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

Gustavo C. Buscaglia
Enzo A. Dari
División Mecánica Computacional, DIA
Centro Atómico Bariloche
8400 S.C. de Bariloche - Argentina

RESUMEN

En este trabajo se discuten aspectos que conciernen a la implementación del método de Lagrange-Galerkin para la resolución por elementos finitos de las ecuaciones de Navier-Stokes. Se desarrolla un esquema que permite tratar redes generales no estructuras, en el que la búsqueda geométrica utiliza un algoritmo basado en quad-trees. Se analizan las dificultades que aparecen en la construcción del segundo miembro y se proponen soluciones.

Se incluyen asimismo resultados numéricos para los problemas de cavidad cuadrada y desprendimiento de vórtices detrás de un obstáculo cuadrado, que muestran buen acuerdo con trabajos previos.

ABSTRACT

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 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 from a square cylinder problem, with satisfactory agreement with previous works.

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 stiffness 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 Ladyshenskaya-Babuska-Brezzi (LBB) condition (see, e.g., [8,9].

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 in very fine meshes, and a certain amount of 'upwinding' is needed to make realistic problems tractable.

We have adopted the Lagrange-Galerkin method (LGM) [1,5,6] as a means of introducing this necessary upwinding. 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. 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.

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. Finally, we show several numerical tests in Section 5, to assess the global behaviour of the proposed scheme. Conclusions are drawn in Section 6.

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

$$\rho\left[\frac{\partial \phi_i}{\partial t} + \phi_j \frac{\partial \phi_i}{\partial x_j}\right] - \frac{\partial}{\partial x_j}\left[\mu\left[\frac{\partial \phi_i}{\partial x_j} + \frac{\partial \phi_j}{\partial x_i}\right]\right] + \frac{\partial p}{\partial x_i} = f_i \qquad (1)$$

$$\frac{\partial \phi_i}{\partial x_i} = 0 \qquad (2)$$

where $\vec{\phi}$ is the velocity field, p the pressure, and $\vec{f}$ the body forces. Equation (1) remains valid even if μ varies throughout the domain, as

is the case if it depends on the temperature or some other field. Appropriate conditions to solve (1)-(2) inside a bounded domain Ω are, for example, to specify the initial velocity field

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

and boundary data such as velocities or surface tractions on all of 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, that is

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

where $\underset{\sim}{x}$ stands for the position at time t' of the particle that passes through x at time t. The LGM [1,5,6] is based upon a natural approximation of Equation (4): If we choose a time step Δt and define

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

where t_n belongs to the chosen time discretization, we can approximate the acceleration terms by

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

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

$$\frac{\rho}{\Delta t}\,\phi_i^n(x) - \frac{\partial}{\partial x_j}\left[\mu\left[\frac{\partial \phi_i^n}{\partial x_j} + \frac{\partial \phi_j^n}{\partial \theta_i}\right]\right] + \frac{\partial p^n}{\partial x_i}(x) = f_i^n(x) + \frac{\rho}{\Delta t}\,\phi_i^{n-1}(\underset{\sim}{x}) \tag{7}$$

$$\frac{\partial \phi_i^n}{\partial x_i} = 0 \tag{8}$$

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

FINITE ELEMENT APPROXIMATION

Equations (7)-(8), at each time t_n, give rise to a generalized Stokes problem which can receive standard finite element treatment. As said in the introduction, we chose triangular elements for all fields, so as to work with unstructured meshes. To be more specific, we implemented the 4xP1/P1 element [8,9], consisting of four equal linear conforming sub-triangles inside the (also linear and

conforming) pressure element (see Fig. 1).

We now perform the usual Galerkin-finite element weighting of Equations (7)-(8), obtaining the linear system that corresponds to the so called 'direct' LGM (we have adopted the nomenclature of [6]):

$$\underline{\underline{K}}_{vv}\,\underline{V}^n - \underline{\underline{K}}_{vp}\,\underline{P}^n = \underline{B}^n \tag{9}$$

$$\underline{\underline{K}}^T_{vp}\,\underline{V}^n \tag{10}$$

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

$$\underline{\underline{K}}^{IJ}_{vv} = \frac{\rho}{\Delta t}\int_\Omega N_i^I\,N_i^J\,dx + \int_\Omega \frac{\mu}{2}\left[\frac{\partial N_i^I}{\partial x_j} + \frac{\partial N_j^I}{\partial x_i}\right]\left[\frac{\partial N_i^J}{\partial x_j} + \frac{\partial N_j^J}{\partial x_i}\right]dx \tag{11}$$

$$\underline{\underline{K}}^{IJ}_{vp} = \int_\Omega \frac{\partial N_i^I}{\partial x_i}\,M^J\,dx \tag{12}$$

