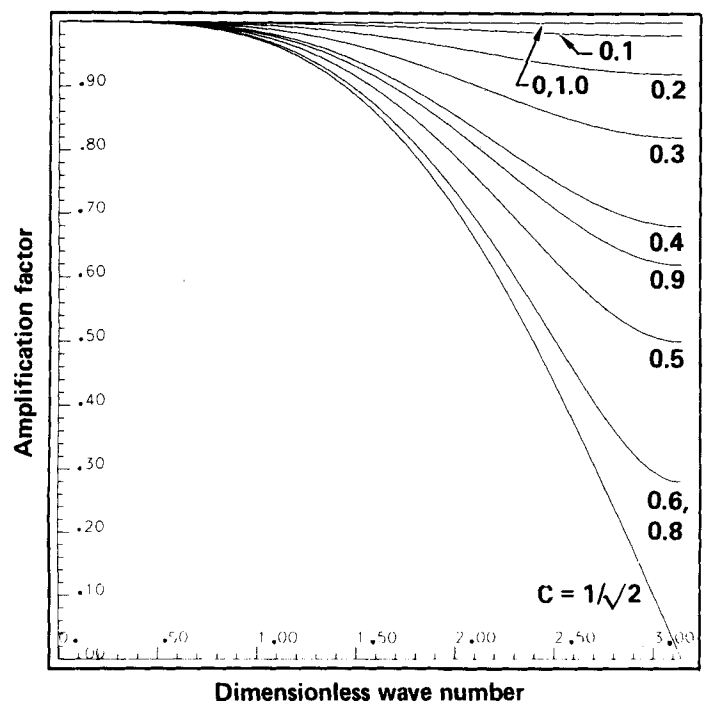

Figure 4. 1D amplitude factor vs wave number at several values of Courant number

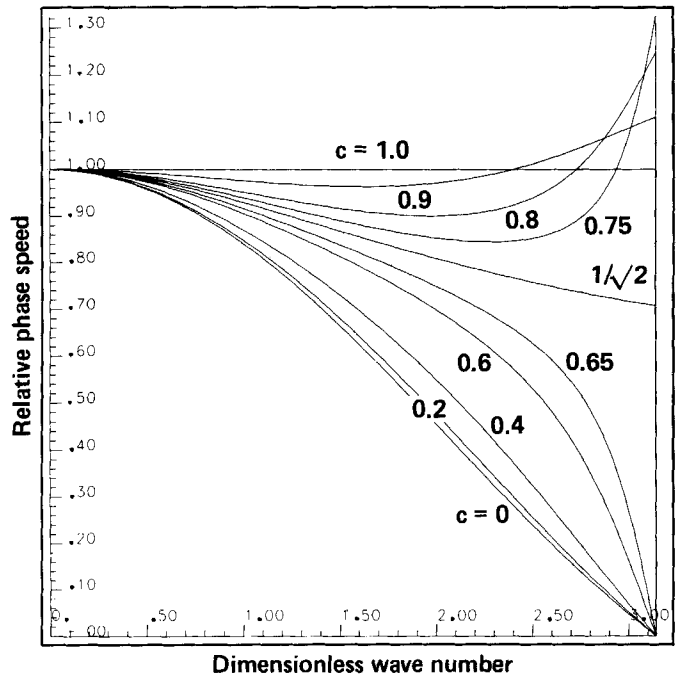

Figure 5. 1D phase speed vs wave number at several values of Courant number

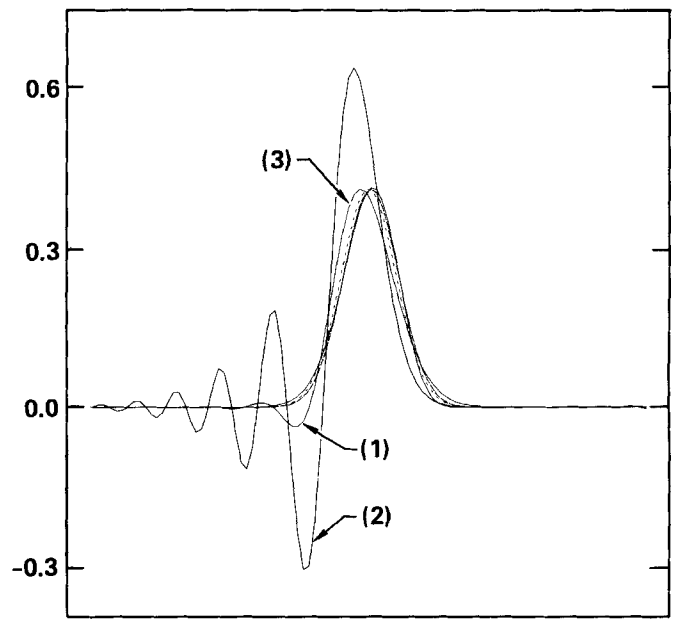

Figure 6. Advection and diffusion of a Gaussian wave

($\Delta t = 0{\cdot}0002$ which is 10 per cent of the stability limit, giving $c = 0{\cdot}02$), (2) FTCS at its stability limit ($\Delta t = 0{\cdot}002$, $c = 0{\cdot}2$) and (3) FTCS plus balancing diffusion at its larger stability limit ($\Delta t = 0{\cdot}00905$, $c = 0{\cdot}905$). Clearly the result using balancing diffusion is superior and at reduced cost owing to the larger time step. The result in curve (2) shows the typical oscillatory solution associated with *pure* advection on a too coarse mesh, thus corroborating the earlier analysis which predicts an effective diffusivity of zero at the stability limit.

4.3.3.2. Two dimensions

If the foregoing analysis is repeated for the 2D AD equation, using lumped mass, 1-point quadrature, and BTD, it becomes long and tedious and the results depend on 4 parameters ($\beta_1 \equiv k_1 \Delta x$, $\beta_2 \equiv k_2 \Delta y$, $c_1 \equiv u\Delta t/\Delta x$ and $c_2 \equiv v\Delta t/\Delta y$) rather than just two. Here we will only present results for a special case in which the dimensionality is reduced from four to two: we take $\beta_1 = \beta_2 = \beta$ and $c_1 = c_2 = c$, which corresponds to a velocity directed 'diagonally' through the grid points (recall Δx and Δy are constant) and a wave number vector directed at the angle $(\pi/2 - \theta)$, where θ is the angle of the velocity vector ($\tan\theta = v/u = \Delta y/\Delta x$). The results are (to lowest order)

$$r = 1 - c^2\beta^4 \tag{44a}$$

and

$$u_p/u = 1 - (5 - 8c^2)\beta^2/12 \tag{44b}$$

which are similar to those in 1D.

Remarks

(i) Again the scheme is fourth-order dissipative.
(ii) Again the phase speed is second-order accurate and is spatially lagging, but less so for finite Δt (time truncation error again reduces phase error).
(iii) Ostensibly the situation is not too different in 3D.

Leaving the asymptotic region, Figures 7 and 8 show the amplitude factor and phase speed over the full range of β and c (still for the special case) and the following remarks apply:

(i) The behaviour is generally less desirable than in 1D.
(ii) The stability limit for this case is $c \leq 1/\sqrt{2}$. (It is $\sum c_i^2 \leq 1$ in the more general case.)
(iii) Time integration error improves the phase speed, but much less so than it does in 1D.
(iv) Damping is slight for short waves and totally absent for the infamous $2\Delta x$ (by $2\Delta y$) wave, which is neither damped nor advected.

