INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 15, 23–36 (1992)

IMPLEMENTATION OF THE LAGRANGE–GALERKIN METHOD FOR THE INCOMPRESSIBLE NAVIER–STOKES EQUATIONS

GUSTAVO C. BUSCAGLIA AND ENZO A. DARI

Centro Atómico Bariloche, 8400 SC de Bariloche, Rio Negro, Argentina

SUMMARY

This paper is concerned with the implementation of Lagrange–Galerkin finite element methods for the Navier–Stokes equations. A scheme is developed to efficiently handle unstructured meshes with local refinement, using a quad-tree-based algorithm for the geometric search. Several difficulties that arise in the construction of the right-hand side are discussed in detail and some useful tricks are proposed.

The resulting method is tested on the lid-driven square cavity and the vortex shedding behind a rectangular cylinder and is found to give satisfactory agreement with previous works. A detailed analysis of the effect of time discretization is included.

KEY WORDS Incompressible Navier–Stokes equations Finite element method Lagrange–Galerkin method Geometric search algorithm Vortex shedding

1. INTRODUCTION

Efficient solution of the incompressible Navier–Stokes equations is of primary importance in many applications of computational fluid dynamics. Several difficulties appear in the numerical treatment of these equations: indefinite and usually ill-conditioned matrices, convection-dominated diffusion of momentum at large Reynolds numbers, non-linearities coming from acceleration terms, and restrictions in the choice of interpolants for velocities and pressure so as to satisfy the well known Babuska–Brezzi (BB) condition (see e.g. References 1–3).

In this paper we will focus our attention on the development of compact codes which appropriately handle the above-mentioned difficulties. For that purpose local mesh refinement is needed so that boundary layers can be resolved without introducing unnecessary degrees of freedom in regions where no steep gradients are expected. This leads to the use of unstructured meshes and we have chosen triangular elements to satisfy this requirement. However, it is by now established that Galerkin weighting gives oscillatory (wiggly) results at large Reynolds numbers even on very fine meshes, and a method with better stability properties is needed to make realistic problems tractable.

We have adopted the Lagrange–Galerkin method (LGM),[4–6] which satisfies this requirement. It is based upon a Lagrangian frame treatment of the material derivatives and also provides a robust way to deal with the non-linear convective terms via time evolution. Once the LGM is used, there remains to solve a Stokes problem at each time step together with the evaluation of the right-hand side. The computational implementation of this last item is essential for the resulting code to be competitive and will be given detailed explanation below.

0271–2091/92/130023–14$12.00

Received February 1991
Revised October 1991

The plan of this paper is as follows. In Section 2 we state the continuous problem and perform the time discretization that leads to the LGM. Section 3 is devoted to the finite element approximation of the resulting set of equations. At this point it is seen that exact integration of the right-hand side is not feasible and Section 4 includes a full description of the numerical scheme we implemented. Throughout Sections 3 and 4 some alternatives to the usual implementation of the LGM are developed. These are compared in the numerical examples of Section 5 and allow a comprehensive analysis of the effect of time discretization. Finally, some conclusions are drawn in Section 6.

2. THE CONTINUOUS PROBLEM AND ITS DISCRETIZATION BY THE LGM

The dynamical behaviour of an incompressible fluid of density ρ and viscosity μ is governed by the Navier–Stokes equations

$$\rho\left(\frac{\partial \vartheta_i}{\partial t}+\vartheta_j\frac{\partial \vartheta_i}{\partial x_j}\right)-\frac{\partial}{\partial x_j}\left[\mu\left(\frac{\partial \vartheta_i}{\partial x_j}+\frac{\partial \vartheta_j}{\partial x_i}\right)\right]+\frac{\partial p}{\partial x_i}=f_i, \tag{1}$$

$$\frac{\partial \vartheta_i}{\partial x_i}=0, \tag{2}$$

where $\vec{\vartheta}$ is the velocity field, p is the pressure and $\vec{f}$ represents the body forces. Appropriate conditions to solve (1) and (2) inside a bounded domain Ω are, for example, to specify the initial velocity field

$$\vartheta_i(x,0)=V_i(x), \quad \forall x\in\Omega, \tag{3}$$

and boundary data, such as velocities or surface tractions, on all the boundary.

Now let us recall from elementary mechanics that the acceleration is the time derivative of the velocity field along the path lines of the fluid particles, i.e.

$$a_i(x,t)=\frac{\partial \vartheta_i}{\partial t}(x,t)+\vartheta_j(x,t)\frac{\partial \vartheta_i}{\partial x_j}(x,t)=\lim_{t'\to t}\frac{\vartheta_i(x,t)-\vartheta_i(\underset{\sim}{x},t')}{t-t'}, \tag{4}$$

where $\underset{\sim}{x}$ stands for the position at time t' of the particle that passes through x at time t. The LGM[4-6] is based upon a natural approximation of equation (4). If we choose a time step Δt and define

$$\vartheta_i^n(x)=\vartheta_i(x,t_n), \tag{5}$$

where t_n belongs to the chosen time discretization, we can approximate the acceleration terms by the *one-step scheme*

$$a_i(x,t_n)\simeq\frac{\vartheta_i^n(x)-\vartheta_i^{n-1}(\underset{\sim}{x})}{\Delta t}. \tag{6}$$

This is the usual practice. We propose, alternatively, the second-order approximation (*two-step scheme*)

$$a_i(x,t_n)\simeq\frac{3\vartheta_i^n(x)-4\vartheta_i^{n-1}(\underset{\sim}{x})+\vartheta_i^{n-2}(\underset{\approx}{x})}{2\Delta t}, \tag{7}$$

where now $\underset{\sim}{x}$ (resp. $\underset{\approx}{x}$) stands for the position at time t_{n-1} (resp. t_{n-2}) of the particle that passes through x at time t_n.