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

$$\left[\underline{B}^n\right]^I = \frac{\rho}{\Delta t}\int_\Omega \phi_i^{n-1}(\underline{x})N_i^I(x)dx + \rho\int_\Omega f_i^n N_i^I dx + \int_{\partial\Omega} g_i^n N_i^I d\Gamma \tag{13}$$

where $\vec{g}$ is the surface traction. The first integral in (13) is not tractable by analytical means, as $\vec{\phi}^{n-1}(\underline{x})$ is a quite general (far from polyhedral in most cases) continuous function. Also, the position $\underline{x}$ cannot be found exactly, as it is the solution of an ordinary differential equation at each point x of Ω:

$$\frac{dX}{dt}(t) = \vec{\phi}\,(X(t),\,t) \tag{14}$$

with initial condition

$$X(t_n) = x \tag{15}$$

and then $\underline{x}$ comes from

$$\underline{x} = X\left(t_{n-1}\right) \tag{16}$$

In the next section, we describe a numerical scheme to evaluate the first integral in Equation (13) by non-exact integration with approximate solution of Eq. (14).

In what concerns the solution of the linear system (9)-(10), we used a conjugate gradient algorithm on pressure unknowns, with the preconditioner proposed by Cahouet and Chabard [5]

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. [8] 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 4.c). In this way, Equation (14) must be solved only at the finite number of points chosen for quadrature.

Particle tracking

Notice that the velocity field at time t_n is not known at the moment of constructing $\underline{B}^n$. Nevertheless, if Δt is small, trajectories can be accurately determined by fixing $\vec{v}$ at t_{n-1}, solving

$$\frac{dX}{dt} = \vec{v}^{n-1}(X(t)); \; X(t_n) = x \tag{17}$$

and setting

$$\underset{\sim}{x} = X\left[t_{n-1}\right] \tag{18}$$

After comparison with other possibilities (explicit Euler, Runge-Kutta), a second-order predictor-corrector method was chosen, giving

$$x^* = x - \frac{\Delta t}{2} \vec{v}^{n-1}(x) \tag{19}$$

$$\underset{\sim}{x} = x - \Delta t \; \vec{v}^{n-1}(x^*) \tag{20}$$

To construct $\underline{B}^n$, this procedure implies thee valuation of the field $\vec{v}^{n-1}$ at two positions (x^* and $\underset{\sim}{x}$) for each quadrature point. This operation involves interpolation within the elements where x^* (or $\underset{\sim}{x}$) lies, which is not known, and requires special attention (see 4.b.). If the point lies OUTSIDE the domain, it may be due to two situations:

- x is near an inflow boundary: Generally, at inflow boundaries, velocities are prescribed. In this case, we assign to $\vec{v}^{n-1}(\underset{\sim}{x})$ the prescribed value at the point where the particle entered the domain.

- Numerical errors in solving Equation (17): The numerical scheme (19)-(20) can lead to x^* or $\underset{\sim}{x}$ spuriously falling outside the domain. To determine if this is the case, every time a point falls outside Ω, the time step is subdivided (up to 1/16 of its original value) and several predictor-corrector substeps applied to more accurately find $\underset{\sim}{x}$ (see Fig. 2). In this way, spurious exits are removed near corners with no need of

correcting the global Δt (this procedure is also discussed in Section 5 and called 'substepping', the only difference being that in that case it is applied all over the mesh).