This last item, a direct consequence of the problem with diffusion discussed earlier, is ostensibly a cause for concern—at least if pure advection simulations are of interest. (In our experience with advection-dominated flows rather than pure advection, the hour-glass correction to the diffusion terms has generally been effective in controlling short waves.) In this regard then, we will show the results when the hour-glass correction is appended to the BTD diffusion term (both temporal and spatial truncation error corrections applied simultaneously to the same term). The 'diffusion coefficient' that multiplies the hour-glass correction matrix is naturally different for pure advection. Although we have not yet actually implemented such a correction term in our codes, some preliminary analysis suggests that the coefficient should be of the form $\sigma\Delta x\,\Delta y(c_1 + c_2)^2/\Delta t$, where σ is dimensionless. For $\sigma = 1/32$, the results for the same special case are shown in Figures 9 and 10 for r and u_p/u,

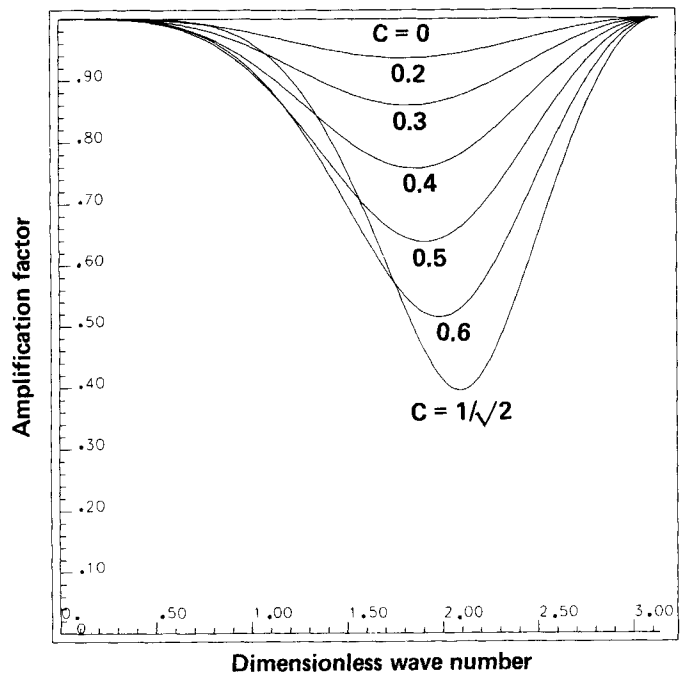

Figure 7. 2D amplitude factor vs wave number at several values of Courant number

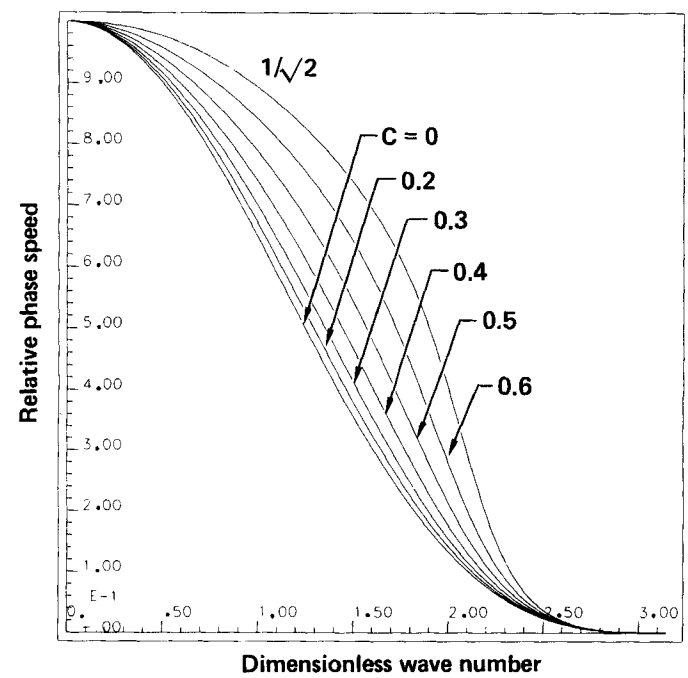

Figure 8. 2D phase speed vs wave number at several values of Courant number

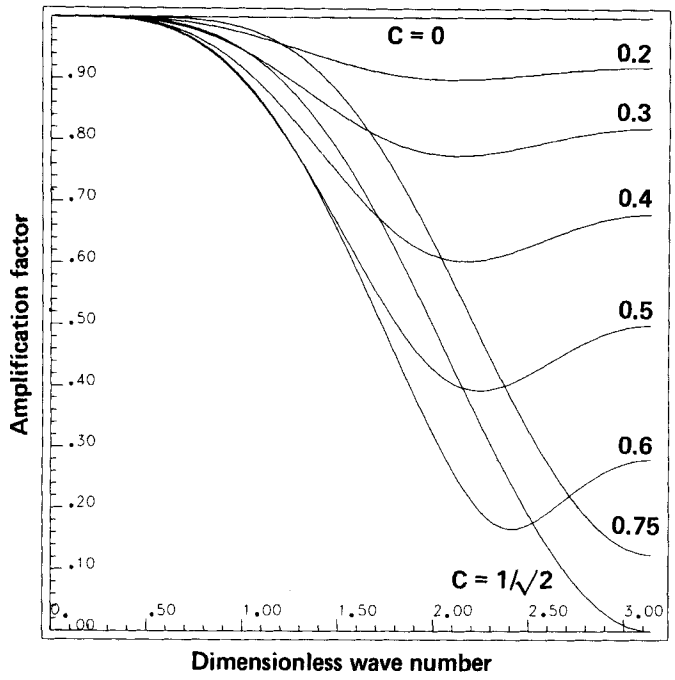

Figure 9. Same as Figure 7 except with hour-glass correction

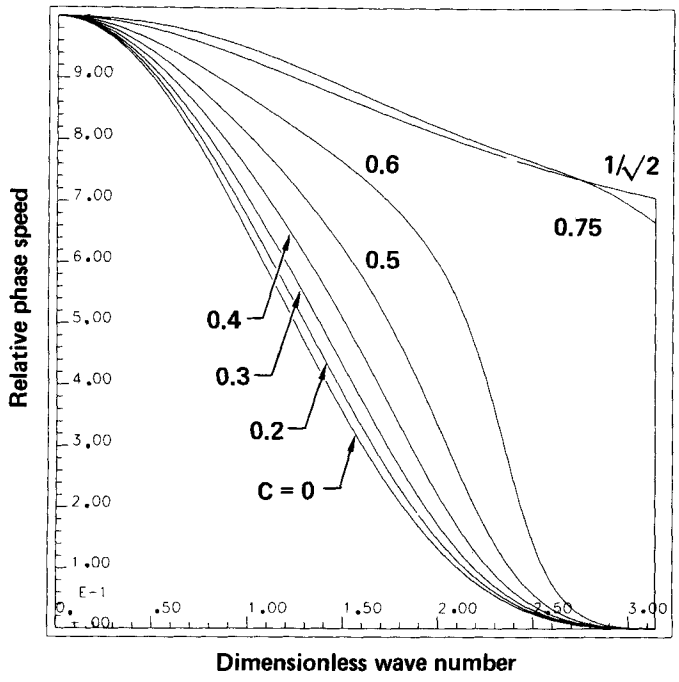

Figure 10. Same as Figure 8 except with hour-glass correction

respectively, wherein we observe:

(1) The stability limit is slightly improved ($c \leq \sim 0{\cdot}75$)—larger or smaller values of σ seem to reduce the stability.
(2) Damping of the short waves is, as desired, recovered. (Now, as in 1D, if $c = 1/\sqrt{2}$, the $2\Delta x$ wave is completely damped in a single step.)
(3) The phase speeds are further improved as Δt is increased toward the stability limit.
(4) A similar diffusion-like term, with a scalar coefficient more like $(u^2+v^2)\,\Delta t/2$, was suggested 20 years ago by Lax and Wendroff,[27] albeit for different reasons, since their scheme does damp the $2\Delta x$ wave.

By studying similar curves for other values of σ, we observed (for this special case of one β, one c) that $\sigma = 1/32$ appears to be nearly optimum regarding stability, damping and phase speed. Clearly, however, more work in this area is required if pure (or nearly pure) advection simulations are of interest in situations which are prone to generate short waves.

In concluding this section, we remark that we have solved the (pure advection) 'rotating cone' problem[28,29] on a uniform grid using BTD (necessarily) but no hour-glass correction. The results were reassuring in that the fidelity was essentially equivalent to that observed by Orzag[28] when Arakawa's second order scheme is used, i.e. noticeable phase error and dispersion, but no obvious artificial diffusion, crosswind (see below) or otherwise. Also, the accuracy improved as the Courant number was increased toward the (poorly defined, in this case) stability limit, in accord with the theory. Thus, although the advection scheme we employ is far from perfect, and requires the use of (easy to apply) truncation error correction terms, it has nevertheless proved quite useful and, especially, cost-effective.