Collecting the previous results and with implicit treatment of the other terms, we have

$$\frac{\rho}{\Delta t}\,\vartheta_i^n(x)-\frac{\partial}{\partial x_j}\left[\mu\left(\frac{\partial \vartheta_i^n}{\partial x_j}+\frac{\partial \vartheta_j^n}{\partial x_i}\right)\right]+\frac{\partial p^n}{\partial x_i}(x)=f_i^n(x)+\frac{\rho}{\Delta t}\,\vartheta_i^{n-1}(\underline{x}), \tag{8}$$

$$\frac{\partial \vartheta_i^n}{\partial x_i}=0. \tag{9}$$

Expressions for the two-step scheme are analogous and will not be included for brevity.

It should be noted that in the neighbourhood of inflow boundaries some of the $\underline{x}$s will fall outside Ω. This is due to the particle-following properties of the Lagrangian representation and must be considered during the spatial discretization.

3. FINITE ELEMENT APPROXIMATION

Equations (8) and (9) at each time t_n give rise to a generalized Stokes problem which can receive standard finite element treatment. As stated in Section 1, we chose triangular elements for all fields so as to work with unstructured meshes. To be more specific, we implemented the $4 \times P_1/P_1$ element[1,3] (also called the Taylor–Hood element) consisting of four equal linear and conforming subtriangles inside the (also linear and conforming) pressure element (see Figure 1).

We now perform the usual Galerkin finite element weighting of equations (8) and (9), obtaining the linear system that corresponds to the so-called 'direct' LGM (we have adopted the nomenclature of Reference 7):

$$\underline{\underline{K}}_{vv}\underline{V}^n-\underline{\underline{K}}_{vp}\underline{P}^n=\underline{B}^n, \tag{10}$$

$$\underline{\underline{K}}_{vp}^{\mathrm{T}}\underline{V}^n=\underline{0}, \tag{11}$$

where $\underline{V}^n$ and $\underline{P}^n$ are the velocity and pressure unknowns at step n and

$$\underline{\underline{K}}_{vv}^{IJ}=\frac{\rho}{\Delta t}\int_\Omega N_i^I N_i^J\,\mathrm{d}x+\int_\Omega \frac{\mu}{2}\left(\frac{\partial N_i^I}{\partial x_j}+\frac{\partial N_j^I}{\partial x_i}\right)\left(\frac{\partial N_i^J}{\partial x_j}+\frac{\partial N_j^J}{\partial x_i}\right)\mathrm{d}x, \tag{12}$$

$$\underline{\underline{K}}_{vp}^{IJ}=\int_\Omega \frac{\partial N_i^I}{\partial x_i}M^J\,\mathrm{d}x, \tag{13}$$

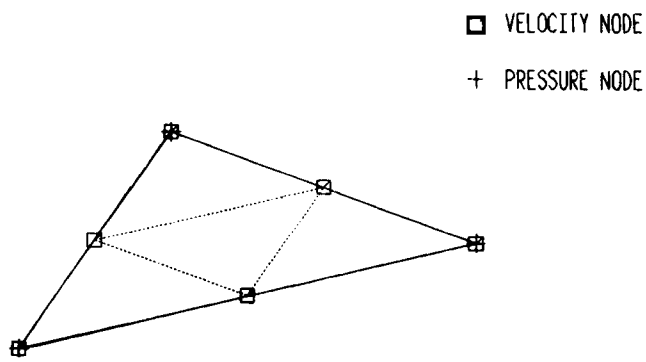

Figure 1. $4 \times P_1/P_1$ mixed element

with $\vec{N}^I$ (M^I) the Ith velocity (pressure) basis function. Also,

$$(\underline{B}^n)^I = \frac{\rho}{\Delta t}\int_\Omega \vec{\vartheta}_i^{n-1}(\underline{x})N_i^I(x)\,dx + \rho\int_\Omega f_i^n N_i^I\,dx + \int_{\partial\Omega} g_i^n N_i^I\,d\Gamma, \tag{14}$$

where $\vec{g}$ is the surface traction. The first integral in (14) is not tractable by analytical means, since $\vec{\vartheta}^{n-1}(\underline{x})$ is a quite general (far from polyhedral in most cases) continuous function. In the next section we describe a numerical scheme to approximate this integral.

In what concerns the solution of the linear system (10), (11) we used a conjugate gradient algorithm on pressure unknowns, with the preconditioner proposed by Cahouet and Chabard.[8]

4. EVALUATION OF THE RIGHT-HAND SIDE

It is known that the LGM, when integration is exact, leads to an unconditionally stable, conservative scheme. Morton *et al.*[7] have shown that approximate evaluation of the right-hand side renders the scheme not only non-conservative but also conditionally unstable in most cases. They proposed a variant, called 'area weighting', that recovers unconditional stability but unfortunately does not work on unstructured meshes. Therefore we implemented non-exact integration by numerical quadrature, but care was taken in the selection of the quadrature points (see Section 4.3). In this way the positions $\underline{x}$ must be found only for the finite number of points chosen for quadrature.

4.1. Particle tracking

The ordinary differential equation for the path line is

$$\frac{dX}{dt} = \vec{\vartheta}(X(t), t), \tag{15}$$

with initial condition $X(t_n) = x$. Once (15) is solved backwards in time, we get $\underline{x} = X(t_{n-1})$ and $\underline{\underline{x}} = X(t_{n-2})$. However, since the velocity field at time t_n is not known at the moment of constructing $\underline{B}^n$, $\vec{\vartheta}$ in (15) must be replaced by some approximation, say $\vec{\beta}$. We implemented two possibilities.

$\vec{\beta}$ fixed. Using the last computed value, $\vec{\beta}(\cdot, t) = \vec{\vartheta}^{n-1}(\cdot)$.