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, with 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 (naif):

```
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 [1,11]:

1. Pre-processing step (outside the temporal loop):
   - Construct an auxiliary mesh formed by rectangles and store the locations $X_i(i=1,NX)$, $Y_i(i=1),NY)$ 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 Fig. 3).
   - For every R_{ij}, construct a table T_{ij}, containing the indexes of the original mesh elements that intersect R_{ij}.
2. Search step (inside the temporal loop):
   - Find the rectangle R_{ij} that contains X (this is easily done as the auxiliary mesh is structured).
   - Sweep table T_{ij} to find the element K of the original mesh that contains X.

- ALGORITHM QT (quad-tree structure [7]): This is a variant of the previous algorithm, replacing the auxiliary rectangular mesh by a quad-tree. The quad-tree leaves being automatically subdivided to account for local densifications of the original mesh.

By inspection of the algorithms, Alg. NA is clearly seen to be of highest order. In fact, as at least one search must be performed for each quadrate point, Alg NA is $O(N^2)$. This order soon renders it prohibitive with increasing the number of degrees of freedom. Alg SAM and Alg. QT are similar, but QT handles more efficiently meshes with local refinement, and is to be prefered. 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 Fig. 4, we show the resulting auxiliary structure for the vortex-shedding experiment to be described in Section 5. With this method, geometric search becomes inexpensive when compared to the solution of the linear system.

Numerical quadrature

We performed the first integral in Eq. (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 more simple problem of advection-diffusion of a scalar quantity. We summarize the possible choices in Table 1.

Table 1: Quadrature formulae

ℓ	POINTS	TRIANGULAR COORDINATES	MULTIPLICITY	WEIGHTS	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	(.81685,.09157,.09157)	3	.1099517	4