4.3.4. *Steady state, streamline upwinding, wiggles.* Since the 'problem' with explicit Euler and its correction via BTD were analysed using a Taylor-series expansion in time, it follows that neither the correction nor the underlying analysis apply to steady-state flows (for which the 'problem' does not exist). The question that naturally arises then is: what happens if a simulation in which BTD is employed approaches or attains a steady state? The first reaction might well be that the BTD terms are completely inappropriate for such cases and if they are employed: (1) the scheme is overly diffusive and (2) the steady-state results are a function of Δt(!).[30] Both of these assertions are true, yet we have found (and will demonstrate) that BTD is also useful (and always cost-effective) for steady-state simulations. Thus, in this section we will attempt to justify the use of BTD for simulations which tend toward steady state, as well as re-addressing the subject of wiggles.[7]

First, and most importantly, we point out that the added tensor diffusivity is operational *only* in the streamline direction (as is indeed the problem with explicit Euler); there is *no* deleterious crosswind diffusion which has been the bane of many upwinded methods. (See also Brooks and Hughes,[10] who advocate a very similar form of streamline upwinding as a means of controlling the wiggles sometimes caused by the advection terms in GFEM models, quite independently of time integration effects.) This can be demonstrated in 2D most simply by applying a pure rotational transformation to the Cartesian tensor diffusivity such that the resulting diffusivity is represented in the streamline (and normal) co-ordinate system. Thus, with $K_{ij} \equiv u_i u_j\,\Delta t/2$, consider the transformation

$$\hat{\mathbf{K}} = \mathbf{R}^{\mathrm{T}}\mathbf{K}\mathbf{R} \tag{45}$$

where

$$\mathbf{R} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} = \frac{1}{|\mathbf{u}|}\begin{bmatrix} u & -v \\ v & u \end{bmatrix}$$

is the appropriate rotation matrix (to streamline and normal co-ordinates; $\tan\theta \equiv v/u$) and $\hat{\mathbf{K}}$ is the representation of $\mathbf{K}$ in the rotated co-ordinate system. The result is

$$\hat{\mathbf{K}} = \begin{bmatrix} (u^2+v^2)\,\Delta t/2 & 0 \\ 0 & 0 \end{bmatrix}$$

showing added diffusivity *only* in the streamline direction, and is related to the singularity of τ mentioned earlier—after (26). (The analogous result also obtains in 3D.) By contrast, if simple upwinding of the form (for $u_i > 0$) $K_{ij} = (u_i\,\Delta x_i/2)\delta_{ij}$ where δ_{ij} is the Kronecker delta is employed, the transformed result is

$$\hat{\mathbf{K}} = \frac{1}{2(u^2+v^2)} \begin{bmatrix} u^3\,\Delta x + v^3\,\Delta v & uv(v\,\Delta y - u\,\Delta x) \\ \text{sym.} & uv(u\,\Delta y + v\,\Delta x) \end{bmatrix}$$

which can cause excessive diffusion *across* streamlines, e.g. when $u \simeq v$ (cf. also References 31 and 32); here $\hat{\mathbf{K}}$ (and, of course, $\mathbf{K}$) is positive definite rather than positive semi-definite (i.e. it is diffusive in *all* directions).

Secondly, it is of interest to attempt to estimate the 'effective Reynolds number' in the case of steady-state (or slowly time-varying) simulations; i.e. even *streamline* upwinding could (will) still be called deleterious by some purists. To this end we write $Re \equiv u_0 L/\nu_{\text{eff}}$ with $\nu_{\text{eff}} \equiv \nu + u_\tau^2\,\Delta t/2$, where u_τ is the velocity along a streamline. (The Reynolds number, from the above discussion, is only affected *along* streamlines; thus we are examining the worst case.) The effective local streamline Reynolds number is therefore

$$Re = \frac{Re_0}{1 + u_\tau^2\,\Delta t/2\nu} = \frac{Re_0}{1 + \frac{1}{2}\,Re_0 \dfrac{u_\tau}{u_0}\dfrac{\Delta x}{L}\dfrac{u_{\tau\Delta t}}{\Delta x}} \tag{46}$$

where $Re_0 \equiv u_0 L/\nu \gg 1$ is the nominal Reynolds number and Δx is a generic grid spacing.

Consider first a region of 'high-velocity' flow, i.e. where $u_\tau \simeq u_0$. If, in addition, the maximum permissible time step is used (the normal goal), we have $u_\tau\,\Delta t/\Delta x \cong 1$ so that

$$Re \simeq Re_0 \Big/ \left(1 + \tfrac{1}{2} Re_0 \frac{\Delta x}{L}\right)$$

a result ostensibly equivalent to that obtained with *simple* upwinding, wherein $\nu_{\text{eff}} = \nu + u\Delta x/2$. If $Re_0\,\Delta x/L \gg 1$ (large grid Reynolds number), the effective Reynolds number is clearly much less than the desired one (but only in the streamline direction, an *important* result *not* obtained with simple upwinding). An example of this 'worst case' would occur on a streamline just below the lid in the lid-driven cavity problem. In cases like this, however, it seems that the results of the simulation are not seriously degraded as long as $Re \gg 1$, i.e. if the flow along such a streamline is *still* advection-dominated. This merely requires $\Delta x/L \ll 1$, a reasonable requirement in any case. Another important, and quite different, case of interest is related to low-speed regions of the domain, such as a recirculation eddy where the true *local* Reynolds number may be much smaller than Re_0 and significant streamline upwinding could ostensibly be quite harmful. In these cases, however, we would usually have $u_\tau/u_0 \ll 1$ and, thus, the local Courant number $u_\tau\,\Delta t/\Delta x$, is also $\ll 1$. Since also $\Delta x/L \ll 1$, we can see from (46) that the reduction of Re from Re_0 will be much less than before. This interpretation helps to account for the success (on good meshes) of the streamline upwinding technique in capturing the details of fine structure associated with low-speed secondary

flows, although the primary argument is still the complete lack of artificial diffusion normal to streamlines. We will demonstrate both of these points in Part 2 of this paper.

What about wiggles? Is streamline upwinding just another wiggle suppressant,[7] generating smooth but inaccurate results? Our answer is simple to state: yes and no. It does a 'good' job of wiggle suppression relative to GFEM, but a 'poor' job relative to the more conventional 'simple' upwind methods (e.g. donor cell or hybrid schemes). Consistent with the philosophy expressed by Gresho and Lee,[7] we are suspicious of schemes which never wiggle; thus, to conclude this section we will show that the streamline upwind scheme (i.e. BTD applied to steady flows) does send out wiggle signals for problems too difficult for the selected mesh. but they may not be very strong.

We begin with the classic 1D prototype model; steady advection–diffusion with Dirichlet boundary data:

$$u\frac{dT}{dx} = K\frac{d^2T}{dx^2}, \qquad 0<x<l \tag{47}$$

$$T(0)=1, \qquad T(l)=0$$

For $Pe \equiv ul/K \gg 1$, the solution to (47), which is

$$T = \frac{1-e^{-Pe(1-x/l)}}{1-e^{-Pe}} \tag{48}$$

undergoes all of its interesting variation within a distance $O(l/Pe)$ of the downstream boundary and is the classic 'tough problem' that can give a numerical scheme a bad case of the wiggles. The reasons that we even consider such a model are two: (i) it represents sort of a limiting case of a multidimensional situation in which there exists a strong gradient (in the transported variable) along a streamline, and (ii) it is very simple to analyse. (The former reason is clearly more important.)

The discretized form of (47), using BTD, is

$$\frac{u}{2\Delta x}(T_{j+1}-T_{j-1}) = \frac{(K+u^2\,\Delta t/2)}{\Delta x^2}(T_{j+1}-2T_j+T_{j-1}) \tag{49}$$

for $j=1,2,\ldots,N-1$, $T_0=1$, $T_N=0$, and $\Delta x = l/N$. The solution of this difference equation is

$$T_j = \frac{1-\beta^{N-j}}{1-\beta^N}, \qquad j=0,1,\ldots,N \tag{50}$$

where $\beta \equiv [1-P(1-c)]/[1+P(1+c)]$, $P \equiv u\,\Delta x/2K$ is the grid Peclet number, and $c \equiv u\,\Delta t/\Delta x$ is the Courant number (which is bounded by (35b) for stability, i.e. we consider (49) and (50) to have been obtained by a time integration to steady state).

Remarks

(*i*) $c=0$ (i.e. the absence of time truncation error) corresponds to GFEM (or FTCS) and $\beta=(1-P)/(1+P)$; this is the infamous wiggle maker.

(ii) The effective grid Peclet number is $P=P/(1+cP)$, so that $c=1$ corresponds to pure upwinding, the infamous wiggle suppressor.

(iii) $\beta=0$ at $c=c_{max}\equiv 1-1/P$; $T_j=1$ for all $j<N$, another method of wiggle suppression.

