INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2000; **32**: 439–464

Finite element modified method of characteristics for the Navier–Stokes equations

Alejandro Allievi[a,*] and Rodolfo Bermejo[b]

[a] *Institute for Marine Dynamics, National Research Council of Canada, PO Box 12093, Station A, St. John's, Newfoundland, Canada A1B 3T5*
[b] *Departamento de Matemática Aplicada, Facultad de Ciencias Matemáticas, Universidad Complutense de Madrid, Madrid 28040, Spain*

SUMMARY

An algorithm based on the finite element modified method of characteristics (FEMMC) is presented to solve convection–diffusion, Burgers and unsteady incompressible Navier–Stokes equations for laminar flow. Solutions for these progressively more involved problems are presented so as to give numerical evidence for the robustness, good error characteristics and accuracy of our method. To solve the Navier–Stokes equations, an approach that can be conceived as a fractional step method is used. The innovative first stage of our method is a backward search and interpolation at the foot of the characteristics, which we identify as the convective step. In this particular work, this step is followed by a conjugate gradient solution of the remaining Stokes problem. Numerical results are presented for:

(a) Convection–diffusion equation. Gaussian hill in a uniform rotating field.
(b) Burgers equations with viscosity.
(c) Navier–Stokes solution of lid-driven cavity flow at relatively high Reynolds numbers.
(d) Navier–Stokes solution of flow around a circular cylinder at $Re = 100$.

KEY WORDS: characteristics finite elements; incompressible viscous flow

1. INTRODUCTION

The idea of using the method of characteristics to integrate convective terms in transport diffusion equations has a long tradition in computational fluid dynamics. The conventional

* Correspondence to: Institute for Marine Dynamics, National Research Council of Canada, PO Box 12093, Station A, St John's, Newfoundland, Canada A1B 3T5. E-mail: allievi@hotmail.com

CCC 0271–2091/2000/040439–26$17.50

Received 18 June 1998
Revised 10 March 1999

manner of implementing this method is strictly based upon the Lagrangian description of the flow. Each mesh-point is identified with a fluid particle at time t_0 and followed at subsequent times $t_1, t_2, \ldots, t_n$ as the flow progresses. Thus, the solution of the transport–diffusion equation is obtained by solving the diffusion operator on the new meshes. The advantages of this pure Lagrangian approach over the conventional time stepping Eulerian schemes are numerical stability and absence of convective terms. The absence of convective terms is an interesting feature from a numerical point of view since, as is well known, these terms are a source of numerical difficulties. However, the deformation that the initial mesh undergoes as time progresses might lead to deterioration of the accuracy of the numerical solution. This drawback can be partially avoided if the mesh is regenerated after a few time steps. On the other hand, mesh regeneration followed by an interpolation of the solution from the old mesh to the new one can be very costly.

The modified method of characteristics (MMC) is an interpretation of the Lagrangian approach that overcomes the previously mentioned shortcoming—mesh deformation—while maintaining its good properties. The MMC does not follow fluid particles forward in time, instead it does so backwards. It seeks the position at time t_n of the particles that will reach the mesh-points at time t_{n+1}. Thus, the diffusion operator is always solved on the fixed initial mesh, avoiding the need for mesh regeneration.

The combination of finite elements and MMC for the solution of the Navier–Stokes equations was first introduced by Benqué *et al.* [1], Douglas and Russell [2] and Pironneau [3]. In all these works, the evaluation of the position of the fluid particles at the foot of the characteristic is carried out by an L^2-projection on the finite element space and the solution of the diffusion/Stokes step is advanced in time by a first-order Euler implicit scheme. As recognized in references [2,3] and other papers, the most difficult part of this approach is the evaluation of the integrals in the L^2-projection.

In this paper we have substituted the L^2-projection at the position of the fluid particles at t_n by the finite element interpolation using the basis functions of the velocity. This efficient, stable and more economical alternative to the L^2-projection is achieved by using the search–locate algorithm developed by Allievi and Bermejo [4]. The implementation of this approach in the context of transport–diffusion equations is clearly explained in pseudo-code format. The Stokes step is discretized in time using a Crank–Nicolson method for all terms involving velocity and external forcing. A first-order Euler implicit scheme is used for the pressure. In this work, the numerical solution of the Stokes system was carried out by the preconditioned conjugate gradient method proposed by Cahouet and Chabard [5] and by Glowinski and collaborators [6,7]. However, a number of other numerical procedures can be used to solve this step of our method.

The organization of the paper is as follows. Section 2 presents the model equations and appropriate boundary conditions. Section 3 introduces the semi-discrete version of the method of characteristics in relation to both the convective step and the weak solution of the Stokes step of our algorithm. Finite element mathematical generalities and fully discrete weak formulations are presented in Section 4. Finally, in Section 5 we apply our method to the solution of the test cases outlined in the summary.

2. THE NAVIER–STOKES EQUATIONS

Let $\Omega \subset \mathbf{R}^d$ ($d = 2$ or 3) be an open bounded domain with boundary Γ sufficiently smooth. We shall assume that $\Gamma = \Gamma_1 \cup \Gamma_2$, with the two subsets Γ_1 and Γ_2 satisfying $\Gamma_1 \cap \Gamma_2 = 0$. The Navier–Stokes equations describing unsteady flow of an incompressible Newtonian fluid are:

$$\frac{D\mathbf{u}}{Dt} - \nu\Delta\mathbf{u} + \nabla p = \mathbf{f} \text{ in } \Omega \times (0, T) \tag{1}$$

$$\nabla \cdot \mathbf{u} = 0 \text{ in } \Omega \times (0, T) \tag{2}$$

These equations are solved subject to the initial condition

$$\mathbf{u}(x, 0) = \mathbf{u}_0(x) \text{ in } \Omega \tag{3}$$

and the boundary conditions

$$\mathbf{u}(x, t) = \mathbf{g}_1(x, t) \text{ on } \Gamma_1 \tag{4}$$

$$-p\mathbf{n} + \nu\frac{\partial \mathbf{u}}{\partial \mathbf{n}} = \mathbf{g}_2(x, t) \text{ on } \Gamma_2 \tag{5}$$

where $\mathbf{n}$ is the outward unit normal to the boundary Γ. Hereafter, vector fields shall be denoted by boldface letters and generic points of $\bar{\Omega} = \Omega \cup \Gamma$ as $x = (x_1, \ldots, x_d)$. The notation used in Equations (1) and (2) is the following:

(i) $\mathbf{u} = (u_i)_{i=1}^d$ is the velocity, p is the pressure and ν is the kinematic viscosity coefficient.
(ii) $D\mathbf{u}/Dt = \partial\mathbf{u}/\partial t + \mathbf{u} \cdot \nabla\mathbf{u}$ denotes the material derivative of $\mathbf{u}$. It measures the rate of change of $\mathbf{u}$ as seen by an observer moving with the fluid particles.
(iii) $\mathbf{f} = (f_i)_{i=1}^d$ is a density of external forces.

Notice that Equation (5) is a condition of the traction vector $\mathbf{t} = -p\mathbf{n} + \nu(\partial\mathbf{u}/\partial\mathbf{n})$. As discussed by Gresho [8], it may result in a better characterization of the outflow boundary condition.

The existence and possible uniqueness of the solution to the system (Equations (1)–(5)) has been studied in a number of books and papers by, among others, Temam [9], Ladyzhenskaya [10], Heywood [11] and Lions [12]. Computing its numerical solution is not trivial due its non-linear structure, the incompressibility condition (Equation (2)) and the coupling of the equations through the terms $\mathbf{u} \cdot \nabla\mathbf{u}$ and $\nabla \cdot \mathbf{u}$.

In order to overcome some of these difficulties, we introduce in this paper a finite element modified method of characteristics (FEMMC) scheme. As we shall see below, this

scheme belongs to the category of methods of fractional steps. The first stage of our method is a Lagrangian step followed by an Eulerian procedure where a Stokes problem is solved.

3. THE SEMIDISCRETE CHARACTERISTIC ALGORITHM

Let the time interval $[0, T]$ be divided into N intervals $[t_n, t_{n+1}]$ of length Δt such that $N\Delta t = T$. For each interval $[t_n, t_{n+1}]$ we integrate Equations (1)–(5) along the trajectories of the fluid particles as described in the following developments.