$\vec{\beta}$ linear in time. A linear estimate according to the last two values is

$$\vec{\beta}(\cdot, t) = \vec{\vartheta}^{n-1}(\cdot) + \frac{t - t_{n-1}}{\Delta t}\,[\vec{\vartheta}^{n-1}(\cdot) - \vec{\vartheta}^{n-2}(\cdot)], \tag{16}$$

for $t \in [t_{n-2}, t_n]$. The resulting ordinary differential equation for the path lines,

$$\frac{dX}{dt} = \vec{\beta}(X(t), t), \tag{17}$$

was solved using a second-order predictor–corrector method with step Δt *or a fraction of it*, $\Delta t/NP$. We call *substepping* this option of subdividing the time step for particle tracking. Clearly, as $NP \to \infty$, (17) is solved exactly.

It must be noted that the numerical solution of (17) requires the evaluation of the velocity field at points not belonging to the mesh. This item is addressed in Section 4.2. Also, $\underline{x}$ or $\underline{\underline{x}}$ can lie *outside the domain*. This may be due to two situations.

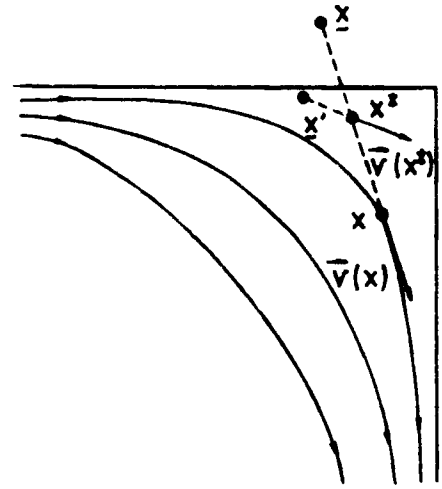

Figure 2. Schematic diagram showing how substepping prevents $\underset{\sim}{x}$ from spuriously falling outside Ω

(a) x is near an inflow boundary. Generally, at inflow boundaries fully developed velocities are prescribed. In this case we assign to $\tilde{\vartheta}^{n-1}(\underset{\sim}{x})$ the prescribed value at the point where the particle entered the domain.
(b) Numerical errors in solving equation (17). Any time a path line leaves the domain, we apply substepping with $NP = 16$ to check if the cause was finite step integration of (17). In this way spurious exits near corners are removed (see Figure 2).

4.2. Geometric search

The LGM, as described above, leads to several evaluations of the velocity field at positions not coincident with any node of the mesh. This can be a very expensive operation if programmed with ingenuity. Below we describe three algorithms, of increasing complexity, that accomplish the geometric search function.

The mesh element K that contains a given point $X = (x, y)$ is to be found in order to interpolate the velocity field at X from its values at the nodes of K.

Algorithm NA (naive)

loop $I = 1$, number of elements in the mesh
  test if X belongs to I
  if this is the case, return $K = I$
  else continue
end loop

Algorithm SAM (structured auxiliary mesh)[9,10]

1. Preprocessing step (outside the temporal loop)
   (a) Construct an auxiliary mesh formed by rectangles and store the locations $X_i\,(i = 1, NX)$ and $Y_i\,(i = 1, NY)$ of the dividing (vertical and horizontal respectively) lines. Rectangle R_{ij} is $[X_i, X_{i+1}] \times [Y_j, Y_{j+1}]$. Of course, the auxiliary mesh must cover the domain (see Figure 3).

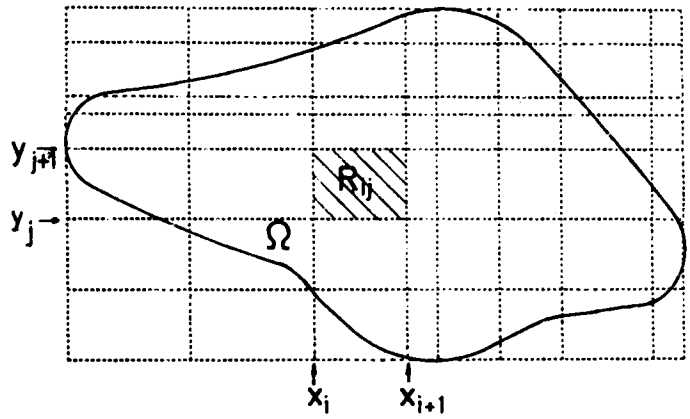

Figure 3. An example of the structured auxiliary mesh

(b) For every R_{ij} construct a table T_{ij} containing the indices of the original mesh elements that intersect R_{ij}.

2. Search step (inside the temporal loop)
   (a) Find the rectangle R_{ij} that contains X (this is easily done since the auxiliary mesh is structured).
   (b) Sweep table T_{ij} to find the element K of the original mesh that contains X.

Algorithm QT (quad-tree structure).[11] This is a variant of the previous algorithm, replacing the auxiliary rectangular mesh by a quad-tree. The quad-tree leaves are automatically subdivided to account for local densifications of the original mesh.

By inspection of the algorithms, Algorithm NA is clearly seen to be of highest order. In fact, since at least one search must be performed for each quadrature point, Algorithm NA is $O(N^2)$. This order soon renders it prohibitive on increasing the number of degrees of freedom. Algorithms SAM and QT are similar, but QT handles meshes with local refinement more efficiently and is to be preferred. However, the best choice turned out to be a *combination* of SAM and QT, each rectangle of an auxiliary *regular* mesh being given a quad-tree structure. In Figure 4 we show the resulting auxiliary structure for the vortex-shedding experiment to be described in Section 5. With this method the geometric search becomes inexpensive when compared to the solution of the linear system.

4.3. *Numerical quadrature*

We performed the first integral in equation (13) numerically by evaluating the integrand at a finite set of points and weighting this value according to the quadrature rule. We tested several quadratures for the simpler problem of advection–diffusion of a scalar quantity. We summarize the possible choices in Table I.