In Figure 11 are shown some plots of (50) for $Pe = 480$, $l = N = 24$ ($P = Pe\,\Delta x/2l = 10$), for which the outflow boundary layer thickness is $\delta = l/Pe = 0{\cdot}05$ and thus not resolvable by the chosen mesh (for which $\Delta x = 1$). The stability limit (cf. 35b) is $c \simeq 0{\cdot}95$ and we present results for $0 \leq c \leq 0{\cdot}9$. The largest wiggles are associated with GFEM ($c = 0$) and the wiggle amplitude is seen to decrease monotonically and rapidly as c increases. The main result is that there are indeed wiggles for $0 < c < c_{max}$, even though they are small compared to those with no BTD.

Moving to 2D, we now consider the steady solution of (1b) for several Peclet numbers with a prescribed velocity field corresponding to flow over a step at $Re = 200$ (cf. Reference 33). The velocity field shown in Figures 12(a) and 12(b) was used to solve the steady AD equation for several values of Pe: $Pe = Re = 200$, $Pe = 2000$ and $Pe = 20{,}000$, both via GFEM (4-node element) and via our 1-point quadrature scheme with BTD and diffusive hour-glass correction, using $\Delta t = 0{\cdot}01$ (about 1/2 of the experimentally-determined stability limit). The boundary conditions for this hot step problem are: $T = 0$ at the inlet, $T = 1$ on the three sides of the step, and $\partial T/\partial n = 0$ elsewhere. Since the mesh was designed to provide a good velocity solution at $Re = 200$, the results from the AD equation at $Pe = 200$ should be about equally good, and indeed this is verified in Figure 12(c) and 12(d), for which GFEM and our current scheme show good agreement. (Also, the result with $\Delta t = 0{\cdot}02$ is essentially (visually) the same as that presented, showing that streamline upwinding has a negligible effect, even though $u_\tau^2\,\Delta t/2\kappa$ is as large as 5–10.) Figures 12(e)–12(h) show the corresponding results when the mesh is 'pushed beyond its design point': namely for $Pe = 2000$ and $Pe = 20{,}000$, respectively. The results more or less corroborate those from 1D; the GFEM scheme generates much larger wiggles upstream of the step than our current scheme, but the latter scheme does indeed send out a recognizable wiggle warning.

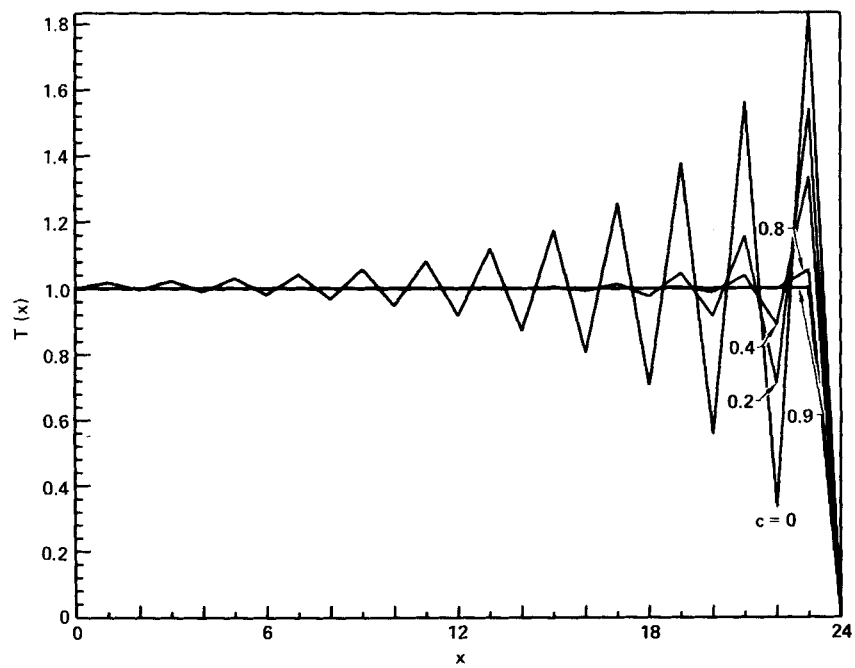

Figure 11. 1D steady-state advection–diffusion for several values of Courant number

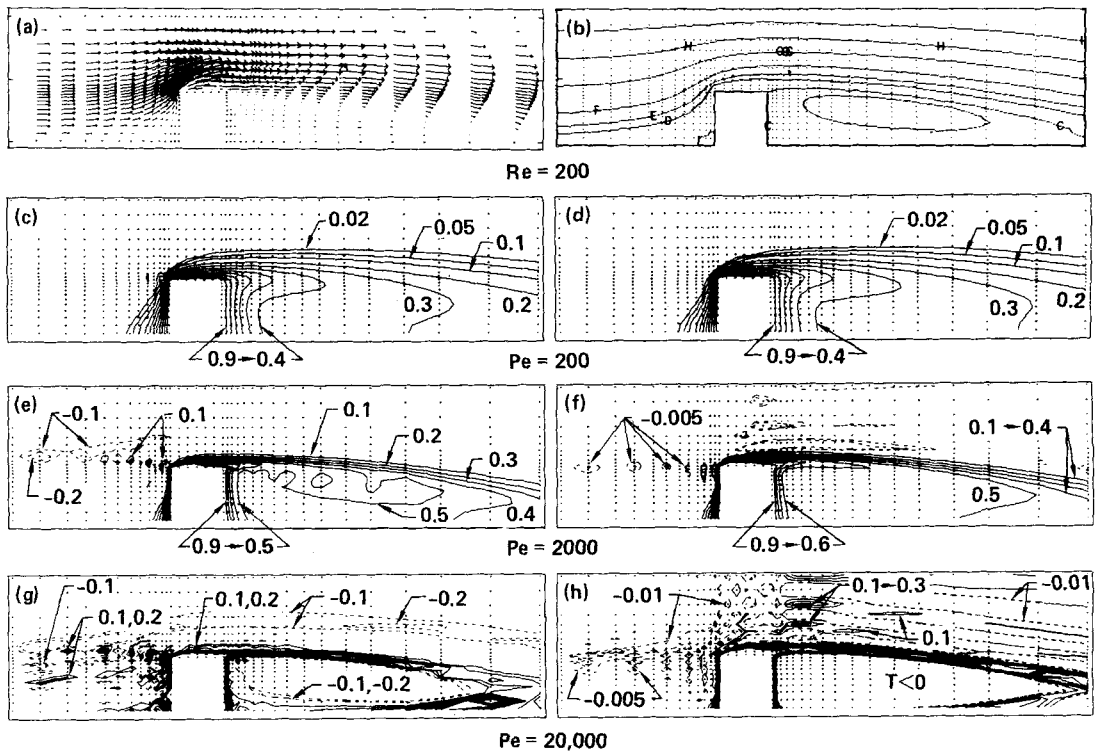

Figure 12. 2D advection–diffusion for flow over a step: (a) and (b) give the flow field; (c), (e) and (g) are from GFEM; (d) (f) and (h) are from the present scheme

Additional remarks:

(i) In these Figures, both GFEM and our current scheme give $T>0$ in the downstream recirculating eddy for $Pe \leq 2000$, and both yield the spurious result (and strong wiggle signal) $T<0$ in this eddy for $Pe = 20{,}000$ ($T_{\min}$ is ~ -2 for GFEM and $\sim -1{\cdot}6$ for our modified scheme). Simple upwind schemes will never yield $T<0$, consistent with their too-strong tendency to suppress wiggles by adding numerical diffusion.

(ii) The hour-glass correction term (diffusion term only) was 'essential' for the high Pe runs; without it, the entire region above the step was even *more* polluted with $2\Delta x$ by $2\Delta y$ oscillations, which are only partially damped even with the hour-glass corection when the diffusion coefficient is small. The fact that these spurious waves are basically normal to the streamlines above the step is a manifestation of the absence of crosswind diffusion via BTD. These oscillations would probably be further reduced if we had also applied the 'advection' hour-glass correction (discussed earlier) to the BTD terms.

(iii) The steady-state solution via BTD seems to be a rather weak function of Δt, even for advection-dominated flow.

(iv) If we had not used BTD, the (approximate) stability limits on Δt (cf. (34)) would have required time steps of $\Delta t \simeq 0{\cdot}001$ ($Pe/200$), i.e. an order of magnitude smaller than that used for $Pe = 200$ and 3 orders of magnitude smaller for $Pe = 20{,}000$.