		(.10810,.44595,.44595)	3	.2233816	

An analysis of the effect of these quadratures on LGM has been reported by Bermúdez et al [4], for steady-state (non-linear) problems. They find that all the formulae in Table 1 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.

The above mentioned tests for the pure transport of a cone also showed that formula 4 is not worthy, as it is much 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 [3,10], and implies the existence of an optimal time step of order h. However, this deleterious effect of very small time steps was only noticeable in transient problems.

NUMERICAL EXAMPLES

In this section, we discuss the performance of the above mentioned method on two well-known test examples: The lid-driven square cavity (see [12] for a comparison of several finite element methods) and the vortex shedding behind a square cylinder (see [9] for experimental data, [13] for numerical predictions). We include an analysis of the effect of the time step on the results.

The lid-driven square cavity

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

$$\phi_1(x_1, x_2=1) = 1$$

$$\phi_1(x_1=0, x_2)=\phi_1(x_1, x_2=0)=\phi_1(x_1=1, x_2)=0$$

$$\phi_2(x_1=0, x_2)=\phi_2(x_1=1, x_2)=\phi_2(x_1, x_2=0)=\phi_2(x_1, x_2=1)=0$$

for a fluid with unit density. The Reynolds number (Re) of this flow is defined as

$$Re = \frac{1}{\mu}$$

We solved this problem for Re = 100 and 400, with the mesh shown in Fig. 5 (1003 unknown) and several time steps. In Figures 6 and 7, we plot the profiles of horizontal velocity along the vertical centre line, for both Re, as compared with a reference computation made on a finer mesh consisting of 2660 degrees of freedom. Also included are the results at Re = 0 (Stokesian flow), to allow for comparison. Clearly, the steady numerical results depend on the choice of the time step. This is not surprising, since the evaluation of inertia is made through Eq. 6 above, and is only exact in the limit of vanishing Δt (cf. Eq. (4)). However, the error in this approximation may come from two different sources: Eq. (6), even