An analysis of the effect of these quadratures on the LGM has been reported by Bermúdez *et al.*[12] for steady state (non-linear) problems. They find that all the formulae in Table I lead to acceptable solutions. However, our results on the pure advection of a cone have shown that formulae 2 and 3 are *unstable* for this transient problem. We do not include the full analysis here for brevity, but formulae 2 and 3 are excluded in the following because of this reason (formula 2 is also reported unstable in Reference 11).

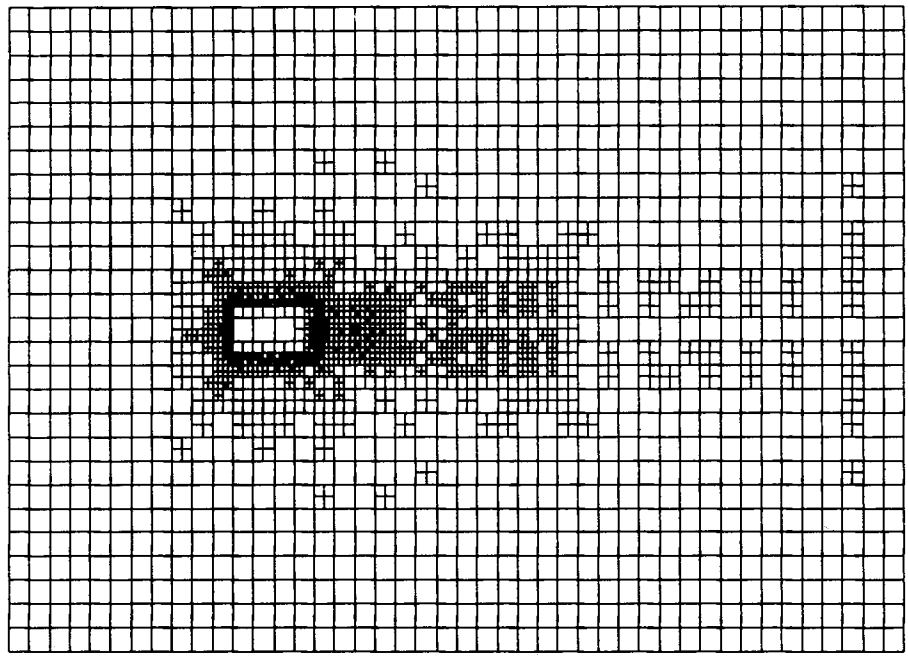

Figure 4. Auxiliary structure automatically generated in the vortex-shedding experiment (corresponds to mesh RECT2)

Table I. Quadrature formulae

Formula	Points	Triangular coordinates	Multiplicity	Weight	Order
1	3	(1/2, 1/2, 0)	3	1/3	2
2	3	(1/6, 1/6, 2/3)	3	1/3	2
3	4	(1/3, 1/3, 1/3)	1	−27/48	3
		(1/5, 1/5, 3/5	3	27/48	
4	6	(0·81685, 0·09157, 0·09157)	3	0·1099517	4
		(0·10810, 0·44595, 0·44595)	3	0·2233816	

The above-mentioned tests for the pure transport of a cone also showed that formula 4 is not suitable, since it is four times more costly than formula 1 but brings no extraordinary improvement.

We chose formula 1 and studied its behaviour in what concerns lack of conservativity and numerical diffusion. Both effects were seen to decrease with mesh refinement. The numerical diffusion increased with smaller time steps. This is in agreement with a term $O(h^2/\Delta t)$ in the error bound for our interpolation[2,13] and implies the existence of an optimal time step of order h.

5. NUMERICAL TESTS; THE EFFECT OF THE TIME STEP

In this section we discuss the performance of the above method on two well known test examples: the lid-driven square cavity (see Reference 3 for a comparison of several finite element methods) and the vortex shedding behind a rectangular cylinder (see Reference 14 for experimental data

and Reference 15 for numerical predictions). Our aim is to analyse the relative importance of the different mechanisms through which the time step enters the discretization procedure. These mechanisms are the following:

(i) truncation of the limit for the material derivative (equation (6))
(ii) finite step size integration of equation (17)
(iii) approximation of $\vec{\vartheta}$ by $\vec{\beta}$ in equation (15)
(iv) implicit treatment of the remaining terms in equation (8).

Throughout the previous sections, several alternatives to the standard implementation of the LGM were introduced. Our purpose was to develop the tools to study the effects of (i), (ii) and (iii) separately. In fact, (i) alone is modified by switching from the *one-step scheme* to the *two-step scheme*, the effect of (ii) can be isolated by *substepping* and that of (iii) by taking $\vec{\beta}$ either *fixed* or *linear in time* (equation (16)).

Notice that (iii) and (iv) do not affect steady solutions. For this reason we begin with a stationary problem.

For the results to be presented hereafter, if nothing is explicitly said, it must be assumed that the *standard* LGM was used (no *substepping*, *one-step scheme*, $\vec{\beta}$ *fixed*).