This last remark emphasizes the reason that we continue to employ BTD even when

steady-state solutions are sought or attained: the calculations without BTD would cost about P times as much, where P is the largest grid Peclet number (or Re for NS), but would not be noticeably more accurate.

In closing we state: in practice, we often see wiggles when a difficult problem is first attacked (for time-dependent as well as steady-state calculations); as in GFEM, we regard these signals as a warning regarding the quality of our mesh design.

4.4. *Internal gravity waves*

In dealing with stably-stratified flows, it is well known[34] that an additional phenomenon can occur: internal gravity waves. Since we have had some experience, and some difficulty,[35] with simulations of this type, it seems fruitful to summarize the type of problem (instability) that can occur and to describe a successful method for overcoming this problem (another modification to the time integration scheme).

A simple, but non-trivial example can be generated by considering a 1D linearized subset of the Boussinesq equations involving only vertical velocity, $v(x, t)$, and temperature, $T(x, t)$—perturbations about a motionless base state with a linearly increasing temperature:

$$\frac{\partial v}{\partial t} - \gamma g T = \nu \frac{\partial^2 v}{\partial x^2} \tag{51a}$$

$$\frac{\partial T}{\partial t} + \beta v = \kappa \frac{\partial^2 T}{\partial x^2} \tag{51b}$$

where $\beta \equiv dT_0/dy > 0$ is a constant (the base state temperature gradient). Setting initial conditions $v(x, 0) = v_0 \cos kx$ and $T(x, 0) = 0$, where k is an arbitrary horizontal wave number, the solution of the initial value problem (or the corresponding periodic initial-boundary value problem) is

$$v = v_0\{\cos \omega t + [k^2(\kappa - \nu)/2\omega] \sin \omega t\} \cos kx e^{-k^2(\nu+\kappa)t/2} \tag{52a}$$

$$T = -(\beta v_0/\omega) \sin \omega t \cos kx e^{-k^2(\nu+\kappa)t/2} \tag{52b}$$

where

$$\omega^2 \equiv N^2 - N_c^2 > 0 \tag{52c}$$

$N \equiv \sqrt{(\beta\gamma g)}$ is the buoyancy frequency

$N_c \equiv k^2 |\kappa - \nu|/2$ is a cut-off frequency

Equation (52) applies for $N > N_c$; if $N < N_c$, the solution to (51) is one of montonic decay (overdamped) rather than damped oscillatory motion (underdamped). Defininig $\omega^2 \equiv N_c^2 - N^2 > 0$ for this case, the trigonometric functions in time are replaced by hyperbolic functions in (52a, b).

Remarks

(i) Oscillatory motion (i.e. internal gravity waves) exists only if $N > N_c$; i.e. if $k < k_c \equiv \sqrt{(2N/|\kappa - \nu|)}$, or if $\lambda \equiv 2\pi/k > \lambda_c = \pi\sqrt{(2|\kappa - \nu|/N)}$.

(ii) Waves shorter than λ_c are monotonically damped owing to a 'mismatch' of diffusivities.

(iii) If $\nu = \kappa$ (i.e. if $Pr = 1$), then $N_c = 0$ and all waves exhibit oscillatory decay.

(iv) For the limiting case of a non-conducting inviscid fluid ($\nu \to 0$, $\kappa \to 0$), we have $v = v_0 \cos Nt \cos kx$, and $T = -v_0\sqrt{(\beta/\gamma g)} \sin Nt \cos kx$, the classic Brunt–Vaisala oscillation.

The 'problem', when forward Euler is used to integrate the associated semi-discretized version of (51) in time, is that an *additional* instability can arise in the case where ν and κ are 'small'; in fact, in the limit (the Brunt–Vaisala oscillation), this method is unconditionally unstable.

Rather than belabour this forward Euler instability (the analysis of which is lengthy when ν and κ are non-zero), we present the 'fix'; consider the following 'sequential' application of the forward Euler to (51) for the worst case ($\kappa = \nu = 0$):

$$v_{n+1} = v_n + \Delta t \gamma g T_n \tag{53a}$$

$$T_{n+1} = T_n - \Delta t \beta v_{n+1} = (1 - N^2 \Delta t^2) T_n - \beta \Delta t v_n \tag{53b}$$

The amplication matrix of this scheme is

$$\mathbf{A} = \begin{bmatrix} 1 & \gamma g \Delta t \\ -\beta \Delta t & 1 - N^2 \Delta t^2 \end{bmatrix} \tag{54}$$

i.e. $(v_{n+1}\ T_{n+1})^{\mathrm{T}} = \mathbf{A}(v_n\ T_n)^{\mathrm{T}}$ and stability requires that the modulus of each of the two eigenvalues of $\mathbf{A} \leq 1$. These eigenvalues are $\lambda = 1 - N^2 \Delta t^2/2 \pm i(N\Delta t/2)\sqrt{(4 - N^2 \Delta t^2)}$ and both have $|\lambda| = 1$ if $N\Delta t \leq 2$ and $|\lambda| > 1$ if $N\Delta t > 2$. Thus, the sequential method defined by (53) is stable *and* neutral (no damping—as in the continuum) if $\Delta t \leq 2/N$.

Remarks

(i) The order of the sequence in (53) is immaterial (i.e. the temperature equation could be advanced first), at least for linear problems.
(ii) The scheme (effectively) regains (conditional) stability by the inclusion of a higher-order term in Δt.
(iii) It has been recently advocated by Sun,[36] who calls it a forward-backward scheme.
(iv) For most practical cases involving internal gravity waves, the advection–diffusion (or CFL) stability limits are much more restrictive than $2/N$. We have not yet determined the combined stability limit.

The technique has been found to work well, at a modest increase in cost (per time step—but the times steps can be larger). (We update the temperature equation first, then use the updated value in the momentum equations.)

4.5. *Subcycling*

For many cases of interest, the stability limits associated with our explict time integration method are still quite restrictive, even when the BTD correction is applied (e.g. there is nothing really sacrosanct that requires the Courant number to be <1 *everywhere* in the flow at *all* times—cf. implicit methods). The cost of using small time steps is especially burdensome for the NS equations when a pressure update is (in principle) required at each time step.

To ease this problem we devised a cost-effective short cut called subcycling, which is based on the following premises: (i) stability often dictates a Δt which is much smaller than would be necessary to resolve with sufficient accuracy the solution to the ODEs and (ii) the discretized pressure gradient and associated continuity equation do not affect the stability of the time integration scheme. Both premises have been justified *a posteriori* from the results of many experiments and the latter has recently been upgraded to a fact, following (and generalizing from the L_2 norm) the recent relevant analysis by Bulgarelli *et al.*[37]

4.5.1. *A summary.* A four-step summary of the subcycling strategy is as follows:

(i) The minor (smaller, and fixed) time step is based on the stability estimates discussed in Section 4.3.1 and is used to (accurately) compute advection and diffusion *only* (these processes are 'subcycled'); the pressure gradient is approximated via simple (linear) extrapolation and the continuity equation (or, equivalently, the Poisson equation) is completely ignored during subcycling.

(ii) A constrained least-squares method (or, equivalently, a projection to a 'solenoidal' velocity field) is used at the conclusion of the subcycle process in order to re-enforce the satisfaction of the continuity equation.

(iii) The pressure field which is consistent with the now 'solenoidal' velocity field is updated in the usual manner, and finally,

(iv) The next major (larger) time step is dynamically computed based on the desired temporal accuracy by using a local error estimate (à la implicit methods; see Reference 1).

The overall scheme is represented graphically in Figure 13 which shows the velocity and pressure at three times and depicts the subcycle process between t_n and t_{n+1}. Here $\tilde{u}$ and $\tilde{P}$ are the approximate velocity and pressure during subcycling and u, P are their mass-consistent analogues (at major time steps only). The details of the procedure are described below.

4.5.2. *The subcycling process.* Between t_n and t_{n+1}, the solution of the following ODEs (cf. (3a)):

$$M\dot{\tilde{u}} + K(\tilde{u})\tilde{u} + C\tilde{P} = f; \qquad \tilde{u}(t_n) = u_n, \tag{55}$$

is approximated using the forward Euler method,

$$\tilde{u}_{m+1} = \tilde{u}_m + \Delta t_s \dot{\tilde{u}}_m = \tilde{u}_m + \Delta t_s M^{-1}[f_m - K(\tilde{u}_m)\tilde{u}_m - C\tilde{P}_m]; \qquad \tilde{u}_0 = u_n \tag{56}$$

where Δt_s is the subcycle time step and m is the subcycle index, $\tilde{P}(t)$ is known via linear extrapolation, $K(\tilde{u})$ includes the BTD correction in the viscous portion, and u_n, satisfying $C^T u_n = 0$, is available. This simple marching scheme is employed for a previously determined number, $S \simeq \Delta t_{n+1}/\Delta t_s$, of (stable) subcycle steps (the computation of S will be discussed later), at which time we have $\tilde{u}_s$ as an *approximation* to the desired velocity, u_{n+1}; in particular, it does not satisfy $C^T\tilde{u}_s = 0$.