We introduce the notation $X(x, t_{n+1}; t)$ to denote the position at time t of a fluid particle that reaches the point $x \in \bar{\Omega}$ at time t_{n+1}. The trajectory of such a particle satisfies the equation

$$\begin{cases} \dfrac{\mathrm{d}X}{\mathrm{d}t}(x, t_{n+1}; t) = \mathbf{u}(X(x, t_{n+1}; t), t) \\ X(x, t_{n+1}; t_{n+1}) = x \end{cases} \tag{6}$$

Note that $X(x, t_{n+1}; t)$ is the point at time t corresponding to the characteristic curve of the operator $(\partial/\partial t) + \mathbf{u} \cdot \nabla$. Assuming that $\mathbf{u}(\cdot, t)$ satisfies the conditions for the existence of a unique solution of Equation (6) for all time t, see [13], we have that for $t_n \le t < t_{n+1}$

$$X(x, t_{n+1}; t_n) = x - \int_{t_n}^{t_{n+1}} \mathbf{u}(X(x, t_{n+1}; t), t)\, \mathrm{d}t \tag{7}$$

It can be shown that Equation (7) defines a continuous transformation of $\bar{\Omega}$ onto itself provided that homogeneous Dirichlet boundary conditions for the velocity are used. For any $(x, t) \in \bar{\Omega} \times [t_n, t_{n+1}]$, we then integrate Equation (1) along the characteristics to obtain

$$\begin{aligned} \mathbf{u}(x, t_{n+1}) = \mathbf{u}(X(x, t_{n+1}; t_n), t_n) + \nu \int_{t_n}^{t_{n+1}} \Delta \mathbf{u}(X(x, t_{n+1}; t), t)\, \mathrm{d}t \\ - \int_{t_n}^{t_{n+1}} \nabla p(X(x, t_{n+1}; t), t)\, \mathrm{d}t + \int_{t_n}^{t_{n+1}} \mathbf{f}(X(x, t_{n+1}; t), t)\, \mathrm{d}t \end{aligned} \tag{8}$$

It is clear that the evaluation of the integrals in Equation (8) has to be approximated by a quadrature rule. In so doing, we shall obtain a time-marching algorithm to approximate $\mathbf{u}^{n+1}$ and p^{n+1}. A number of characteristic Galerkin schemes have been proposed by various authors to integrate Equation (8), see for instance [3,14]. Using the first-order upper limit quadrature rule to integrate Equation (8) yields the characteristic backward Euler scheme. Using quadratic interpolation at the feet of the characteristics and assuming sufficient regularity, Bermejo [15] showed that Crank–Nicolson is optimal for the velocity. Therefore, in this work we propose a scheme that is obtained by combining the trapezoidal rule for the

viscous term integral with the upper limit rule for the pressure term integral. Thus, a characteristic Crank–Nicolson scheme for the velocity and a first-order implicit scheme for pressure are obtained. Hence, we can formulate our time-marching scheme to integrate Equations (1)–(5) as follows.

Given $\mathbf{u}^0$, for any $x \in \bar{\Omega}$ and $n = 0, 1, \ldots, N$

(i) Evaluate

$$X(x, t_{n+1}; t_n) = x - \int_{t_n}^{t_{n+1}} \mathbf{u}(X(x, t_{n+1}; t), t)\, \mathrm{d}t. \tag{9}$$

(ii) Compute

$$\mathbf{u}^{*n} = \mathbf{u}^n(X(x, t_{n+1}; t_n)). \tag{10}$$

(iii) Solve

$$\mathbf{u}^{n+1} = \mathbf{u}^{*n} + \frac{\Delta t \nu}{2}[\Delta \mathbf{u}^{n+1} + \Delta \mathbf{u}^{*n}] - \Delta t \nabla p^{n+1} + \frac{\Delta t}{2}[\mathbf{f}^{n+1} + \mathbf{f}^n(X(x, t_{n+1}; t_n))], \tag{11}$$

$$\nabla \cdot \mathbf{u}^{n+1} = 0, \tag{12}$$

subject to

$$\mathbf{u}^0 = \mathbf{u}_0(x) \text{ in } \Omega, \tag{13}$$

$$\mathbf{u}^{n+1} = \mathbf{g}_1^{n+1} \text{ on } \Gamma_1, \tag{14}$$

$$-p^{n+1}\mathbf{n} + \nu \frac{\partial \mathbf{u}^{n+1}}{\partial n} = \mathbf{g}_2^{n+1} \text{ on } \Gamma_2. \tag{15}$$

We shall now consider in further detail steps (i) and (ii) of this scheme, and postpone until the next section the description of the numerical evaluation of step (iii).

3.1. Computation of $X(x, t_{n+1}; t_n)$

A crucial step in the application of our method is to accurately compute the points $X(x, t_{n+1}; t_n)$, which are the feet of the characteristic curves of the operator $\mathrm{d}u/\mathrm{d}t$. Bermejo [15] has shown that this is necessary in order to maintain an optimal order of accuracy. These analyses indicate that $X(x, t_{n+1}; t_n)$ must be computed with an error at least one order higher than that of the algorithm used to approximate the solution $\mathbf{u}(x, t_{n+1})$. Some authors estimate $X(x, t_{n+1}; t_n)$ by a second-order explicit Runge–Kutta scheme, which we have found not to be accurate enough to maintain a particle on its curved trajectory. A better choice is a fourth-order explicit Runge–Kutta scheme. However, it is inconvenient

that the velocity has to be extrapolated at both, $t_{n+1/2}$ and t_{n+1}. Pironneau [3] has proposed in the context of finite elements an explicit first-order Euler scheme to compute $X(x, t_{n+1};\ t_n)$ within each element. In this paper, we used a method first proposed by Temperton and Staniforth [16] in the framework of semi-Lagrangian schemes to integrate the weather prediction equations. In our own experience, this is a very accurate and efficient scheme when combined with the recently developed search–locate algorithm [4]. Turning our attention to Equation (7), we are particularly interested in computing $X(x, t_{n+1};\ t_n)$ corresponding to the mesh-points $\{x_i\}_{i=1}^{MV}$. Therefore, by virtue of the uniqueness of the solution of Equation (7), there is a unique point $X_i(x_i, t_{n+1};\ t_n)$ in $\bar{\Omega}$, associated with the grid-point x_i. In general, $X_i(x_i, t_{n+1};\ t_n)$ will not coincide with the spatial position of a grid-point. A requirement is then that the scheme to compute $X_i(x_i, t_{n+1};\ t_n)$ be provided with a search–locate algorithm to find the host element where such a point is located. A general, efficient and easy way to implement a scheme to perform this step in arbitrary grids is presented in [4]. We note that this method does not assume the characteristics to be straight lines. We approximate the characteristics by the mid-point rule together with fixed point iteration for greater accuracy and stability. Error estimates are given below. Then, to compute $X_i(x_i, t_{n+1};\ t_n)$, define

$$\alpha_i = x_i - X_i(x_i, t_{n+1}; t_n). \tag{16}$$

If we approximate the integral in Equation (7) by the mid-point rule, we have:

$$\alpha_i = \Delta t\,\mathbf{u}(X_i(x_i, t_{n+1}; t_{n+1/2}), t_{n+1/2}) + O(\Delta t^3). \tag{17}$$

Note that $X_i(x_i, t_{n+1};\ t_{n+1/2})$ is the mid-point of the arc that joins the points (x_i, t_{n+1}) and $(X_i(x_i, t_{n+1};\ t_n),\ t_n)$. Then we write

$$X_i(x_i, t_{n+1}; t_{n+1/2}) = x_i - \frac{1}{2}\alpha_i + O(\Delta t^2) \tag{18}$$

and

$$\mathbf{u}(X_i(x_i, t_{n+1}; t_{n+1/2}), t_{n+1/2}) = \mathbf{u}\left(x_i - \frac{1}{2}\alpha_i, t_{n+1/2}\right) + O(\Delta t^2). \tag{19}$$

Hence, substitution of Equation (18) into Equation (17) gives

$$\alpha_i = \Delta t\,\mathbf{u}\left(x_i - \frac{1}{2}\alpha_i, t_{n+1/2}\right) + O(\Delta t^3). \tag{20}$$

Note that the velocity $\mathbf{u}$ in Equation (19) is evaluated at time $t_{n+1/2}$. We carry out such an evaluation using the Adams–Bashforth formula

$$\mathbf{u}(x, t_{n+1/2}) = \frac{3}{2}\mathbf{u}(x, t_n) - \frac{1}{2}\mathbf{u}(x, t_{n-1}) + O(\Delta t^2), \tag{21}$$

recommended by Temperton and Staniforth [16] in relation to their semi-Lagrangian numerical weather prediction models. Based on an algorithm introduced by Robert [17], they also proposed the following iterative scheme to compute α_i.

Set

$$\alpha_i^0 = \Delta t\left[\frac{3}{2}\mathbf{u}(x_i, t_n) - \frac{1}{2}\mathbf{u}(x_i, t_{n-1})\right], \tag{22}$$

Then $\alpha_i^{(0)}$, compute for $k = 0, 1, \ldots$

$$\alpha_i^{(k+1)} = \Delta t\left[\frac{3}{2}\mathbf{u}\left(x_i - \frac{1}{2}\alpha_i^{(k)}, t_n\right) - \frac{1}{2}\mathbf{u}\left(x_i - \frac{1}{2}\alpha_i^{(k)}, t_{n-1}\right)\right]. \tag{23}$$

Now, if we set

$$e_i^{(k)} = \alpha_i - \alpha_i^{(k)},$$

$$e^{(k)} = \max|e_i^{(k)}|,$$

it follows from Equations (21) and (23) and a Taylor expansion that

$$e^{(k+1)} \leq \frac{1}{4}K\Delta t e^{(k)}. \tag{24}$$

when $K = \max|\nabla\mathbf{u}|$. Hence, if $K\Delta t$ is sufficiently small, we can conclude from Equation (24) that a few iterations (three or four) are enough to approximate α_i up to $O(\Delta t^3)$.