5.1. *The lid-driven square cavity*

The problem is to find the steady state 2D flow inside a unit square cavity with the boundary conditions

$$\vartheta_1(x_1, x_2=1)=1, \qquad \vartheta_1(x_1=0, x_2)=\vartheta_1(x_1, x_2=0)=\vartheta_1(x_1=1, x_2)=0,$$
$$\vartheta_2(x_1=0, x_2)=\vartheta_2(x_1=1, x_2)=\vartheta_2(x_1, x_2=0)=\vartheta_2(x_1, x_2=1)=0$$

for a fluid with unit density. The Reynolds number of this flow is defined as $Re=1/\mu$. We solved this problem for $Re=100$ and 400 with the mesh shown in Figure 5 (1003 unknowns) and several

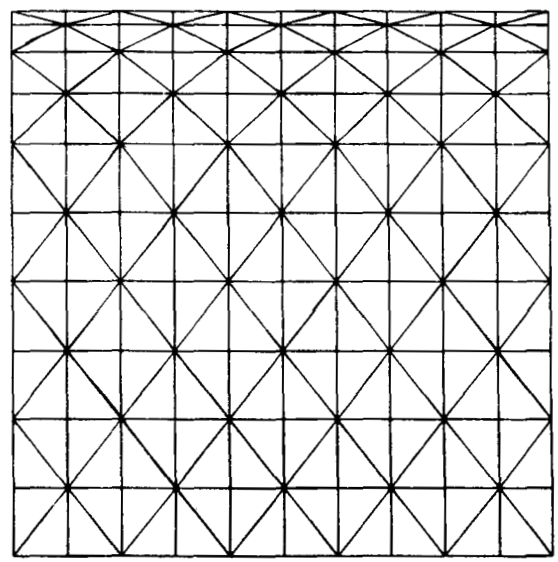

Figure 5. Finite element mesh for the lid-driven square cavity problem

time steps. In Figures 6 and 7 we plot the profiles of ϑ_1 along the vertical centreline for both Re as compared with a reference computation made on a 40×40 mesh (3803 unknowns) with a Galerkin method. Also included are the results at $Re = 0$ (creeping flow) to allow for comparison. Clearly, the steady solution depends on Δt. As previously mentioned, this may come from (i) or (ii) above. We thus redid the calculations with *substepping* but no improvement was noticed even with $NP = 8$. From this we conclude that the first-order approximation of the material derivative in equation (6) is responsible for the errors arising from time discretization. A cure for this,

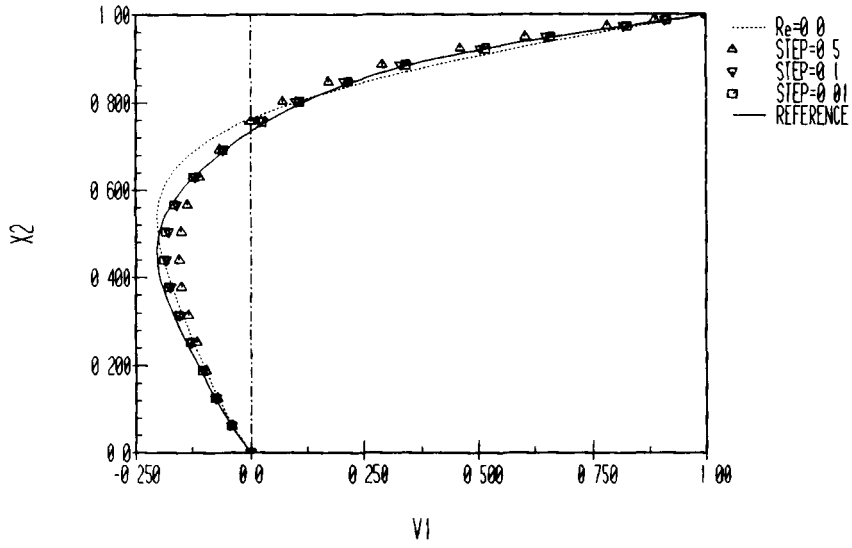

Figure 6. Profiles of horizontal velocity along vertical centreline at $Re = 100$: effect of time step

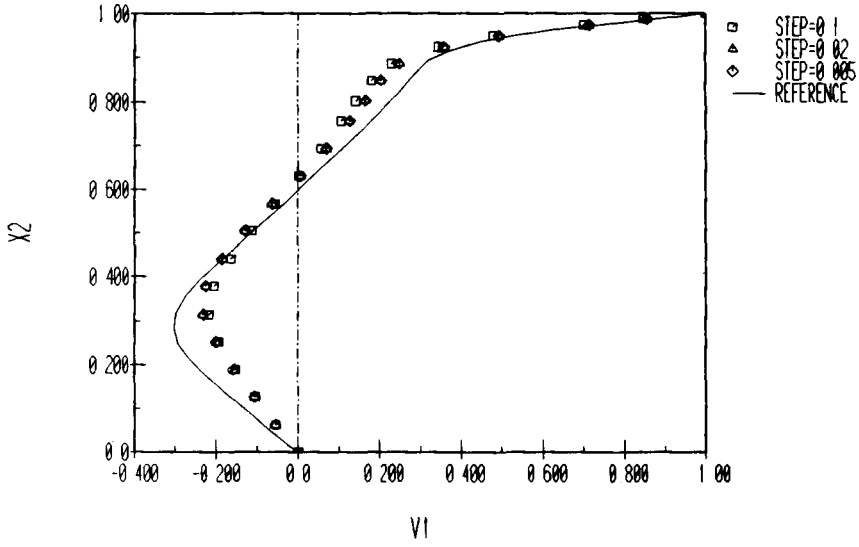

Figure 7. Same as Figure 6 but at $Re = 400$

allowing larger time steps to be used, is the *two-step scheme* (equation (7)). In Figures 8 and 9 we compare the profiles obtained at $Re = 100$ and 400 with $\Delta t = 0{\cdot}1$ by both schemes. The improvement is clear and computing times increase only slightly since the geometric search is rendered inexpensive by the algorithm of Section 4.2.

We now turn to the study of unsteady flows so as to complete the picture.