if x is traced-back exactly, introduces an error of order Δt. On the other hand, the time step is also involved in the determination of $\underset{\sim}{x}$, as can be seen from Eqs. (19) and (20). Notice that, when obtaining steady-states, Eq. 17 implies no additional approximation as $\vec{\phi}^{n-1}$ equals $\vec{\phi}^{n}$ for large n. To see which of the previous reasons is the origin of the sensitivity to temporal discretization shown if Figs. 6 and 7, we solved Eq. (17) by successive application of (19)-(20) NP times, with auxiliary time step $\Delta t' = \Delta t/NP$. This procedure, which we will call substepping, brought no noticeable improvement in the results even with NP=8. From this we conclude that the first order approximation of the material derivative in Eq. (6) is responsible for the time step dependence of the steady velocity field. This observation provides a means of estimating the adequate time step for simulation of steady phenomena. In fact, the error introduced in (6) can be calculated locally at each quadrature point simply by comparing the accelerations obtained with the proposed Δt and a suitable small fraction of it that properly approximates the limit of Eq. (4).

But probably the most important consequence of the observations reported in the previous paragraph is the following: As the particle-tracking shows unexpensive with the quad-tree-based algorithm, the method would be improved by replacing the first-order scheme of Eq. (6) by the second-order accurate two-step method

$$\vec{a}(x) \approx \frac{3\vec{\phi}^{n}(x) - 4\vec{\phi}^{n-1}(\underset{\sim}{x}) + \vec{\phi}^{n-2}(\underset{\approx}{x})}{2\Delta t} \qquad (21)$$

where the points $\underline{x}$ and $\underline{\bar{x}}$ are the locations, at times t_{n-1} and t_{n-2} respectively, of the particle that is at x at time t_n. We tested this alternative, which we will call 'two-step scheme' in the following, and confirmed the above conjecture. In Figure 8, we compare the profiles obtained, with $\Delta t = 0.1$ at Re = 100, for the one-step and two-step schemes. The reference curve corresponds to a finer mesh as above. The same is done, at Re=400, in Figure 9. Clearly, bigger time steps are allowed when using Eq.(21), and the computing times reduce consequently. For all results at Re=400 we used the substepping procedure, with MP = 2.