3.2. The Stokes problem

The computation of the total derivative by the MMC leads to the solution of the following Stokes-like problem:

$$\mathbf{u}^{n+1} - \beta\Delta\mathbf{u}^{n+1} + \Delta t\nabla p^{n+1} = \mathbf{F}^{n+1} \text{ in } \Omega, \tag{25}$$

$$\nabla\cdot\mathbf{u}^{n+1} = 0 \text{ in } \Omega, \tag{26}$$

$$\mathbf{u}^0 = \mathbf{u}_0(x) \text{ in } \Omega, \tag{27}$$

$$\mathbf{u}^{n+1} = \mathbf{g}_1^{n+1} \text{ on } \Gamma_1, \tag{28}$$

$$-p^{n+1}\mathbf{n} + \nu \frac{\partial \mathbf{u}^{n+1}}{\partial \mathbf{n}} = \mathbf{g}_2^{n+1} \text{ on } \Gamma_2 \tag{29}$$

where

$$\beta = \frac{1}{2}\Delta t \nu,$$

$$\mathbf{F}^{n+1} = \mathbf{u}^{*n} + \beta \Delta \mathbf{u}^{*n} + \frac{\Delta t}{2}[\mathbf{f}^{n+1} + \mathbf{f}^n(X(x, t_{n+1}; t_n))].$$

In order to facilitate the presentation of the numerical procedure of the next section, we shall next formulate the weak solution of Equations (25)–(29). We first define the spaces where the solution is found and introduce some mathematical notation. Let the subsets

$$\mathbf{V}_g \subset \mathbf{H}_1(\Omega)^d,$$

$$\mathbf{V}_0 \subset \mathbf{H}_1(\Omega)^d,$$

be defined as

$$\mathbf{V}_g = \{\mathbf{u} \in \mathbf{H}_1(\Omega)^d \,|\, \mathbf{u} = \mathbf{g}_1 \text{ on } \Gamma_1\},$$

$$\mathbf{V}_0 = \{\mathbf{u} \in \mathbf{H}_1(\Omega)^d \,|\, \mathbf{u} = 0 \text{ on } \Gamma_1\}. \tag{30}$$

$\mathbf{H}_1(\Omega)^d$ is the Hilbert space of vector functions with first derivatives in $\mathbf{L}^2(\Omega)$. Let the space for the pressure $Q = L^2(\Omega)/\mathbf{R}$ be defined as

$$Q = \{[p] \,|\, \forall p, q \in L^2(\Omega) p, q \in [p] \Rightarrow p - q = \text{constant}\} \tag{31}$$

Note that Q is a quotient space whose elements are cosets of $L^2(\Omega)$ functions. If no confusion arises, it is customary to denote each coset $[p]$ by its representative p. In particular cases, $\Gamma_1 \cup \Gamma_2 = \Gamma_1$, i.e., $\Gamma_2 = \varnothing$, Q is defined as

$$Q = L_0^2(\Omega) = \left\{ p \in L^2(\Omega) \,|\, \int_\Omega p \, \mathrm{d}x = 0 \right\}. \tag{32}$$

We shall now introduce the notation $(f, g) = \int_\Omega fg \, \mathrm{d}\Omega$ and $\langle f, g \rangle_\Gamma = \int_\Gamma fg \, \mathrm{d}\Gamma$. Next, we define the following bilinear forms:

$$a\colon \mathbf{H}_1(\Omega)^d \times \mathbf{H}_1(\Omega)^d \to \mathbf{R}$$

$$b\colon Q \times \mathbf{H}_1(\Omega)^d \to \mathbf{R}$$

as

$$a(\mathbf{u}, \mathbf{v}) = (\mathbf{u}, \mathbf{v}) + \beta(\nabla\mathbf{u}, \nabla\mathbf{v}),\ \forall \mathbf{u}, \mathbf{v} \in \mathbf{H}_1(\Omega)^d,$$

$$b(q, \mathbf{v}) = (q, \nabla\cdot\mathbf{v}),\ \forall q \in Q \text{ and } \mathbf{v} \in \mathbf{H}_1(\Omega)^d. \tag{33}$$

The weak formulation of the problem is then as follows. Find $\mathbf{u}^{n+1} \in \mathbf{V}_g$ and $p^{n+1} \in Q$ such that for any $\mathbf{v} \in \mathbf{V}_0$ and $q \in Q$

$$a(\mathbf{u}^{n+1}, \mathbf{v}) - \Delta t b(p^{n+1}, \mathbf{v}) = (\mathbf{F}^{n+1}, \mathbf{v}) + \langle g_2^{n+1}, \mathbf{v}\rangle \Gamma_2$$

$$b(q, \mathbf{u}^{n+1}) = 0, \tag{34}$$

It can be proven (see [18]) that there exists a unique solution to Equation (34) that satisfies the inf–sup condition. We have used in this paper a conjugate gradient algorithm (CGA) devised by Glowinski and his collaborators [6,7] to obtain the numerical solution of Equation (34).

4. THE FINITE ELEMENT SOLUTION

We shall describe in this section the finite element formulation of the semi-discrete scheme. To approximate the solution $(\mathbf{u}, p)$, we use Taylor–Hood finite elements (P_2/P_1 or Q_2/Q_1, i.e., quadratic polynomials for the velocity and bilinear polynomials for the pressure on simplices or quadrilaterals respectively). For this class of elements, both velocity and pressure are continuous over the element's boundary. It is well known that for such elements the discrete velocity and pressure fields satisfy the inf–sup condition [19–21]. Let us now start by describing some generalities related to finite element theory.

For convenience, let Ω be a convex domain and D_h a regular partition of Ω in small elements $\{k_j\}_{j=1}^{NE}$. Let us assume these elements to be quasi-uniform in the sense of [22], i.e., they satisfy the minimum angle condition. If k_i and k_j are two different elements of D_h, then some of the following conditions are satisfied:

$$k_i \cap k_j = \begin{cases} \varnothing, \text{ void intersection, or} \\ \Gamma_{ij}, \text{ a boundary in common, or} \\ P_{ij}, \text{ a vertex.} \end{cases}$$

The conforming finite element spaces for velocity and pressure are:

$$\begin{aligned} \mathbf{V}_h &= \{\mathbf{u}_h \in C^0(\bar{\Omega})^d | \mathbf{u}_h|_{k_j} \in S(k_j)\ \forall j\}, \\ Q_h &= \{p_h \in C^0(\bar{\Omega})^d | p_h|_{k_j} \in R(k_j)\ \forall j\}. \end{aligned} \tag{35}$$

where $S(k_j)$ and $R(k_j)$ are spaces of polynomials defined on k_j. Specifically,

$S(k_j) = P_2(k_j)$ for simplices,
$S(k_j) = Q_2(k_j)$ for quadrilaterals,
$R(k_j) = P_1(k_j)$ for simplices,
$R(k_j) = Q_1(k_j)$ for quadrilaterals.

In addition, it is useful to define the following finite dimensional spaces:

$$\begin{aligned} \mathbf{V}_{h0} &= \{\mathbf{u}_h \in \mathbf{V}_h | \mathbf{u}_h|_{\Gamma_1} = 0\}, \\ M_h &= \{w_h \in Q_h | w_h|_{\Gamma_2} = 0\}. \end{aligned} \tag{36}$$

Remark 4.1. For s and r integers such that $1 \le s \le 2$ and $r = 1$, if $\mathbf{u}$ and p are sufficiently smooth, then the finite element solutions $\mathbf{u}_h$ and p_h satisfy the following error bounds:

$$\|\mathbf{u} - \mathbf{u}_h\|_0 + h\|\nabla(\mathbf{u} - \mathbf{u}_h)\|_0 \le K_1 h^{s+1} \|\mathbf{u}\|_{s+1},$$

$$\|p - p_h\|_0 \le K_2 h^r \|p\|_{H^r/\mathbf{R}},$$

where K_1 and K_2 are constants, and $\|o\|_0$, $\|o\|_{s+1}$, and $\|o\|_{H^r/\mathbf{R}}$ denote the norms in the Hilbert spaces $L^2(\Omega)$, $H^{s+1}(\Omega)$ and $H^r(\Omega)/\mathbf{R}$ respectively. These estimates mean that if $(\mathbf{u}, p)$ are sufficiently regular, then $(\mathbf{u}_h, p_h)$ will approximate $(\mathbf{u}, p)$ up to $(O(h^3), O(h^2))$ in space.

Remark 4.2. At each time t_n, $\mathbf{u}_h(x, t_n) \in \mathbf{V}_h$ satisfies the Lipschitz condition, i.e., $\forall x_1, x_2 \in \Omega$ there exists a constant L such that

$$\max|\mathbf{u}_h(x_1, t_n) - \mathbf{u}_h(x_2, t_n)| \le L|x_1 - x_2|$$

In addition, for any (x, t_n):

$$\max|\mathbf{u}_h(x, t)| \le \infty.$$

These two conditions are important in order to approximate the unique solution of the departure points.

We can formulate the finite element solutions to $\mathbf{u}^n(x)$ and $p^n(x)$ as

$$\begin{aligned} \mathbf{u}_h^n(x) &= \sum_{j=1}^{MV} \mathbf{U}_j^n \theta_j, \\ p_h^n(x) &= \sum_{k=1}^{MP} P_k^n \omega_k. \end{aligned} \tag{37}$$

In Equation (37), $\{\theta_j\}$ and $\{\omega_k\}$ are the basis function sets for $\mathbf{V}_h$ and Q_h respectively, and $\mathbf{U}_j^n$ and P_k^n are the corresponding nodal values of $\mathbf{U}_h^n(x)$ and $p_h^n(x)$. We can now introduce the algorithm to compute the fully approximate solution of the Navier–Stokes equations. Following the formulation described for the semi-discrete scheme, we distinguish two steps. In the first one, termed the convective step, we compute the departure points $X(x, t_{n+1}; t_n)$ and the velocity $\mathbf{u}^{*n}(x) = \mathbf{u}^n(X(x, t_{n+1}; t_n))$. The second step corresponds to the solution of the Stokes problem. If the algorithm is envisioned in this manner, it can be conceived as a fractional-step method.