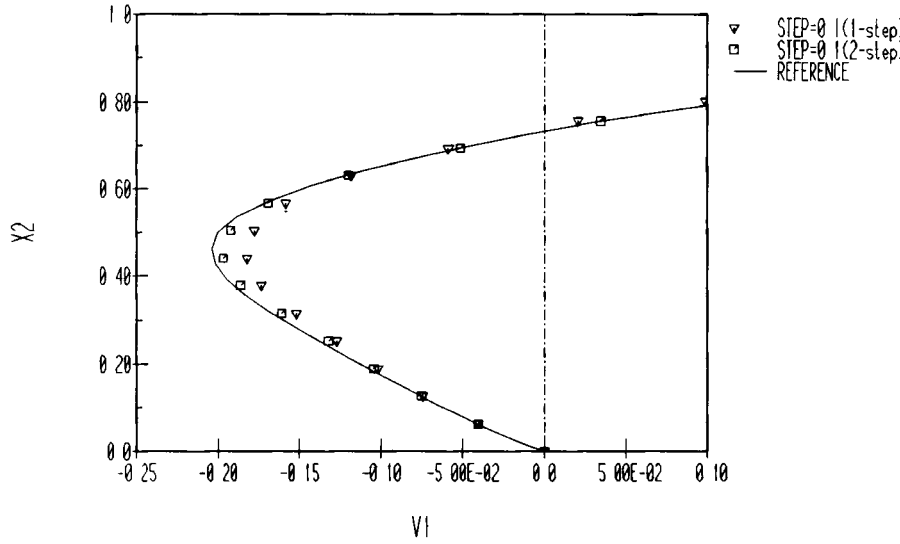

Figure 8. Comparison of one-step and two-step schemes at $Re = 100$

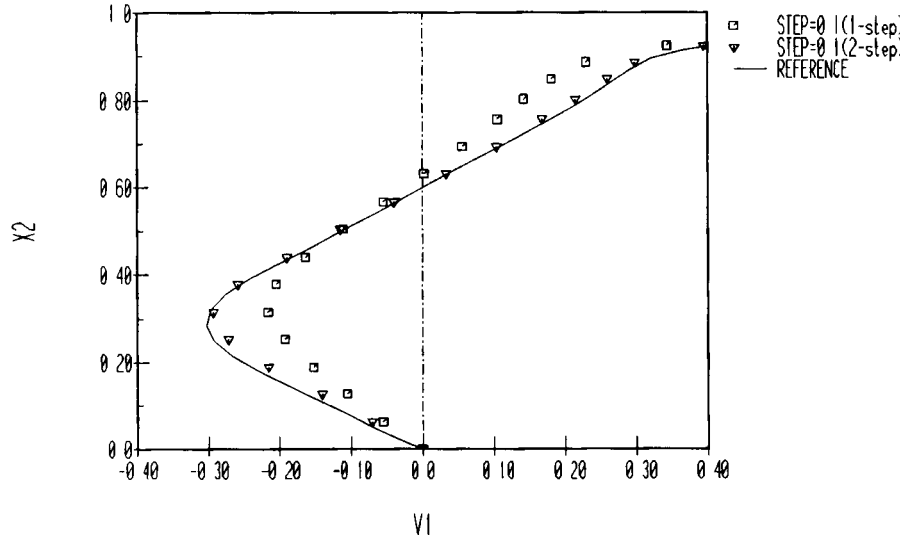

Figure 9. Same as Figure 8 but at $Re = 400$

5.2. Vortex shedding behind a rectangular cylinder

We consider here the 2D flow past a rectangular obstacle of a fluid with unit freestream velocity, viscosity μ and unit density. Re is again $1/\mu$, since we have assumed the obstacle to have unit width. Adherence conditions are imposed at the obstacle, the exit (right) is traction-free, a uniform profile is imposed at the inlet (left) and symmetry is assumed at the top and bottom boundaries. No mass forces are present and also no perturbation is needed to obtain an oscillatory flow with vortices periodically leaving the obstacle. A typical dimensionless number for comparison is the Strouhal number (St), which with our definitions equals the frequency of the shedding.

It is most interesting to make the rectangle have an aspect ratio (length/width) of two. For this geometry St is seen to increase monotonically with Re in the range $60 < Re < 400$ from $\sim 0{\cdot}10$ to $\sim 0{\cdot}17$. This sensitivity allows a rough evaluation of the accuracy of the results by comparing computed St with reported values.