Remark

Viewed as a continuum problem, it is clear that $\tilde{\mathbf{u}}$ is subject to the same boundary conditions as $\mathbf{u}$.

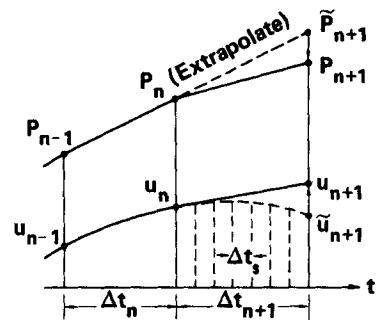

Figure 13. Schematic diagram of the subcycling process

4.5.3. *The return to mass consistency.* There are two equivalent ways in which to view the velocity adjustment process described below:

(1) Given a discrete approximation, $\tilde{u}$, to a velocity field which satisfies the desired BCs and nearly satisfies the discrete equations of mass and momentum conservation, seek a nearby field, v, which is 'as close as possible' to $\tilde{u}$ in some sense, and which satisfies $C^{\mathrm{T}}v = 0$. As we shall show, the appropriate way to perform this velocity adjustment is to minimize $(v-\tilde{u})^{\mathrm{T}}M(v-\tilde{u})$ subject to $C^{\mathrm{T}}v = 0$ or, equivalently, find the extremum of the functional

$$F(v, \lambda) = \tfrac{1}{2}(v-\tilde{u})^{\mathrm{T}}M(v-\tilde{u}) + \lambda^{\mathrm{T}}C^{\mathrm{T}}v \tag{57}$$

over all admissible vectors, v and λ. Here λ is a vector of Lagrange multipliers (one for each discretized continuity equation—like the pressure) and the distance measure between v and $\tilde{u}$ is seen to be related to kinetic energy (KE), since $\mathrm{KE}(u) = \frac{1}{2}u^{\mathrm{T}}Mu$. Setting the first variations of $F(v, \lambda)$ to zero yields the following Euler–Lagrange equations,

$$M(v-\tilde{u}) + C\lambda = 0 \tag{58a}$$

$$C^{\mathrm{T}}v = 0 \tag{58b}$$

which are more conveniently solved sequentially as

$$(C^{\mathrm{T}}M^{-1}C)\lambda \equiv A\lambda = C^{\mathrm{T}}\tilde{u} \tag{59a}$$

and

$$v = \tilde{u} - M^{-1}C\lambda \tag{59b}$$

since A is already available in factored form. Application of (59) to $\tilde{u}_{\mathrm{s}}$ gives the minimally-adjusted mass-consistent velocity field at t_{n+1}, i.e. $\mathrm{KE}(v-\tilde{u}_{\mathrm{s}})$ is minimal.

At the conclusion of subcycling, we apply (59) to $\tilde{u}_{\mathrm{s}}$ and use v as the best approximation to u_{n+1}, the velocity obtained without subcycling.

Remarks

(i) A is positive-definite since any possible pressure modes have already been properly dealt with (as discussed earlier); thus λ and v are unique. (Even if pressure modes are present, v is unique—when it exists. The existence of v (and λ) in the presence of pressure modes is a subtle point since it would appear that (58) always has a solution; suffice it to say here that serious difficulties will arise if the consistency requirements for well-posedness[6] are violated.)

(ii) Gaussian elimination is of course not the only solution method to be considered when subcycling is employed. It could also be cost-effective if the Poisson equations (for P and λ) are solved by iterative methods. In any case, the degree of cost-effectiveness is 'proportional to' the fractional cost of the pressure update to advance one time step.

(iii) The velocity adjustment is accomplished using the gradient of a scalar (i.e. $M^{-1}C \sim \nabla$).

(2) Given the same discrete approximation, $\tilde{u}_{\mathrm{s}}$, perform the unique orthogonal projection of $\tilde{u}_{\mathrm{s}}$ onto the subspace of vectors which is divergence-free (in the discrete sense). Since this projection must be performed using the gradient of a scalar (see Appendix II), we are led to (59) again.

Additional remarks

(i) The velocity adjustment is one of 'potential flow' and thus

(ii) The vorticity associated with the subcycled velocity field is correct (i.e. it is (essen-

tially) as good an approximation to that without subcycling as v is to u_{n+1}, which is usually quite good).

(iii) The continuous (in time) projection of $\tilde{u}$ onto the divergence-free subspace is, in fact, the *exact* soltuion of the discretized NS equations, $u(t) = B^{\mathrm{T}}\tilde{u}(t)$—see Appendix II; the subcycling process approximates this ideal.

(iv) The above velocity adjustment scheme is useful in other situations besides subcycling, namely whenever an approximation to a discretely divergence-free velocity field needs to be modified to the closest field which is 'properly' divergence-free, e.g. (1) for use as a valid initial condition for the NS equations, (2) for computing the stream function by contour integration around element boundaries—cf. Reference 38, (3) for use in objective analysis in meteorology.[39]

4.5.4. *The pressure update.* Once a mass consistent velocity field is available, the compatible pressure field, P_{n+1}, is obtained by solving (17), with n replaced by $n+1$, i.e. the factored A matrix is 'hauled out' one more time; a fact which shows (properly) that subcycling is only cost-effective when S is significantly larger than 2.

4.5.5. *The next major step size.* The last process in the subcycling strategy is the appropriate selection of the next major step size, Δt_{n+2} or, what is equivalent, the new subcycle ratio, $S = \Delta t_{n+2}/\Delta t_s$. In the actual algorithm, the step-size calculation is the first process in the sequence rather than the last; the order of presentation was chosen to provide (we hope) greater clarity.

Given u, $\dot{u}$ and P at t_n and t_{n+1}, the local (single step) time truncation error,

$$d_{n+1} \equiv u_{n+1} - u(t_{n+1}) \tag{60}$$

where $u(t_{n+1})$ is the (unknown) exact solution, needs to be estimated. Assuming that u_n is exact, a Taylor series expansion yields

$$u(t_{n+1}) = u_n + \Delta t_{n+1}\dot{u}_n + \Delta t_{n+1}^2 \ddot{u}_n/2 + O(\Delta t_{n+1}^3) \tag{61}$$

If we ignore the subcycling process, we also have

$$u_{n+1} = u_n + \Delta t_{n+1}\dot{u}_n \tag{62}$$

via the explicit Euler method. Thus

$$d_{n+1} = -\Delta t_{n+1}^2 \ddot{u}_n/2 + O(\Delta t_{n+1}^3) \tag{63}$$

Finally, since $\ddot{u}_n = (\dot{u}_{n+1} - \dot{u}_n)/\Delta t_{n+1} + O(\Delta t_{n+1})$, we get, neglecting higher-order terms,

$$d_{n+1} \simeq -\Delta t_{n+1}(\dot{u}_{n+1} - \dot{u}_n)/2 \tag{64}$$

as the local error estimate, where the acceleration vectors are obtained directly from (3a) at t_n and t_{n+1}.

In order to determine the next step size, Δt_{n+2}, we use (63) again:

$$\begin{aligned} d_{n+2} &= -\Delta t_{n+2}^2 \ddot{u}_{n+1}/2 + O(\Delta t_{n+2}^3) \\ &= \Delta t_{n+2}^2 \ddot{u}_n/2 + O(\Delta t_{n+1}\,\Delta t_{n+2}^2 + \Delta t_{n+2}^3) \\ &\simeq -(\Delta t_{n+2}/\Delta t_{n+1})^2 d_{n+1} \end{aligned} \tag{65}$$

where higher-order tems in Δt have been neglected. To get Δt_{n+2}, we set the norm of the local error to be committed during the next step to a user-specified tolerance, ε:

$$\|d_{n+2}\| \simeq (\Delta t_{n+2}/\Delta t_{n+1})^2 \|d_{n+1}\| = \varepsilon \tag{66}$$

which yields

$$\Delta t_{n+2} = \Delta t_{n+1}[\varepsilon/\|d_{n+1}\|]^{1/2} \tag{67}$$

as the next step size. In order that ε represent a relative error, we employ the following combined, weighted RMS norm; written for a 2D problem for simplicity:

$$\|d_{n+1}\| = 0{\cdot}5\Delta t_{n+1}\left\{\frac{1}{2N}\sum_{i=1}^{N}[(\dot{u}_{n+1})_i - (\dot{u}_n)_i]^2/u_{\max}^2 + [(\dot{v}_{n+1})_i - (\dot{v}_n)_i]^2/v_{\max}^2\right\}^{1/2} \tag{68}$$

where u, v are now the two components of the velocity, N is the total number of nodes (vector length), and $u_{\max} \equiv \max_i |(u_{n+1})_i|$ and similarly for $v_{\max}$.