4.1. Fully discrete algorithm for the convective step

We shall hereafter use the notation X_h^n to denote the approximate value of the departure points $X_h(x, t_{n+1}; t_n)$. In referring to the departure point corresponding to the i-th mesh-point $x = x_i$, the notation will be X_{hi}^n. It is clear that the departure points of interest are those associated with the mesh-points $\{x_i\}_{i=1}^{MV}$. Then, by virtue of Equation (16) we set

$$X_{hi}^n = x_i - \alpha_{hi}, \tag{38}$$

where α_{hi} is calculated by the iterative procedure

$$\alpha_{hi}^{(k+1)} = \frac{\Delta t}{2}\left[3\mathbf{u}_h^n\left(x_i - \frac{1}{2}\alpha_{hi}^{(k)}\right) - \mathbf{u}_h^{n-1}\left(x_i - \frac{1}{2}\alpha_{hi}^{(k)}\right)\right]. \tag{39}$$

The values $\mathbf{u}_h^n(x_i - \frac{1}{2}\alpha_{hi}^{(k)})$ and $\mathbf{u}_h^{n-1}(x_i - \frac{1}{2}\alpha_{hi}^{(k)})$ are obtained by finite element interpolation according to Equation (37). Note that an important ingredient to carry out the iterative procedure (Equation (39)) is the identification of the element where $x_i - \frac{1}{2}\alpha_{hi}^{(k)}$ is located. To compute the velocity $\mathbf{u}_h^{*n}(x) = \mathbf{u}_h^n(X_h^n)$, we set

$$\mathbf{u}_h^{*n}(x) = \sum_{i=1}^{MV} \mathbf{U}_i^{*n}\theta_i(x). \tag{40}$$

The values $\{\mathbf{U}_i^{*n}\}$ are evaluated by finite element interpolation of $\mathbf{u}_h^n(x)$ at the feet of the characteristics $\{X_{hi}^n\}$ determined in Equation (38). Huffenus and Khaletzky [14] and Temperton and Staniforth [16] also employ interpolation to calculate the velocity at the points $\{X_{hi}^n\}$. A theoretical analysis of the interpolation procedure to compute $\mathbf{u}_h^{*n}(x)$ was presented by Bermejo in [15]. Note that the conventional characteristic Galerkin schemes of [2,3] calculate $\mathbf{u}_h^{*n}(x)$ by an L^2-projection on the space of the velocity $\mathbf{V}_h$—a projection that may be computationally expensive to achieve.

The convective step is more easily visualized in a pseudocode format. As mentioned previously, a search–locate algorithm for the fluid particles at the feet of the characteristics has to be provided. In the developments below, the subroutine **slalg** identifies the host elements of

the fluid particles at the feet of the characteristics. It also provides the co-ordinates of these points referred to the reference element. The **slalg** algorithm, described in detail by Allievi and Bermejo in [4], uses a Newton method to invert a bijective map from the reference element into a given mesh element, together with a criterium to move from element to element in the mesh. Computation of the values $\{\mathbf{U}_i^{*n}\}$ is then carried out as follows,

input MV, n, $NODES$, ISE, $ITFCH$, $ITSLA$, $\{\mathbf{U}_i^n\}$, $\{\mathbf{U}_i^{n-1}\}$, $\{\mathbf{x}_i\}$, ε_1, ε_2, Δt

MV: number of velocity nodes.
n: number of velocity nodes per element.
$NODES$: two-dimensional array of node connectivity for each element.
ISE: vector array containing neighbor element number for each element.
$ITFCH$: maximum number of iterations for calculation of the feet of the characteristics (see Section 3.1).
$ITSLA$: maximum number of iterations for search–locate Newton method.
$\mathbf{U}_i^n$, $\mathbf{U}_i^{n-1}$: nodal values of velocities at previous two time steps.
$\mathbf{x}_i$: mesh points co-ordinates.
ε_1: tolerance for convergence of calculation of foot of characteristic.
ε_2: tolerance for convergence of search–locate Newton method.
Δt: time step
IHE: vector array containing host elements i.e. of points X_{hi}^i.
$\bar{\mathbf{x}}$: vector array containing co-ordinates of points X_{hi}^i.
$\mathbf{z}$: working array to store values α_i^k at each iteration k (see Equation (23)).

Part (A) of the pseudocode starts by computing the extrapolated values of the velocity $\mathbf{u}$ according to Equation (21), the values of $\alpha_i^{(0)}$ according to Equation (22) and the co-ordinates $\bar{x}_i = \mathbf{x}_i - \frac{1}{2}\alpha_i^{(0)}$.

$$
(\mathbf{A})\begin{cases}
\textbf{for } i = 1, \ldots, MV \textbf{ do} \\
\mathbf{V}_i \leftarrow \dfrac{1}{2}(3\mathbf{U}_i^n - \mathbf{U}_i^{n-1}) \text{ (Store } \mathbf{V} \text{ in } \mathbf{U}^{n-1}) \\
\mathbf{z}_i \leftarrow \Delta t \mathbf{V}_i \\
\bar{\mathbf{x}}_i \leftarrow \mathbf{x}_i - \dfrac{1}{2}\mathbf{z}_i \\
\textbf{end do}
\end{cases}
$$

Part (B) of the pseudocode performs the iterative procedure described by Equation (23). With input ($\bar{\mathbf{x}}$, $NODES$, ISE, IHE, ε_2, $ITSLA$), function **slalg** identifies the elements *ie* hosting the points $x_i - \frac{1}{2}\alpha_i$. On exit, **slalg** returns the array IHE containing all *ie* as well as the array $\bar{\mathbf{x}}$ overwritten with the reference element co-ordinates of the points $x_i - \frac{1}{2}\alpha_i$. With this information this section proceeds to apply Equation (23) for $k = 1, 2, \ldots, ITFCH$. The finite element interpolation is performed on the reference element by the expression $\mathbf{s} = \mathbf{s} + \Delta t \mathbf{V}_{iej}\lambda_j(\bar{\mathbf{x}})$, where λ_j are the basis functions for velocity. Following interpolation, the remainder of this section tests convergence.

(B)

for $k = 1, 2, \ldots\ ITFCH$ **do**

call slalg ($\bar{\mathbf{x}}$, $NODES$, ISE, IHE, ε_2, $ITSLA$)

for $i = 1, 2, \ldots, MV$ **do**

$ie = IHE(i)$

$\mathbf{s} = 0.0$

for $j = 1, 2, \ldots, n$ **do**

$iej = NODES\ (ie, j)$

$\mathbf{s} = \mathbf{s} + \Delta t \mathbf{V}_{iej} \lambda_j(\bar{\mathbf{x}}_j)$

end do

$\bar{\mathbf{x}}_i \leftarrow \mathbf{s}$

end do

if max $|\bar{\mathbf{x}}_i - \mathbf{z}_i| > \varepsilon_1$ **then**

for $i = 1, 2, \ldots, MV$ **do**

$\mathbf{z}_i \leftarrow \bar{\mathbf{x}}_i$

$\bar{\mathbf{x}}_i \leftarrow \mathbf{x}_i - \frac{1}{2}\mathbf{z}_i$

end do

ELSE

for $i = 1, 2, \ldots, MV$ **do**

$\mathbf{z}_i \leftarrow \bar{\mathbf{x}}_i$

$\bar{\mathbf{x}}_i \leftarrow \mathbf{x}_i - \mathbf{z}_i$

end do

output $\{\bar{\mathbf{x}}_i\}$

end if

end do

Part (C) of the pseudocode describes the procedure to evaluate $\mathbf{U}_i^{*n}$ (see Equation (40)) by finite element interpolation at the points X_{hi}^i. Note that the structure of this section is the same as in Part (B) before testing convergence.

$$
(\mathbf{C})\begin{cases}
\textbf{call slalg } (\bar{\mathbf{x}},\ NODES,\ ISE,\ IHE,\ \varepsilon_2,\ ITSLA)\\
\textbf{for } i = 1, 2, \ldots, MV \textbf{ do}\\
ie = IHE(i)\\
\mathbf{s} = 0.0\\
\textbf{for } j = 1, 2, \ldots, N \textbf{ do}\\
iej = NODES\ (ie, j)\\
\mathbf{s} = \mathbf{s} + \mathbf{U}_{iej}^{n-1}\lambda_j(\bar{\mathbf{x}}_j)\ \text{(Remember } \mathbf{V} \text{ is stored in } \mathbf{U}^{n-1})\\
\textbf{end do}\\
\mathbf{U}_i^{n-1} \leftarrow \mathbf{U}_i^n\ \text{(Switch arrays)}\\
\mathbf{U}_i^{*n} \leftarrow \mathbf{s}\ \text{(Store } \mathbf{U}_i^{*n} \text{in } \mathbf{U}_i^n)\\
\textbf{end do}\\
\textbf{output } \{\mathbf{U}_i^{*n}\}
\end{cases}
$$

4.2. Fully discrete algorithm for the Stokes step

The finite element formulation for the Stokes problem (Equations (25)–(29)) is as follows. Find $(\mathbf{u}_h^{n+1}, p_h^{n+1}) \in \mathbf{V}_{h0} \times Q_h$, such that for any $(\mathbf{v}_h, q_h) \in \mathbf{V}_h \times Q_h$

$$a(\mathbf{u}_h^{n+1}, \mathbf{v}_h) - \Delta t b(p_h^{n+1}, \mathbf{v}_h) = (\mathbf{F}_h^{n+1}, \mathbf{v}_h) + \langle g_{2h}^{n+1}, \mathbf{v}_h \rangle_{\Gamma_2}, \tag{41}$$

$$b(q_h, \mathbf{u}_h^{n+1}) = 0.$$

where

$$\mathbf{u}_h^0 = \mathbf{u}(x, 0),$$

$$\mathbf{u}_n^{n+1}\big|_{\Gamma_1} = \mathbf{g}_{1h}^{n+1},$$

$$-p_h^{n+1}\cdot\mathbf{n} + \nu(\nabla\mathbf{u}_h^{n+1}\cdot\mathbf{n})\big|_{\Gamma_2} = \mathbf{g}_{2h}^{n+1},$$

$$(\mathbf{F}_h^{n+1}, \mathbf{v}_h) = \left(\mathbf{u}_h^{*n} + \beta\,\Delta\mathbf{u}_h^{*n} + \frac{\Delta t}{2}[\mathbf{f}^{n+1} + \mathbf{f}_h^{*n}], \mathbf{v}_h\right),$$

$$\mathbf{f}_h^{*n}(x) = \sum_{i=1}^{MV} \mathbf{f}_i^{*n}\theta_i(x).$$

The values $\mathbf{f}_i^{*n}$ are the values of $\mathbf{f}^n(X_{hi}(x_i, t_{n+1}; t_n))$ calculated by finite element interpolation as we did with $\mathbf{U}_i^{*n}$.