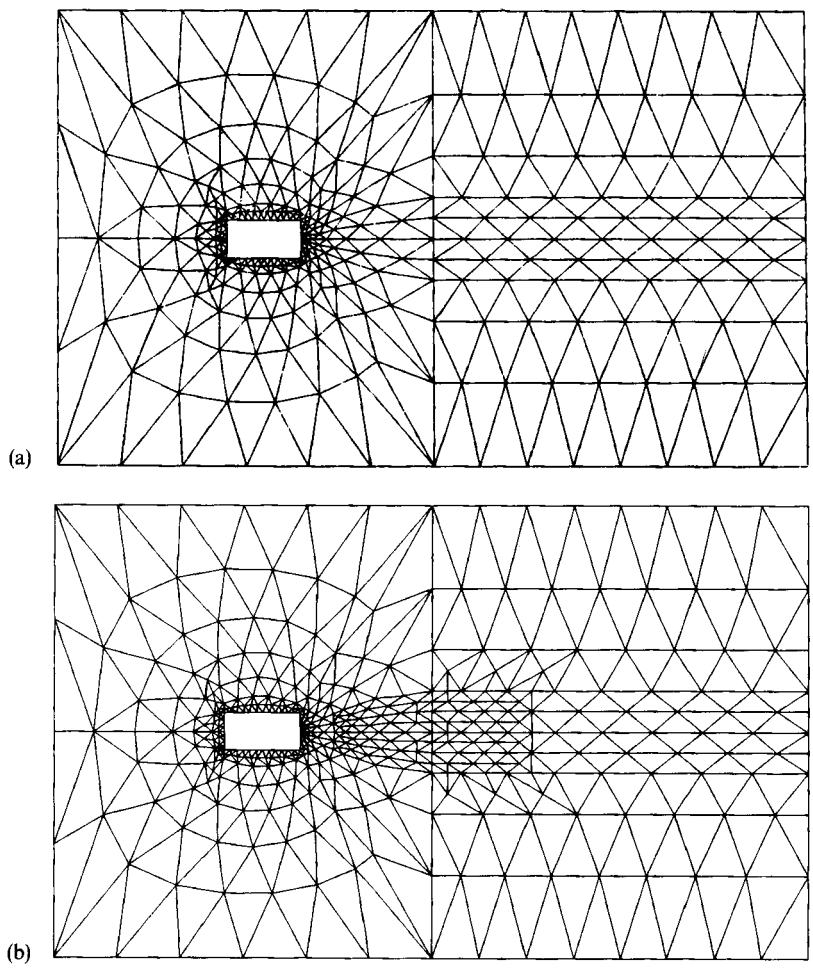

Figure 10. Triangulations used for vortex shedding behind a rectangular cylinder: (a) RECT1; (b) RECT2

We performed this simulation on the meshes shown in Figure 10. Some information about them can be seen in Table II. Mesh RECT1 is dense only around the obstacle, to capture boundary layers, while RECT2 allows better modelling of the vortices.

Our first observation was that mesh RECT1 is inadequate to simulate the flow at $Re > 100$. St for $Re = 80$ was 0·104, which lies within the reported range, but at $Re = 150$ it had only moved to 0·110 and at $Re = 300$ it even lowered to 0·109. These results could not be improved by reducing the time step and thus the errors mainly come from spatial discretization.

Mesh RECT2, on the other hand, allowed a thorough analysis of the time discretization. A comparison of our computed St with experimental values[14] can be seen in Table III. We will focus on the case $Re = 300$, which exhibits a strong dependence on Δt that cannot be removed by *substepping* or by applying the *two-step scheme* (see Table III). We switched from taking $\vec{\beta}$ *fixed* to the *linear approximation* of equation (16), keeping the *two-step scheme* and $\Delta t = 0{\cdot}25$. With this choice the computed St was 0·155. These results confirm that when the velocity field varies rapidly with time, tracing back the particle path lines with frozen velocities can be an important source of time discretization errors.

Table II. Some information concerning meshes RECT1 and RECT2

Mesh	Pressure elements	Pressure nodes	Velocity elements	Velocity nodes	Unknowns
RECT1	574	329	2296	1232	2793
RECT2	722	403	2888	1528	3459

Table III. Numerical Strouhal numbers obtained with mesh RECT2

Re	Step = 0·1 (two-step)	Step = 0·1 (one-step)	Step = 0·25 (two-step)	Step = 0·25 (one-step)	Experimental (Reference 14)
80	0·113	—	—	—	0·10–0·12
150	0·126	0·127	0·124	0·122	0·13–0·15
300	0·137	0·134	0·126	0·126	0·15–0·17

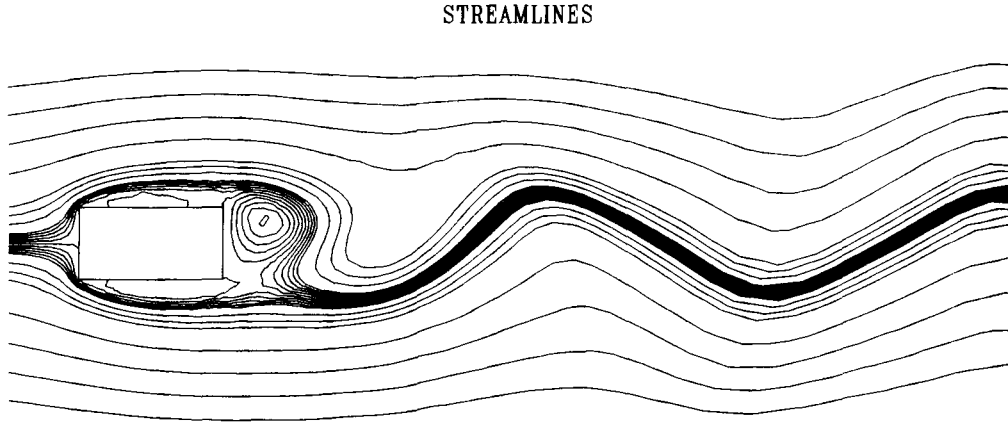

Figure 11. Plot of instantaneous streamlines near the obstacle at $Re = 300$ (mesh RECT2, $\Delta t = 0{\cdot}25$, two-step scheme, $\vec{\beta}$ linear)