Remarks

(i) $\varepsilon = 10^{-3}$ is usually a good choice, although a smaller vaue (e.g. 10^{-4}) is sometimes required.

(ii) If the effects of subcycling are included in the error estimate, we obtain $d_{n+1} \simeq -\Delta t_s \Delta t_{n+1} \ddot{u}_n/2$ rather than (63); efforts to control the step size using this result have not, thus far, proved to be as successful. (See, however, Reference 40.)

(iii) We have taken the ostensibly non-rigorous expedient of assuming that Δt_s is always small enough to give good accuracy. Thus, (1) we do not consider the temperature in the local error estimate and (2) if the algorithm tells us to use a step size smaller than Δt_s, we simply use Δt_s and do not subcycle (in fact, we also omit subcycling if $\Delta t_{n+2} \leq 2{\cdot}5\,\Delta t_s$) based on the belief that the inherent spatial error does not warrant any additional accuracy in time. (See also Reference 7.)

An overall algorithmic summary of the subcycling process may be useful; thus, given u_n and S,

(1) Use (56) for S subcycle steps to give $\tilde{u}_s$. (Also, update T via (16), with n replaced by m and Δt replaced by Δt_s, during these same S steps.)
(2) Solve (59a) for λ and compute v from (59b); set $u_{n+1} = v$.
(3) Compute the norm of the local error from (68).
(4) Compute the next major step size from (67).
(5) Update S $(= \Delta t_{n+2}/\Delta t_s)$ and go to (1). If $S < S_c$ (currently 3 in our code), omit subcycling.

This constitutes the the description of our numerical scheme. In Part 2 we will demonstrate the scheme, offer further comments regarding its viability (especially in 3D), and draw some conclusions.

ACKNOWLEDGEMENTS

This work was performed under the asuspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

APPENDIX I. REQUIREMENTS FOR WELL-POSEDNESS

In this appendix we review the restrictions on the diffusivity tensor in (18) in order that the problem be well-posed. The symmetric diffusivity tensor is

$$\mathbf{K} = \begin{bmatrix} K_1 & K_{12} \\ K_{12} & K_2 \end{bmatrix}$$

in 2D and

$$\mathbf{K}=\begin{bmatrix} K_1 & K_{12} & K_{13} \\ K_{12} & K_2 & K_{23} \\ K_{13} & K_{23} & K_3 \end{bmatrix}$$

in 3D. (Note that we are no longer dealing with dimensionless quantities—for convenience.) In order for the associated AD equation to be well-posed (i.e. to be a parabolic partial differential equation), the matrix of coefficients defining $\mathbf{K}$ must be positive definite, which in turn requires, in 2D

(i) $K_1>0$
(ii) $K_2>0$
(iii) $\det \mathbf{K}=K_1K_2-K_{12}^2>0$

In 3-D, the requirements are, in addition to those above,

(iv) $K_3>0$
(v) $K_1K_3-K_{13}^2>0$
(vi) $K_2K_3-K_{23}^2>0$
(vii) $\det \mathbf{K}>0.$

These conditions are necessary and sufficient for $\mathbf{K}$ to be positive definite[41] which we henceforth assume. If they are violated, (18) becomes, in part, the 'negative heat equation' and admits exponential growth in time.

APPENDIX II. VELOCITY PROJECTION

In this appendix, we follow and attempt to generalize somewhat (at least in the discrete case) the notions put forth by Chorin and Marsden[42] regarding incompressible flow. We begin by defining an inner product associated with the $(n\times n)$ positive definite symmetric mass matrix, M: $(a, b)\equiv b^{\mathrm{T}}Ma$, which also induces the (kinetic energy) norm $\|x\|=(x^{\mathrm{T}}Mx)^{1/2}$. We now state the orthogonal decomposition

Theorem

Any n-vector, $\tilde{u}$, can be uniquely decomposed in the form

$$\tilde{u}=v+M^{-1}CP \tag{69}$$

where $C^{\mathrm{T}}v=0$ and $(v, M^{-1}CP)=0$.

Remarks

(i) C is the same $(n\times m,\ n>m)$ matrix defined in (6d) (or (6d)′) and P is an m-vector.
(ii) We will call a vector divergence-free (i.e. weakly solenoidal) if it is annihilated by C^{T}.
(iii) $M^{-1}CP$ approximates ∇P in the continuum.
(iv) M can be either the consistent or lumped mass matrix.
(v) v is the unique orthogonal projection of $\tilde{u}$ onto the divergence-free subspace.

Proof. First we establish the orthogonality relationship: the subspace of divergence-free vectors is orthogonal to all vectors which are the gradients of scalars: if $C^{\mathrm{T}}v=0$, then $(v, M^{-1}CP)=0$ for *all* m-vectors P; i.e.

$$(v, M^{-1}CP)=(M^{-1}CP)^{\mathrm{T}}Mv=P^{\mathrm{T}}C^{\mathrm{T}}v=0$$

This orthogonality property can be used to demonstrate uniqueness of v and P. Suppose that two solutions exist, i.e. suppose

$$\tilde{u} = v_1 + M^{-1}CP_1 = v_2 + M^{-1}CP_2$$

or

$$v_1 - v_2 = M^{-1}C(P_2 - P_1)$$

Now form $\|v_1 - v_2\|^2$ to give

$$\|v_1 - v_2\|^2 = (v_1 - v_2)^{\mathrm{T}} M(v_1 - v_2) = [M^{-1}C(P_2 - P_1)]^{\mathrm{T}} M(v_1 - v_2) = (P_2 - P_1)^{\mathrm{T}} C^{\mathrm{T}}(v_1 - v_2) = 0$$

Thus $v_1 = v_2$ and $M^{-1}C(P_2 - P_1) = 0$. But since we have excluded pressure modes, $P_1 = P_2$ and the solution is unique.

To prove existence, and indeed to find the solution, we observe that (69) implies

$$(C^{\mathrm{T}}M^{-1}C)P = AP = C^{\mathrm{T}}\tilde{u} \tag{70}$$

Since A is positive definite and symmetric, A^{-1} exists and $P = A^{-1}C^{\mathrm{T}}\tilde{u}$ exists (and is unique). Finally it follows that v can be computed directly (and uniquely) from $v = \tilde{u} - M^{-1}CP$. QED

Additional remarks

(*i*) Given an arbitrary vector $\tilde{u}$ and any divergence-free vector u (e.g. the solution of the discretized NS equations), it can be shown that the unique divergence-free vector, v, computed as above, is closer to u than is $\tilde{u}$, i.e.

$$\|u - \tilde{u}\|^2 = \|v - \tilde{u}\|^2 + \|u - v\|^2 \quad \text{and thus} \quad \|u - v\| \leq \|u - \tilde{u}\|.$$

It also follows that $\|v - \tilde{u}\|^2 = P^{\mathrm{T}}AP = (CP)^{\mathrm{T}}M^{-1}(CP) = \|M^{-1}CP\|^2$

(ii) The projection operation can be (formally) represented as $v = B^{\mathrm{T}}\tilde{u}$, where $B \equiv I - CA^{-1}C^{\mathrm{T}}M^{-1}$. B^{T} is a projection operator since $(B^{\mathrm{T}})^2 = B^{\mathrm{T}}$ and $B^{\mathrm{T}}v = v$ where v is the projection of $\tilde{u}$ onto the divergence-free subspace.

(iii) Finally, $B^{\mathrm{T}}M^{-1}CP = M^{-1}BCP = 0$ since $B^{\mathrm{T}}M^{-1}$ is symmetric and $BC = 0$, i.e. $M^{-1}CP$ is orthogonal to the divergence-free subspace, a restatement of the orthogonality condition. (The subspace of vectors which are gradients of scalars is the orthogonal complement of the divergence-free subspace.)