In this work, we have applied the finite element version of the CGA in [6,7] to obtain the numerical solution of Equation (41). Note that we only solve a number of symmetric algebraic systems of linear equations.

We now conclude this section where we have described the essentials of our FEMMC together with its implementation in the context of the fully discrete Stokes problem. In our particular case, we have solved the Stokes system using the method described in [6,7]. However, other solvers will suffice to carry out this step. In the section that follows, we shall apply it to benchmark scenarios that are customarily used in testing numerical algorithms leading to the solution of the Navier–Stokes equations.

5. NUMERICAL RESULTS

In this section, we present a series of numerical tests with the intention of clearly demonstrating the performance of our FEMMC algorithm and of presenting a comparison with other numerical and experimental results. Before introducing results for the Navier–Stokes equations in Section 5.3, we shall apply our interpolation algorithm to the solution of the convection–diffusion and viscous Burgers equations. Section 5.1 and 5.2 describe these two problems.

5.1. Convection–diffusion equation. The Gaussian hill

The mathematical formulation of this problem is as follows:

$$\frac{\partial \mathbf{c}}{\partial t} + \mathbf{u}\frac{\partial \mathbf{c}}{\partial x} + \mathbf{v}\frac{\partial \mathbf{c}}{\partial y} = \nu\left(\frac{\partial^2 \mathbf{c}}{\partial x^2} + \frac{\partial^2 \mathbf{c}}{\partial y^2}\right) \text{ in } \Omega \times (0, T) \tag{42}$$

with the boundary and initial conditions

$$\frac{\partial \mathbf{c}}{\partial \mathbf{n}} = 0 \tag{43}$$

$$\mathbf{c}(x, 0) = 100 \exp\left(-\frac{(x - x_0)^2 + (y - y_0)^2}{2\sigma^2}\right) \tag{44}$$

where $\mathbf{c}$ is a passive scalar which is being convected with known velocity components ($\mathbf{u}$, $\mathbf{v}$) and ν is a coefficient of diffusivity.

In this particular example, we look at the diffusion of a Gaussian hill in a uniform rotating field. The parameters used have been taken from the work of Pudykiewicz and Staniforth [23]. These are: Ω is a 3200 km $\times$ 3200 km domain, the components of the velocity field are $\mathbf{u} = -\omega y$ and $\mathbf{v} = -\omega x$ where $\omega = 10^{-5}\ \mathrm{s}^{-1}$ is the angular velocity. In an infinite plane the analytical solution of this problem is given as:

$$\mathbf{c}(x, t) = \frac{100}{1 + (2\nu t/\sigma^2)} \exp\left(- \frac{\bar{\mathbf{x}}^2 + \bar{\mathbf{y}}^2}{2(\sigma^2 + 2\nu t)} \right) \tag{45}$$

where

$$\bar{\mathbf{x}} = x - x_0 \cos \omega t + y_0 \sin \omega t$$

$$\bar{\mathbf{y}} = y - x_0 \sin \omega t + y_0 \cos \omega t$$

$(x_0, y_0) = (-800 \text{ km}, 0 \text{ km})$ (center of initial condition)

$$\sigma^2 = 2 \cdot 10^{10} \text{ m}^2$$

Table I shows the variation with the grid size h_0 of the relative L^2-error of the numerical solution for various experiments we have conducted with CFL numbers of 2 and 10. We define such error as:

$$\frac{\left(\int_\Omega |\mathbf{u} - \mathbf{u}_h|^2 \, d\Omega \right)^{1/2}}{\left(\int_\Omega |\mathbf{u}|^2 \, d\Omega \right)^{1/2}}.$$

Columns 2–5 correspond to the results obtained by FEMMC with quadratic interpolation for the convective step and bilinear elements for the diffusive step. The last column shows results of an Eulerian scheme that uses bilinear finite elements in space and Crank–Nicolson in time. It is clear from this table that the FEMMC error is approximately between 3 and 6 times smaller than the error obtained with the Eulerian scheme. In addition, the FEMMC error decreases roughly by a factor of 4 as we move down from row to row of the table (with the exception of the last row for the column corresponding to CFL = 2 and $\nu = 4 \cdot 10^4$ m^2 s^{-1}). For our experiments $\Delta t = O(h_0)$ which means that the convergence of the FEMMC is quadratic both in time and space. Note that the quadratic convergence of the Eulerian scheme can only be achieved as the grid size becomes very small ($h_0/4$, $h_0/8$).

Table I. L^2-error after one revolution of Gaussian hill ($h_0 = 100$ km)

h	FEMMC				ECN
	$\nu = 10^4$ m^2 s^{-1}		$\nu = 4 \cdot 10^4$ m^2 s^{-1}		$\nu = 10^4$ m^2 s^{-1}
	CFL = 2	CFL = 10	CFL = 2	CFL = 10	CFL = 2
h_0	0.28983	0.29840	0.10499	0.21607	0.8903
$h_0/2$	0.08785	0.07955	0.02296	0.05407	0.4513
$h_0/4$	0.02452	0.02009	0.00687	0.01405	0.1414
$h_0/8$	0.00684	0.00501	0.00328	0.00450	0.0367

5.2. Burgers equation with viscosity

Our next test case is related to the capacity of the FEMMC algorithm to simulate the formation of strong discontinuities. To this end, we solve the problem:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\frac{\partial \mathbf{u}}{\partial x} + \mathbf{v}\frac{\partial \mathbf{u}}{\partial y} = \nu\left(\frac{\partial^2 \mathbf{u}}{\partial x^2} + \frac{\partial^2 \mathbf{u}}{\partial y^2}\right) \text{ in } \Omega \times (0, T) \tag{46}$$

with periodic boundary conditions and the initial condition

$$\mathbf{u}(x, y, 0) = \frac{1}{4} + \frac{1}{2}\sin \pi\left(\frac{x+y}{2}\right) \text{ in } \Omega \tag{47}$$

The solution domain Ω is the square $(-2, 2) \times (-2, 2)$ and the grid size used $h = 1/40$. The solution of Equation (46) exhibits the formation of discontinuities as ν decreases. As $\nu \to 0$, the solution tends to the solution of the inviscid equation which consists of a shock that develops at time $t = 2/\pi$. Figure 1 shows numerical results for this case using bilinear elements for the diffusion term. As in the inviscid case, we note that two discontinuities become more evident as ν decreases. A sectional cut from the lower-left corner to the upper-right corner of the domain shows the profile of the solution along the diagonal. It is clear that the algorithm captures the position of the shock quite well.

5.3. Navier–Stokes equations

In this sub-section, we present numerical results for flows developed in a lid-driven square cavity and around a circular cylinder positioned inside a straight channel with rigid boundaries. For these test cases, we have used unequal order Q2/Q1 finite elements, i.e., continuous quadratic and continuous bilinear Lagrange quadrilateral elements for velocity and pressure respectively. All algebraic systems were solved using the Conjugate Gradient Method with Incomplete Cholesky Decomposition (ICCG) and a convergence criteria of 10^{-15}. In general, the systems associated with the solution for the velocities required not more than ten ICCG iterations. For the Poisson problems, the number of ICCG iterations were approximately 60 or less for all cases presented in this work. Note that we have identified the conjugate gradient solver for the algebraic systems as ICCG and as CGA the solution algorithm outlined in Section 4.2. All computations were carried out using a time step $\Delta t = 0.1$.

5.3.1. Lid-driven cavity flow in two dimensions. For the computations in this case, the cavity dimensions in both directions and the lid velocity were set equal to unity. The Reynolds number *Re* was then defined as $1/\nu$. Calculations were carried out using the grids defined in Table II. All simulations were conducted for a time period of 150 s. The lid horizontal velocity was regularized in time using a linear function in the interval [0, 25]. Average number of iterations given below were calculated from $t_1 = 0$ and the time t_2 at which the CGA started to converge with only one iteration.

Table III presents the number of iterations required for the CGA to converge to a tolerance of 10^{-3} in the maximum norm, i.e., max $|\nabla \cdot \mathbf{u}| \leq 10^{-3}$. We have spanned a relatively wide spectrum of Reynolds numbers. In so doing, we have attempted to expose the computational

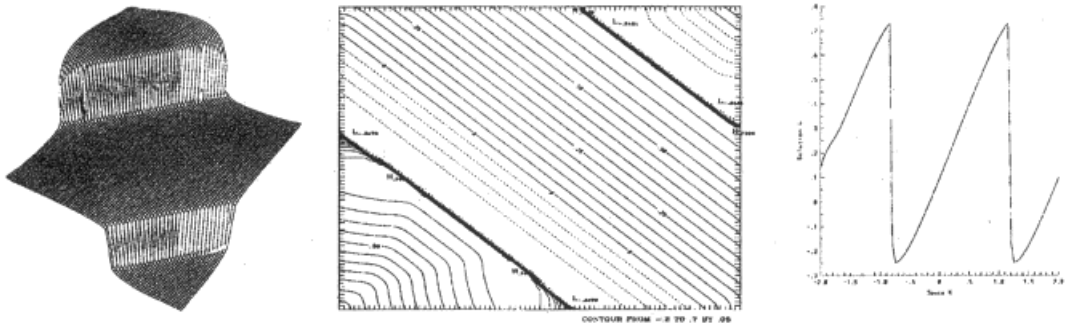

Fig 1.a: $t = 1.$, $\nu = 0.001$, $\Delta t = \frac{1}{16}$.