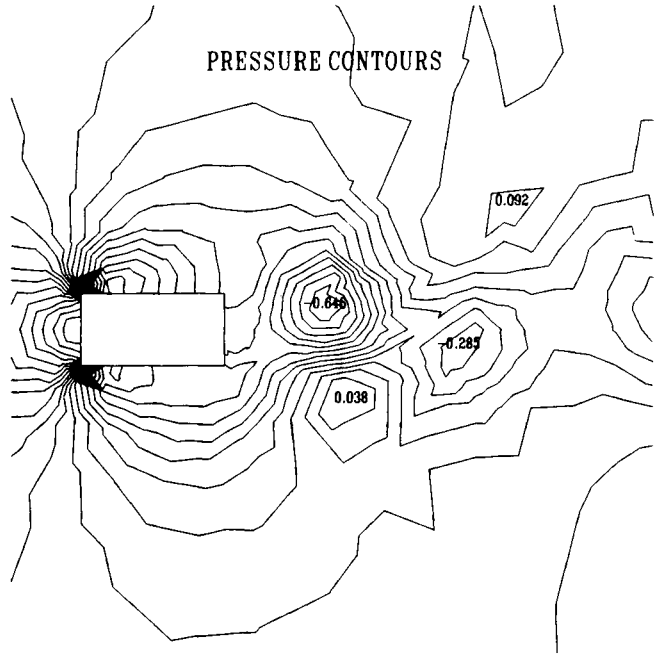

Figure 12. Same as Figure 11 but for pressure contours

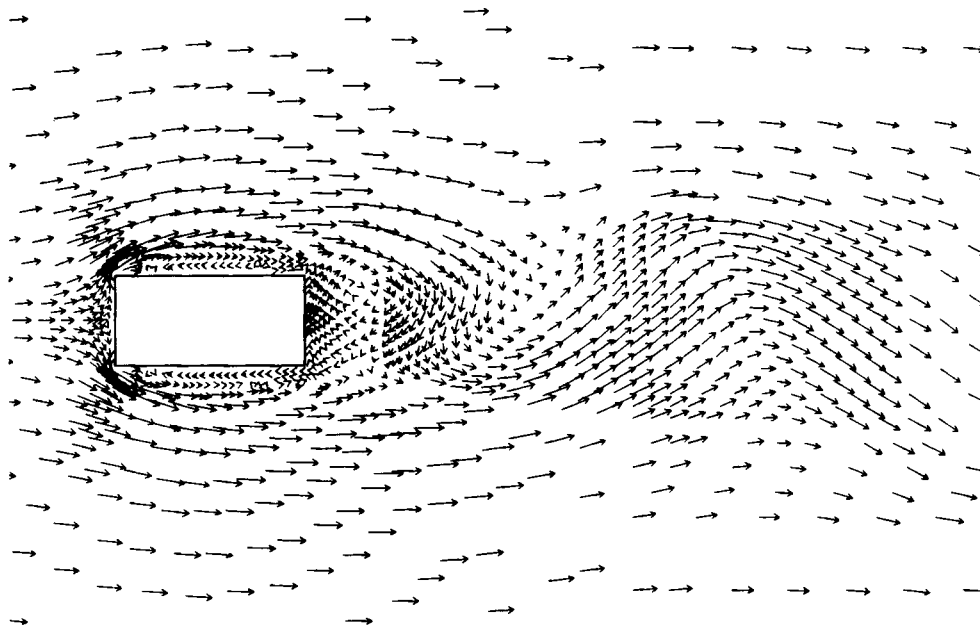

Figure 13. Same as Figure 11 but for nodal velocities

For completeness we also include instantaneous plots of streamlines (Figure 11), pressure contours (Figure 12) and nodal velocities (Figure 13) as obtained with the last computation.

6. CONCLUSIONS

We have presented a Lagrange–Galerkin-based method to solve the unsteady Navier–Stokes equations. Several difficulties that appear in the computation of the right-hand side have been given satisfactory solution, in such a way that non-trivial problems can be handled on small computers (most of the examples were run on VAX 11-780 and MICROVAX II models). To allow for comparison, we report a typical CPU time of 6 s per time step on a CRAY XMP-24 for mesh RECT2 (3459 unknowns).

An analysis of the effect of the time step led us to propose a two-step scheme with second-order approximation for the derivative along path lines. This modification appears to be effective in steady problems. Its extension to transient problems requires some additional care, since accurate particle tracking seems to dominate the time discretization error. The velocity field should not be frozen at the last computed value as is usually done, but rather approximated linearly in time for our scheme to remain effective.

Our experiments also reveal a mesh-dependent numerical viscosity introduced by the method. This should be kept in mind for high-Reynolds-number simulations to be effective.

ACKNOWLEDGEMENTS

The authors acknowledge the contributions of Jorge Pierini and Marcelo Vénere to some topics of Section 4. Computations on mesh RECT2 were performed on the CRAY XMP-24 of the Alabama Supercomputer Network. Our thanks are also due to one of the unknown reviewers, who detected an error in one of the original figures.

REFERENCES

1. V. Girault and P.-A. Raviart, *Finite Element Methods for Navier–Stokes Equations*, Springer, Berlin/Heidelberg, 1986.
2. O. Pironneau, 'Finite elements for flow problems', in *V Escola de Matemática Aplicada*, Lab. Nac. Comput. Científica, Rio de Janeiro, Brasil, 1985, pp. 309–383.
3. F. Thomasset, *Implementation of Finite Element Methods for Navier–Stokes Equations*, Springer, New York/Heidelberg/Berlin, 1981.
4. J. P. Benqué, G. Labadie and J. Ronat, in T. Kawai (ed.), *Finite Element Flow Analysis*, North-Holland, Amsterdam, 1982, pp. 295–302.
5. M. Bercovier and O. Pironneau, in T. Kawai (ed.), *Finite Element Flow Analysis*, North-Holland, Amsterdam, 1982, pp. 67–74.
6. J. Douglas Jr. and T. F. Russell, *SIAM J. Numer. Anal.*, **19**, 871–885 (1982).
7. K. W. Morton, A. Priestley and E. Süli, *Math Modell. Numer. Anal.*, **22**, 625–653 (1988).
8. J. Cahouet and J.-P. Chabard, *Int. j. numer. methods fluids*, **8**, 869–895 (1988).
9. V. Akman, W. R. Franklin, M. Kankanhalli and C. Narayanaswami, *Comput. Aided Des.*, **21**, 410–420 (1989).
10. S. Pissanetzky and F. G. Basombrío, *Int. j. numer. methods eng.*, **17**, 231–237 (1981).
11. R. A. Finkel and J. L. Bentley, *Acta Inform.*, **4**, 1–9 (1974).
12. A. Bermúdez, J. Durany, M. Posse and C. Vázquez, *Int. j. numer. methods eng.*, **28**, 2021–2039 (1989).
13. A. Bermúdez and J. Durany, *Math. Modell. Numer. Anal.*, **21**, 7–26 (1987).
14. A. Okajima, *J. Fluid Mech.*, **123**, 379–398 (1982).
15. Y. Yoshida and T. Nomura, *Int. j. numer. methods fluids*, **5**, 873–890 (1985)