REFERENCES

1. P. Gresho, R. Lee and R. Sani, 'On the time-dependent solution of the incompressible Navier–Stokes equations in two and three dimensions', in *Recent Advances in Numerical Methods in Fluids*, vol. 1, Pineridge Press, Swansea, U.K., 1980, p. 27.
2. M. Engelman, 'FIDAP: a fluid dynamics analysis package', *Adv. Eng. Software*, **4**, (4), 163 (1982).
3. D. Flanagan and T. Belytschko, 'A uniform strain hexahedron and quadrilateral with orthogonal hourglass control', *Int. J. Num. Meth. Eng.*, **17**, 679–706 (1981).
4. L. Petzold, 'Differential/algebraic equations are not ODEs', *SIAM J. Sci. Stat. Comput.*, **3**, (3), 367 (1982).
5. P. Gresho, 'Comments on: The significance of chequerboarding in a Galerkin finite element solution of the Navier–Stokes equations', *Int. J. Num. Meth. Eng.*, **18**, 1260–1262 (1982).
6. R. Sani, P. Gresho, R. Lee and D. Griffiths, 'The cause and cure (?) of the spurious pressures generated by certain FEM solutions of the incompressible Navier–Stokes equations', *Int. J. Num. Meth. Fluids*, **1**, 17–43, 171–204 (1981).
7. P. Gresho and R. Lee, 'Don't suppress the wiggles—they're telling you something!', *Comp. Fluids*, **9**, (2), 223 (1981).
8. M. J. P. Cullen, 'The finite element method', in *Numerical Methods Used in Atmospheric Models, Vol II*, GARP Publication Series No. 17, World Meteorological Organization, 1979.

9. J. Leone, P. Gresho, S. Chan and R. Lee, 'A note on the accuracy of Gauss–Legendre quadrature in the finite element method', *Int. J. Num. Meth. Eng.*, **14,** 769–784 (1979).
10. A. Brooks and T. Hughes, 'Streamline upwind/Petrov Galerkin formulation for convection-dominated flows with particular emphasis on the incompressible Navier–Stokes equations', *Comp. Meth. Appl. Mech. Eng.*, **32,** 199 (1982).
11. O. Zienkiewicz, *The Finite Element Method*, McGraw-Hill, London, 1977.
12. R. Lee, P. Gresho and R. Sani, 'Smoothing techniques for certain primitive variable solutions of the Navier–Stokes equations', *Int. J. Num. Meth. Eng.*, **14,** 1785–1804 (1979).
13. M. Cullen, 'Analysis of some low order finite element schemes for the Navier–Stokes equations', *J. Comp. Phys.*, **51,** 273 (1983).
14. P. Gresho, R. Lee and R. Sani, 'Further studies on equal-order interpolation for Navier–Stokes', *Fifth Int. Sym. on Finite Elements in Flow Problems, Proceedings*, Austin, Texas, 23–26 January 1984; also UCRL-89094.
15. D. Griffiths, personal communication, 1981.
16. L. Trefethen, 'Group velocity in finite difference schemes', *SIAM Review*, **24,** (2), 113 (1982).
17. G. Haltiner and R. Williams, *Numerical Weather Prediction and Dynamic Meteorology*, Wiley, New York, 1980, p. 477.
18. P. Smolarkiewicz, 'The multidimensional Crowley advection scheme', *Monthly Weather Review*, **110,** 1968 (1982).
19. D. Kosloff and G. Frazier, 'Treatment of hourglass patterns in low order finite element codes', *Int. J. Num. Anal. Meth. Geomech.*, **2,** 57–72 (1978).
20. J. Hallquist, 'Users manual for DYNA2D—an explicit two-dimensional hydrodynamic finite element code with interactive rezoning', *Lawrence Livermore National Laboratory Report UCID*-18756, 1980.
21. G. Goudreau and J. Hallquist, 'Recent developments in large-scale finite element Lagrangian hydrocode technology', *Comp. Meth. Appl. Mech. Eng.*, **33,** (1–3), 725 (1982).
22. R. Taylor, E. Wilson and S. Sackett, 'Direct solution of equations by frontal and variable band, active column methods, in *Nonlinear Finite Element Analysis in Structural Mechanics*, Springer-Verlag, 1981, 521.
23. A. Hindmarsh and P. Gresho, 'The stability of explicit Euler time integration for certain finite difference approximations of the multi-dimensional advection-diffusion equation', *Int. J. Num. Meth. Fluids*, to be published. Also available as *Lawrence Livermore National Laboratory Report UCRL*-88519.
24. J. Dukowicz and J. Ramshaw, 'Tensor viscosity method for convection in numerical fluid dynamics', *J. Comp. Phys.*, **32,** 71 (1979).
25. L. Lapidus and G. Pinder, *Numerical Solution of Partial Differential Equations*, Wiley, New York, 1982.
26. P. Roache, *Computational Fluid Dynamics*, Hermosa Publishers, P.O. Box 8172, Albuquerque, N.M., 1976.
27. P. Lax and B. Wendroff, 'Difference schemes for hyperbolic equations with high order of accuracy', *Comm. Pure & Appl. Math.*, **17,** 381 (1964).
28. S. Orzag, 'Numerical simulation of incompressible flows within simple boundaries: accuracy', *J. Fl. Mech.*, **49,** 75 (1971).
29. P. Gresho, R. Lee and R. Sani, 'Advection-dominated flows, with emphasis on the consequences of mass lumping', in *Finite Elements in Fluids—Vol.* 3, Wiley, Chichester, 1978, Chap. 19, p. 335.
30. P. Roache, 'On artificial viscosity', *J. Comp. Phys.* **10,** (2), 169 (1972).
31. G. deVahl Davis and G. Mallison, 'An evaluation of upwind and central difference approximations by a study of recirculating flow', *Comp. Fluids*, **4,** 29 (1976).
32. G. Raithby, 'Skew upstream differencing schemes for problems involving fluid flow', *Comp. Meth. Appl. Mech. Eng.*, **9,** 153 (1976).
33. J. Leone and P. Gresho, 'Finite element solutions of steady, two-dimensional, viscous incompressible flow over a step', *J. Comp. Phys.*, **41,** (1), 167 (1981).
34. J. Turner, *Buoyancy Effects in Fluids*, Cambridge University Press, Cambridge, 1973.
35. P. Gresho and C. Upson, 'Current progress in solving the time-dependent, incompressible Navier–Stokes equations in three dimensions by (almost) the FEM', *Proceedings, Fourth Int. Conf. on Finite Elements in Water Resources*, Hanover, Germany, 21–15 June 1982. Also, UCRL-87445.
36. W. Sun, 'A forward-backward time integration scheme to treat internal gravity waves', *Monthly Weather Review*, **108,** 402 (1980).
37. U. Bulgarelli, V. Casulli and D. Greenspan, 'Pressure methods for the approximate solution of the Navier–Stokes equations', *Num. Meth. in Laminar & Turbulent Flow, Proc., 3rd Int. Conf.*, Seattle, 8–11 August 1983.
38. R. Sani, P. Gresho, D. Tuerpe and R. Lee, 'The imposition of incompressibility constraints via variational adjustment of velocity fields', in *Proceedings, Num. Methods for Laminar and Turbulent Flow*, Pentech Press, London, 1978, p. 983. Also *Lawrence Livermore National Laboratory Report UCRL*-80553.
39. C. Sherman, 'A mass-consistent model for wind fields over complex terrain', *J. Appl. Met.*, **17,** 312 (1976).
40. P. Gresho and C. Upson, 'Application of a modified finite element method to the time-dependent thermal convection of liquid metal', *Proceedings, Int. Conf. Num. Meth. Laminar and Turbulent Flow*, University of Washington, 8–11 August 1983. Also Lawrence Livermore Laboratory Report UCRL-88990.
41. B. Noble, *Applied Linear Algebra*, Prentice-Hall, N.J., 1969.
42. A. Chorin and J. Marsden, *A Mathematical Introduction to Fluid Mechanics*, Springer-Verlag, N.Y., 1979.

CONTENTS OF PART 2

The remainder of this paper (Part 2), to appear in the next issue of the journal, contains:

1. Numerical results for:
   (1) Lid-driven cavity
   (2) Vortex shedding behing a cylinder
   (3) Simulation of a heavy gas release
2. Discussion of
   (1) Steady-state, stability, subcycling and normal modes
   (2) 2D vs. 3D solution strategy
3. Conclusions