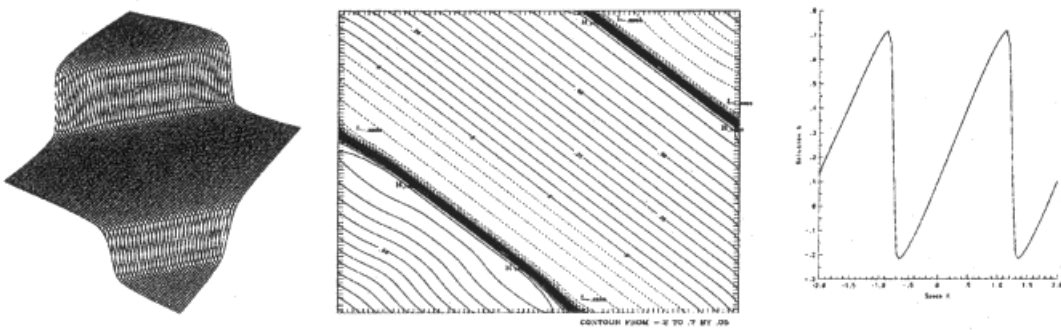

Fig 1.b: $t = 1.$, $\nu = 0.01$, $\Delta t = \frac{1}{16}$.

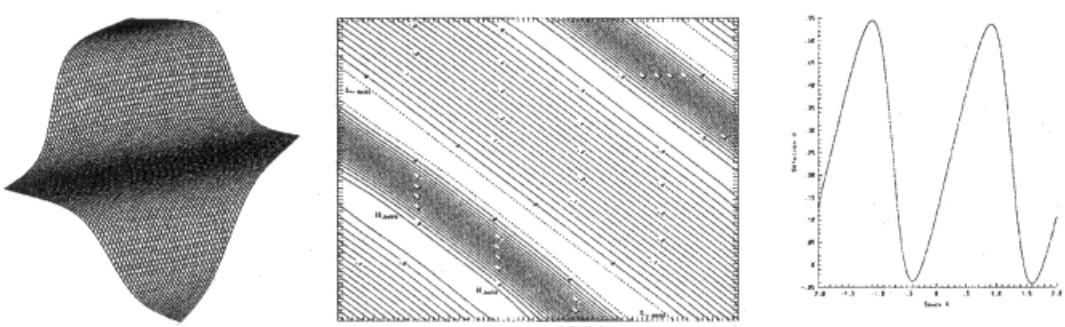

Fig 1.c: $t = 1.$, $\nu = 0.1$, $\Delta t = \frac{1}{16}$.

Figure 1

Table II. Computational conditions

Lid-driven cavity flow in two dimensions		
Mesh	Nodes p	Nodes (u, v)
30×30	961	3721
62×40	2583	10 125

Table III. Average number of iterations CGA[a]

Lid-driven cavity flow in two dimensions							
$Re = 100$	$Re = 1000$	$Re = 2000$	$Re = 3200$	$Re = 5000$	$Re = 6000$	$Re = 8000$	$Re = 10\,000$
6.76	4.55	3.65	3.36	3.14	3.77	7.45	7.33

[a] Mesh 30×30.

capabilities of the method presented in this work. Note that initially the number of CGA iterations decreases as the Reynolds number (Re) increases. Beyond a critical value of Re, the number of iterations increases again. In spite of this, fast convergence is observed throughout the Re range used. It is interesting to note that from $Re = 6000$ to $Re = 8000$ there is a pronounced increase in the number of iterations of the CGA followed by a tendency to level off. These two values of the Reynolds number coincide with those identified in the experimental investigations of Koseff and Street [24] as the range where parts of the flow appeared to change from laminar to turbulent.

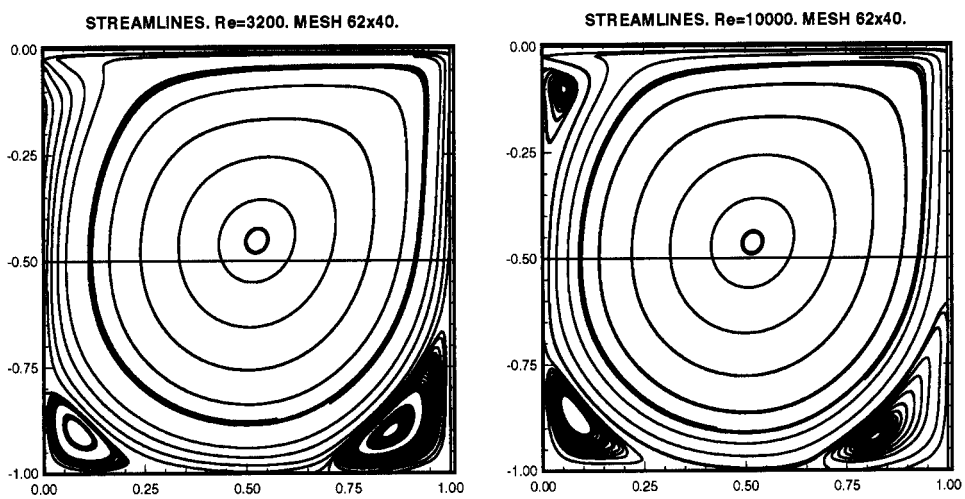

Figure 2. Cavity flow. $Re = 3200$ and $Re = 10000$. Streamlines.

Figure 2 shows streamline contours for $Re = 3200$ and $Re = 10000$ for the 62×40 mesh. The maximum CFL condition for this set of results is 6.2 and the scheme exhibited no sign of any instabilities. This fact is evidence for the good stability of our FEMMC even with quadratic elements. Velocity profiles (Figures 3 and 4) at the mid-lines of the cavity are compared with the experimental three-dimensional results by Koseff and Street [25]. Good agreement is obtained for the finer mesh. Numerical results by Ghia *et al.* [26] are also presented in these illustrations.

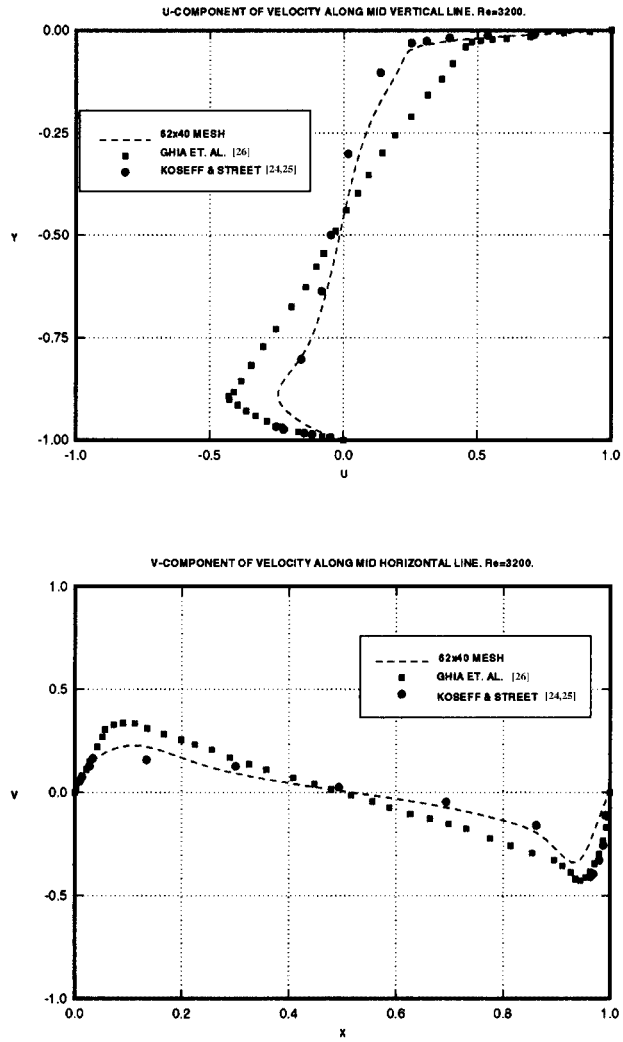

Figure 3. Cavity flow. $Re = 3200$. Velocity profiles at mid-vertical and horizontal lines.

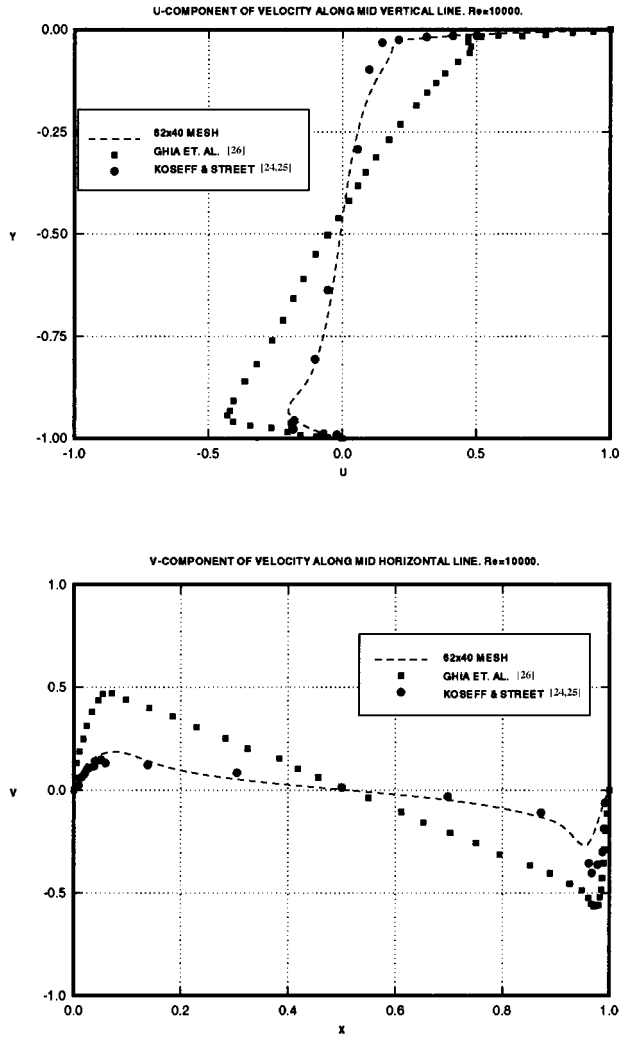

Figure 4. Cavity flow. $Re = 10\,000$. Velocity profiles at mid-vertical and horizontal lines.