Vortex shedding behind a square cylinder

We consider here the two-dimensional flow past a square obstacle, of a fluid with freestream velocity U, viscosity μ and unit density. The Reynolds number is defined as

$$Re = \frac{R\ell}{\mu}$$

where ℓ is the width of the obstacle. For our experiments, we took β=1 and Re=100 and 400. The finite element mesh and dimensions of the computational domain are shown in Figure 10. Adherence conditions are imposed at the obstacle, the exit is traction-free, the fluid enters the domain with uniform velocity U (horizontal), and symmetry is assumed at the top and bottom boundaries. No mass forces are present, and also no perturbation of the boundary conditions is needed to obtain an oscillatory flow, with vortices periodically leaving the obstacle. Both experimental [9] and numerical [13] results are available in the literature for this problem. A typical dimensionless value for comparison is the Strouhal number, defined by

$$St = \frac{f\ell}{U}$$

where f is the frequency of the shedding. With our method, we obtained St = 0.13 at Re = 100 and St = 0.14 at Re = 400, in reasonable agreement with reported values. The flow pattern (see instantaneous streamlines around the obstacle in Fig. 11 for Re = 400) and pressure contours (see Fig. 12) also resemble published results.

We now turn to a study of the effect of the time step. In Table 2, we show the Strouhal numbers obtained with several Δt. Clearly, the shedding frequency is sensitive to temporal discretization. Substepping only modifies the results with $\Delta t = 0.5$.

Table 2: Effect of the time step on the Strouhal number for the vortex-shedding experiment.

Re	STEP=0.05	STEP=0.20	STEP=0.5	STEP=0.5(MP=4)
100	0.130	0.130	0.116	0.113
400	0.145	0.142	0.124	0.123