5.3.2. Flow around a circular cylinder at Re = 100. In this section we present results corresponding to flow around a circular cylinder at $Re = 100$. Figure 5 shows the relatively coarse finite element grid used [27]. Domain dimensions used in the computations for this case are available from this figure. Uniform flow was assumed at the entrance of the channel. At the open boundary we have imposed the pseudo-stress condition $-p\mathbf{n} + \nu(\partial\mathbf{u}/\partial\mathbf{n}) = 0$. For all remaining boundaries, we have used $\mathbf{u} = 0$. The flow simulation was carried out for 3000 time steps ($0 \leq t \leq 300$).

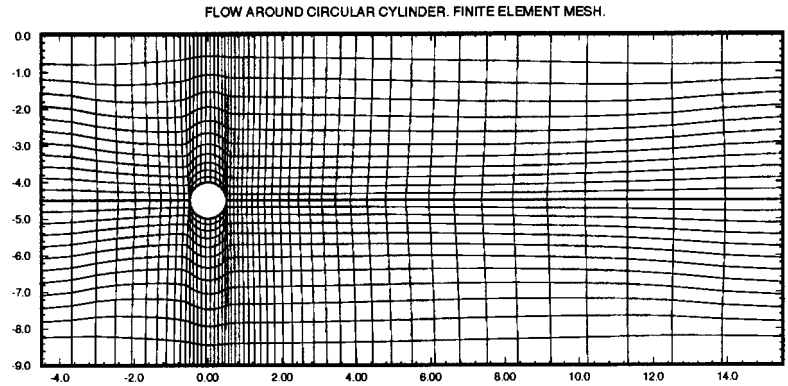

Figure 5. Flow around circular cylinder. $Re = 100$. Finite element mesh.

The computational conditions for this case are shown in Table IV. For this mesh, the number of iterations required for the scheme to converge to a tolerance of 10^{-3} in the maximum norm have been summarized in Table V. About 83% of the entire run required less than 20 iterations to converge.

Figures 6 and 7 show the time evolution of streamlines as the flow progresses past the cylinder. Initially, a pair of symmetric vortices grow behind the cylinder. The vortices become noticeably asymmetrical at approximately $t = 20$ s. From then onwards, they reach unstable proportions and begin shedding alternatively, initiating the Karman vortex street. It should be

Table IV. Computational conditions flow around a circular cylinder at $Re = 100$

Re	Mesh	Nodes p	Nodes (u, v)
100	50×24	1285	4970

Table V. Performance of CGA

Flow around a circular cylinder at $Re = 100$		
	Number of iterations N_{it}	% of Total
$N_{it} < 5$	651	21.69
$5 \leq N_{it} < 10$	164	5.46
$10 \leq N_{it} < 15$	1148	38.24
$15 \leq N_{it} < 20$	518	17.26
$20 \leq N_{it} < 30$	326	10.86
$30 \leq N_{it} < 40$	25	0.83
$40 \leq N_{it} < 50$	2	0.07
$N_{it} > 50$	168	5.59

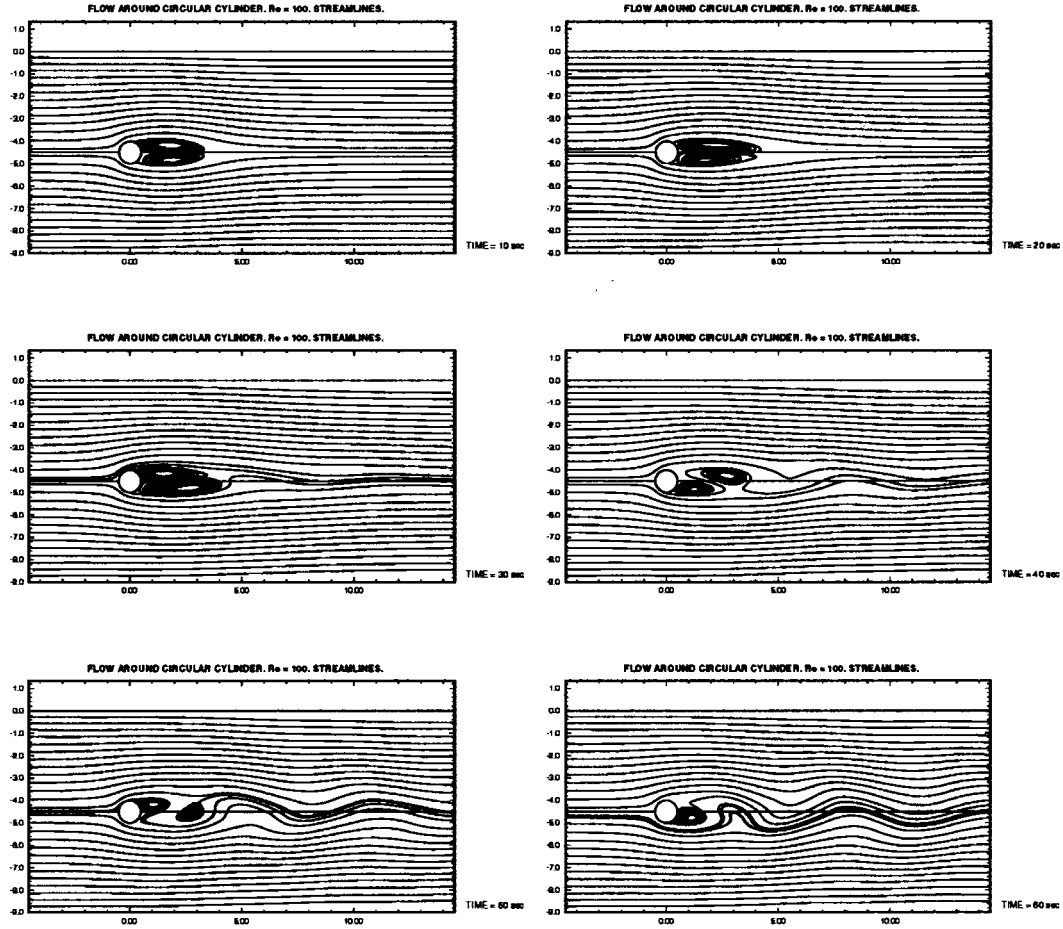

Figure 6. Flow around circular cylinder. $Re = 100$. Streamlines.

pointed out that no artificial triggering was necessary to initiate the asymmetry in the wake nor to provoke vortex shedding.

Force coefficients are shown in Figure 8. The initial sequence of events discussed in the context of Figure 6 corresponds to an essentially zero lift coefficient during the first 20 s of the run, followed by an oscillating tendency of rapidly increasing amplitude. A periodical motion with fairly uniform amplitude is obtained past 80 s. For this section of the curve, the period of oscillation is 6 s and results in a Strouhal number of 0.167. Note that the corresponding time interval for the drag coefficient exhibits fluctuations of higher frequency than the lift. To summarize, we present in Table VI a comparison of our results with numerical and experimental data compiled by Graham [28]. Good agreement is obtained with all variables presented in this reference.

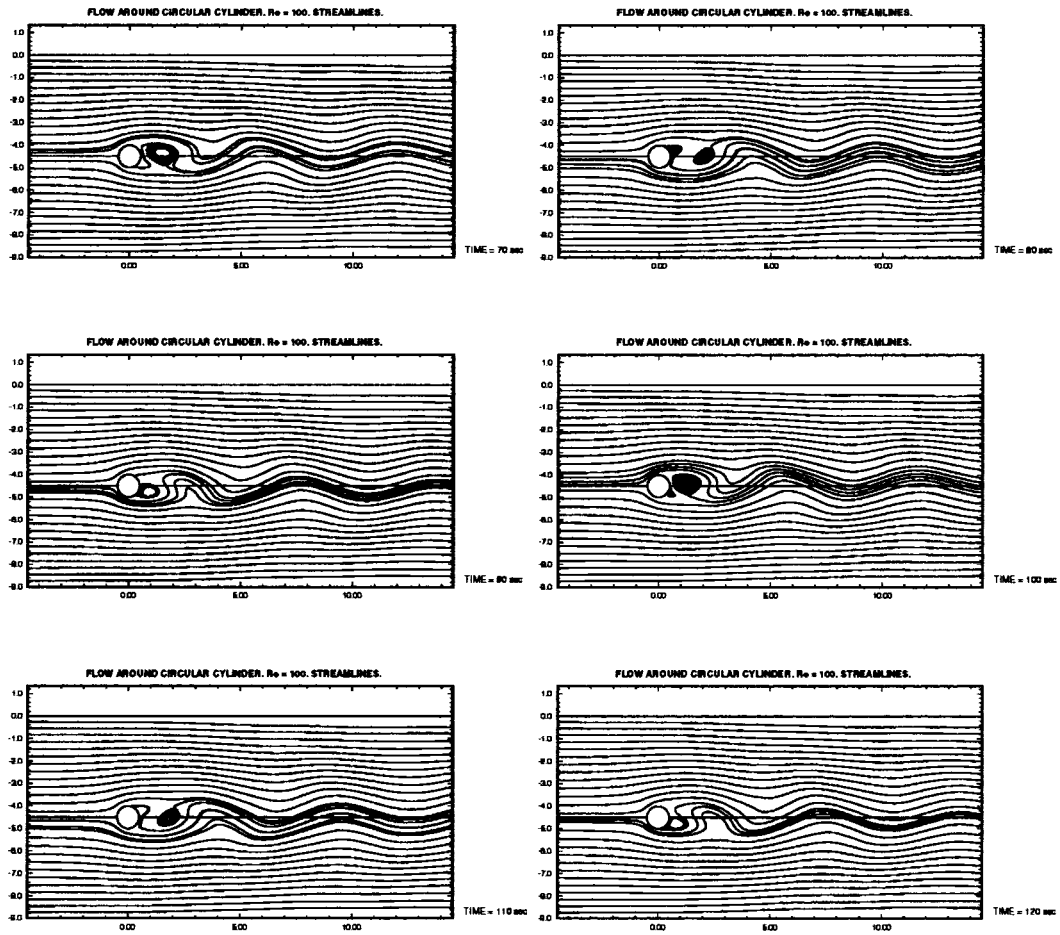

Figure 7. Flow around circular cylinder. $Re = 100$. Streamlines.