However, as mentioned in 4.c, smaller time steps with fixed mesh enhance numerical diffusion for transient problems, and probably the results obtained with $\Delta t = 0.05$ are no better than those with $\Delta t = 0.2$. The small effect of substepping suggests that, for this problem also, errors in solving Eq.(17) are unimportant. We thus conjecture that the two-step scheme (21) could be useful. Further tests are in progress to verify this assertion.

CONCLUSIONS

We have presented a Lagrange-Galerkin based method to solve the transient 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 relatively small computers (we used VAX 11-780 and MICROVAX II). Numerical results have been shown to assess the numerical accuracy of the method.

An analysis of the effect of the time step led us to propose a two-step approximation for the derivative along characteristics. This modification, when applied to the lid-driven cavity problem, gave encouraging results and further tests are under way.

Acknowledgements: The authors acknowledge the contributions of Jorge Pierini and Marcelo Vénere to some topics of Section 4.

REFERENCES

[1] V. Akman, W.R. Franklin, M. Kankanhalli and C. Narayanaswami; Comp. Aided Des. 21, No.7, 1989.
[2] J.P. Benqué, G. Labadie and J. Ronat, in T. Kawai (Ed.), "Finite Element Flow Analysis", North Holland, 1982, pp.295-302.
[3] A. Bermudez and J. Durany, Math. Modell. and Numer. Anal. 21, 7-26 (1987).
[4] A. Bermúdez, J. Durany, M. Posse and C. Vázquez, Int. J. Num. Meth. Engn. 28, 2021-2039 (1989).
[5] J. Cahouet and J.-P. Chabard, Int. J. Num. Meth. in Fluids 8, 869-895 (1988).
[6] J. Douglas Jr., and T.F. Russell, SIAM J. Numer. Anal. 19, 871-885 (1982)
[7] R.A. Finkel and J.L. Bentley, Acta Informatica 4, 1-9 (1974).
[8] K.W. Morton, A. Priestley and E. Süli, Math Modell. and Numer. Anal. 22, 625-653 (1988).
[9] A. Okajima, J. Fluid Mech. 123, 379-398 (1982).
[10] O. Pironneau, 'Finite Elements for Flow Problems', in V Escola de Matematica Aplicada, Lab. Nac. de Computacao Cientifica, Rio de Janeiro, July 1985.
[11] S. Pissanetzky and F.G. Basombrio, Int. J. Num. Meth. Engng. 17, 231-237 (1981).
[12] F. Thomasset, "Implementation of Finite Element Methods for Navier-Stokes Equations", Springer-Verlag, New York - Heidelberg - Berlin, 1981.
[13] Y. Yoshida and T. Nomura, Int. J. Num. Meth. in Fluids 5, 873-890 (1985).

FIGURE CAPTIONS

Fig. 1: $4\times P_1/P_1$ mixed element.

Fig. 2: Schematic diagram showing how substepping prevents x^* from spuriously falling outside Ω.
Fig. 3: An example of the structured auxiliary mesh.
Fig. 4: Auxiliary structure automatically generated in the vortex-shedding experiment.
Fig. 5: Finite element mesh for the lid-driven square cavity problem.
Fig. 6: Profiles of horizontal velocity along vertical centre line at Re = 100. Effect of the time step.
Fig. 7: Idem Fig. 6 at Re = 400.
Fig. 8: Comparison of 1-step and 2-step schemes at Re = 100.
Fig. 9: Idem Fig. 8 at Re = 400.
Fig. 10: Finite element mesh and dimensions for the vortex-shedding example.
Fig. 11: A plot of instantaneous streamlines near the obstacle at Re = 400.
Fig. 12: Instantaneous pressure contours at Re = 400.

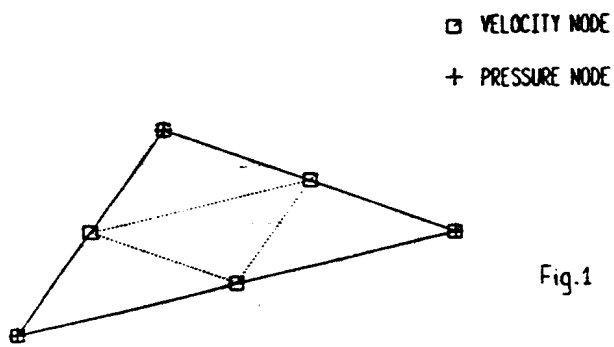


Fig.1

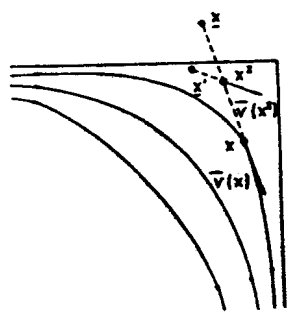


Fig.2

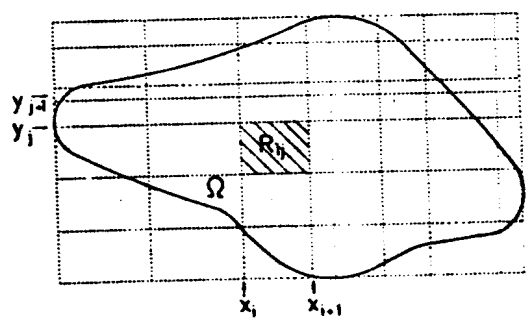


Fig.3

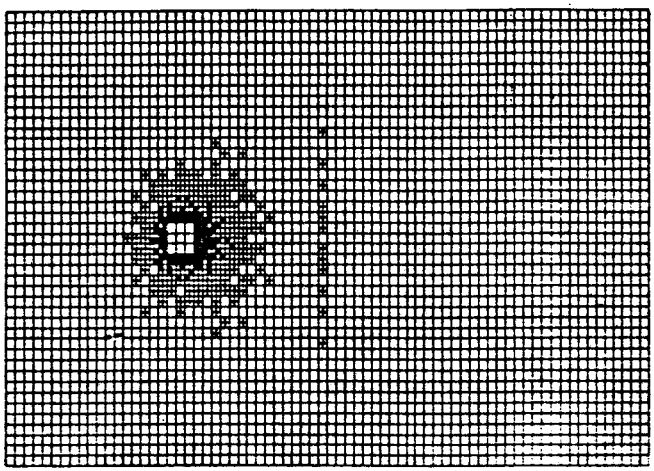


Fig.4

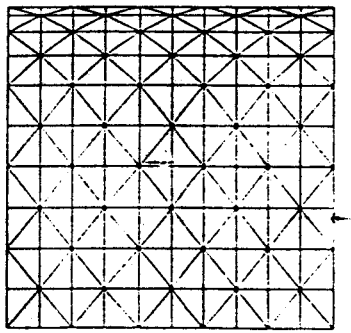
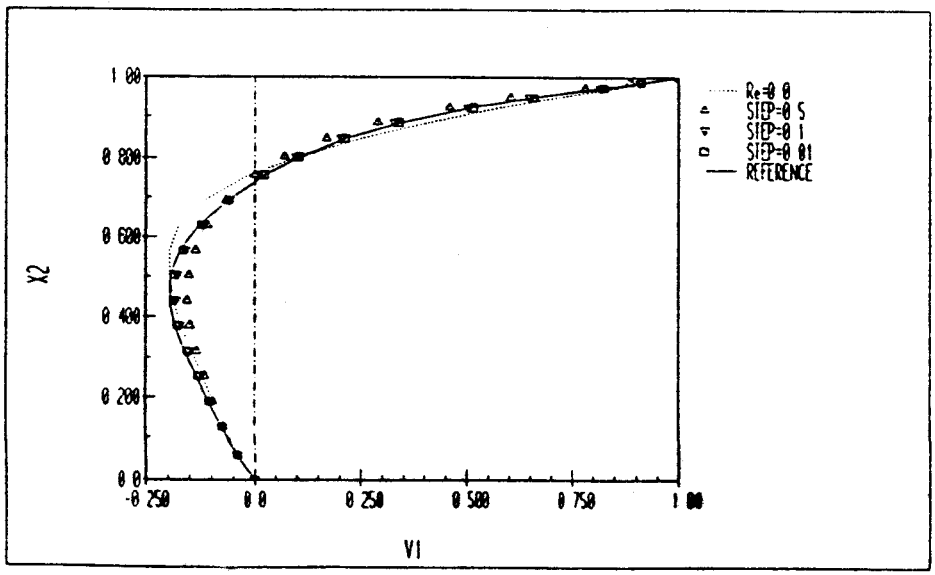


Fig.5

Fig.6



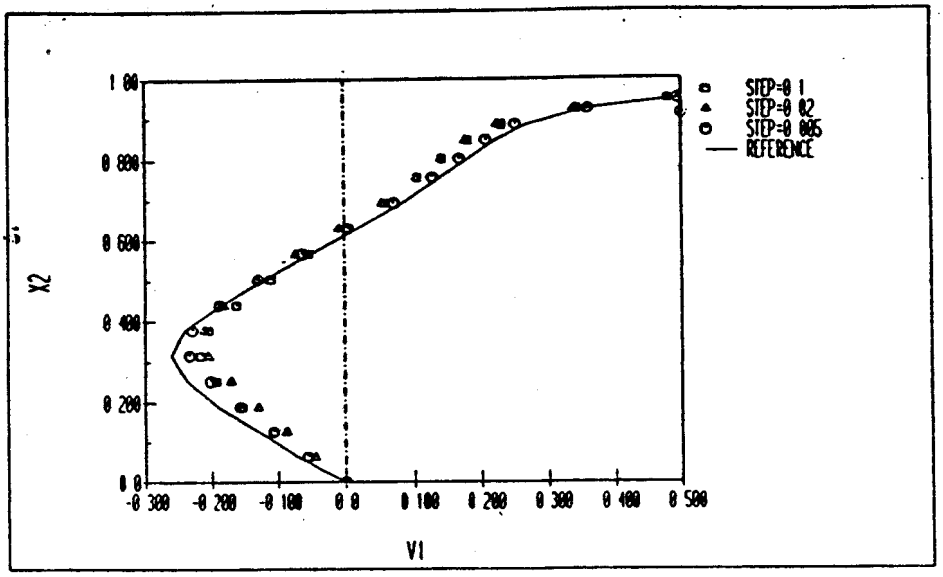
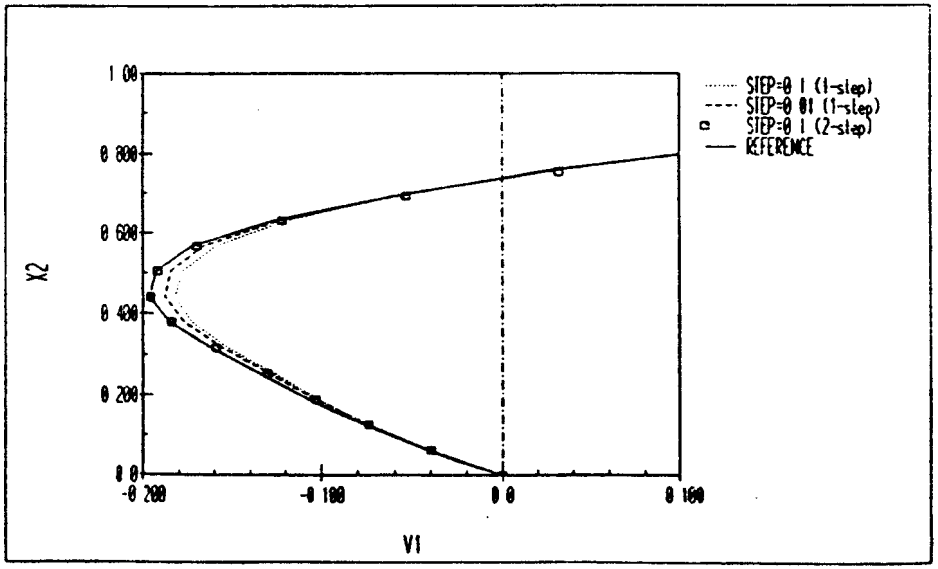

STEP=0 1
STEP=0 02
STEP=0 005
REFERENCE
X2
V1
1 00
0 800
0 600
0 400
0 200
0 0
-0 300
-0 200
-0 100
0 0
0 100
0 200
0 300
0 400
0 500

Fig.7

Fig.8


STEP=0 1 (1-step)
STEP=0 01 (1-step)
STEP=0 1 (2-step)
REFERENCE
X2
V1
1 00
0 800
0 600
0 400
0 200
0 0
-0 200
-0 100
0 0
0 100

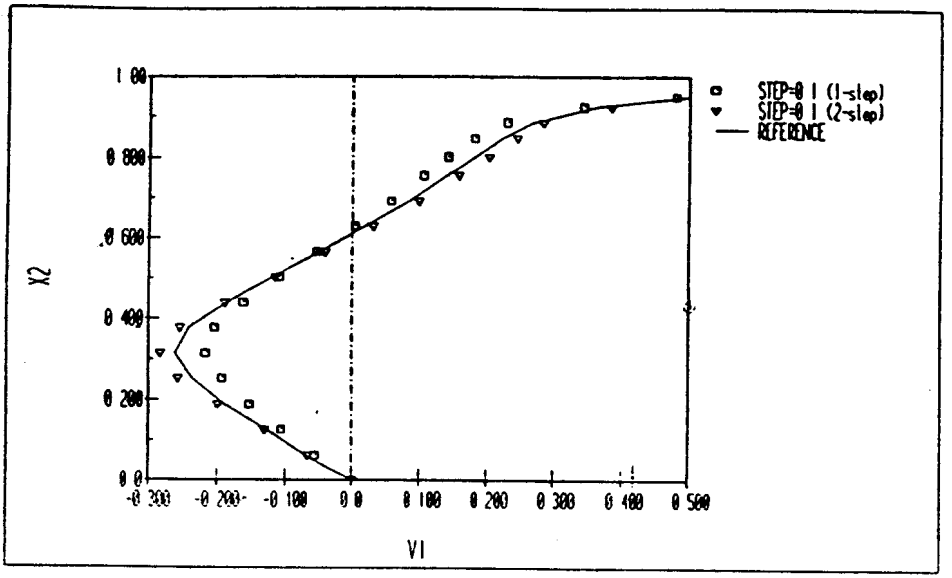

STEP=0 1 (1-step)
STEP=0 1 (2-step)
REFERENCE
X2
V1
1 00
0 800
0 600
0 400
0 200
0 0
-0 300
-0 200
-0 100
0 0
0 100
0 200
0 300
0 400
0 500

Fig. 9

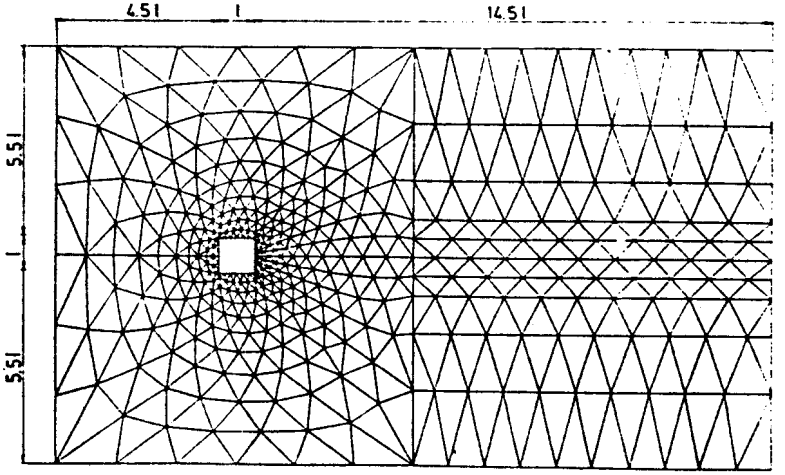

4.5 I
1
14.5 I
5.5 I
1
5.5 I

Fig. 10

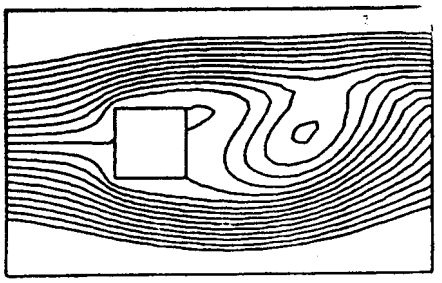


Fig. 11

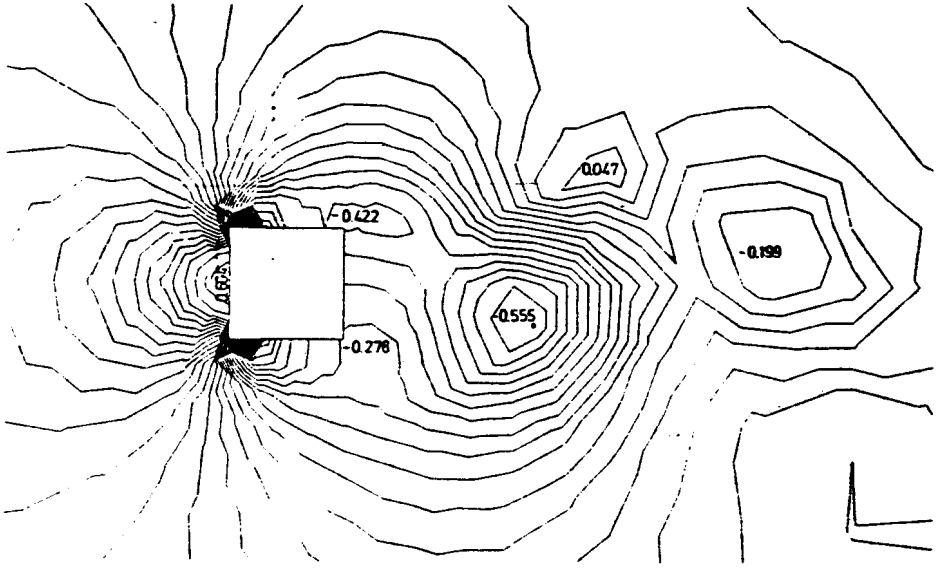


Fig. 12