6. CONCLUSIONS

A finite element method based on the MMC has been developed based on a more economical and efficient interpolatory convective step. A series of numerical experiments including the unstationary Navier–Stokes equations show promising results when compared with numerical and experimental observations. Experiments for convection–diffusion and Burgers equations show evidence of our method's sound error, convergence and accuracy characteristics. Solutions of the incompressible Navier–Stokes equations for lid-driven cavity flow at a wide range of Reynolds numbers and for flow around a circular cylinder at $Re = 100$ compared very favorably with published data.

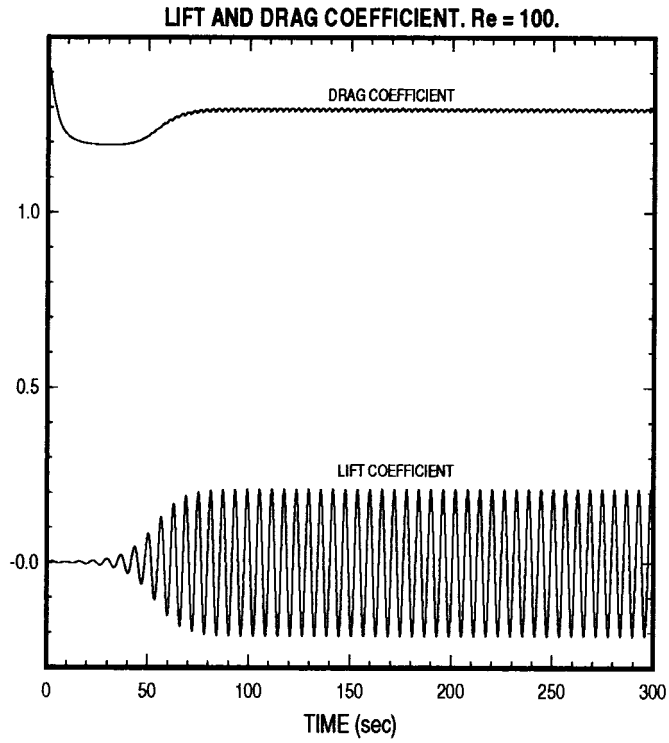

Figure 8. Flow around circular cylinder. $Re = 100$. Force coefficients.

Table VI. Comparison of results

Flow around a circular cylinder at $Re = 100$		
	Graham [24]	This work
Strouhal number	0.16	0.167
Mean drag coefficient	1.25–1.46	1.295
RMS of fluctuating drag coefficient	0.0042–0.04	0.0085
RMS of fluctuating lift coefficient	0.157–0.39	0.162

ACKNOWLEDGMENTS

Dr A. Allievi gratefully acknowledges the support of FSI Technologies while this work was conducted. Dr R. Bermejo was supported by grant CLI95-1823 from the Comisión Interministerial de Ciencia y Tecnología de España.

REFERENCES

1. J.P. Benqué, B. Iblier, A. Keramsi and G. Labadie, ' A finite element method for Navier–Stokes equations coupled with a temperature equation', in *Fourth International Symposium on Finite Elements in Flow Problems*, North-Holland, Amsterdam, 1982, pp. 295–301.
2. J. Douglas and T.F. Russell, 'Numerical methods for convection-dominated diffusion problems based on combining the method of characteristics with finite element or finite difference procedures', *SIAM J. Numer. Anal.*, **19**, 871–885 (1982).
3. O. Pironneau, 'On the transport-diffusion algorithm and its applications to the Navier–Stokes equations', *Numer. Math.*, **38**, 309–322 (1982).
4. A. Allievi and R. Bermejo, 'A generalized particle search–locate algorithm for arbitrary grids', *J. Comput. Phys.*, **132**, 157 (1997).
5. J. Cahouet and J.P. Chabard, 'Some fast 3D finite element solvers for the generalized Stokes problem', *Int. J. Numer. Methods Fluids*, **8**, 869–895 (1988).
6. E.J. Dean and R. Glowinski, 'On some finite element methods for the numerical simulation of incompressible viscous flow', in M.D. Gunzburger and R.A. Nicolaides (eds.), *Incompressible Computational Fluid Dynamics*, Cambridge University Press, Cambridge, UK, 1993, pp. 17–66.
7. M.O. Bristeau, R. Glowinski and J. Periaux, 'Numerical methods for the Navier–Stokes equations. Applications to the simulation of compressible and incompressible viscous flows', *Comput. Phys. Rep.*, **6**, 73–187 (1987).
8. P.M. Gresho, 'Some current CFD issues relevant to the incompressible Navier–Stokes equations', *Comput. Methods Appl. Mech. Eng.*, **87**, 201–252 (1991).
9. R. Temam, 'Navier–Stokes equations: theory and numerical analysis', in *Theory and Numerical Analysis of the Navier–Stokes Equations*, North-Holland, Amsterdam, 1985.
10. O. Ladyzhenskaya, 'The mathematical theory of viscous incompressible flow', in *The Mathematical Theory of Viscous Incompressible Flow*, Gordon & Breach, New York, 1969.
11. J.G. Heywood, 'The Navier–Stokes equations: on the existence, regularity and decay of solutions', *Indiana Univ. Math. J.*, **29**, 639–681 (1980).
12. P.L. Lions, 'Mathematical topics in fluid mechanics', in *Oxford Lecture Series in Mathematics and its Applications*, *Vol. 1: Incompressible Models*, Oxford University Press, Oxford, 1996.
13. C. Foias, C. Guillopé and R. Temam, 'Lagrangian representation of the flow', *J. Diff. Eqns.*, **57**, 440–449 (1985).
14. J.P. Huffenus and D. Khaletzky, 'A finite element method to solve the Navier–Stokes equations using the method of characteristics', *Inter. J. Numer. Methods Fluids*, **4**, 247–269 (1984).
15. R. Bermejo, 'A Galerkin-characteristic algorithm for transport-diffusion equations', *SIAM J. Numer. Anal.*, **32**, 425 (1995).
16. C. Temperton and A. Staniforth, 'An efficient two-time level semi-Lagrangian semi-implicit integration scheme', *Q. J. R. Meteorol Soc.*, **113**, 1025–1039 (1987).
17. A. Robert, 'A stable numerical integration scheme for the primitive meteorological equations', *Atmos–Ocean*, **19**, 35–46 (1981).
18. V. Girault and P.A. Raviart, 'Finite element methods for Navier–Stokes equations: theory and algorithms', in *Finite Element for Navier–Stokes Equations*, Springer, Berlin, 1986.
19. R. Verfurth, 'Error estimates for a mixed finite element approximation of the Stokes equations', *RAIRO Numer. Anal.*, **18**, 175–182 (1984).
20. R. Stenberg, 'Error analysis of some finite element methods for the Stokes problem', *Math. Comp.*, **54**, 495–508 (1992).
21. J.P. Bercovier and O. Pironneau, 'Error estimates for finite element solution of the Stokes problem in the primitive variables', *Numer. Math.*, **33**, 211 (1979).
22. P. Ciarlet, 'The finite element method for elliptic problems', in *The Finite Element Method for Elliptic Problems*, North-Holland, Amsterdam, 1978.
23. J. Pudykiewicz and A. Staniforth, 'Some properties and comparative performance of semi-Lagrangian method of Robert in the solution of the advection–diffusion equation', *J. Atmos. Sci.*, **31**, 371–393 (1984).
24. J.R. Koseff and R.L. Street, 'Visualization studies of a shear driven three-dimensional recirculation flow', *ASME J. Fluids Eng.*, **106**, 21 (1984).
25. J.R. Koseff and R.L. Street, 'The lid-driven cavity flow: a synthesis of qualitative and quantitative observations', *ASME J. Fluids Eng.*, **106**, 390 (1984).
26. U. Ghia, K.N. Ghia and C.T. Shin, 'High-*Re* solutions for the incompressible flow using the Navier–Stokes equations and a multigrid method', *J. Comput. Phys.*, **48**, 387 (1982).
27. A. Allievi and S.M. Calisal, 'A Bubnov–Galerkin formulation for orthogonal grid generation', *J. Comput. Phys.*, **98**, 163 (1992).
28. J.M.R. Graham, 'The effects of waves on vortex shedding from cylinders', *IUTAM Symposium on Bluff-Body Wakes, Dynamics and Instabilities*, Gottingen, Germany, 1992.

