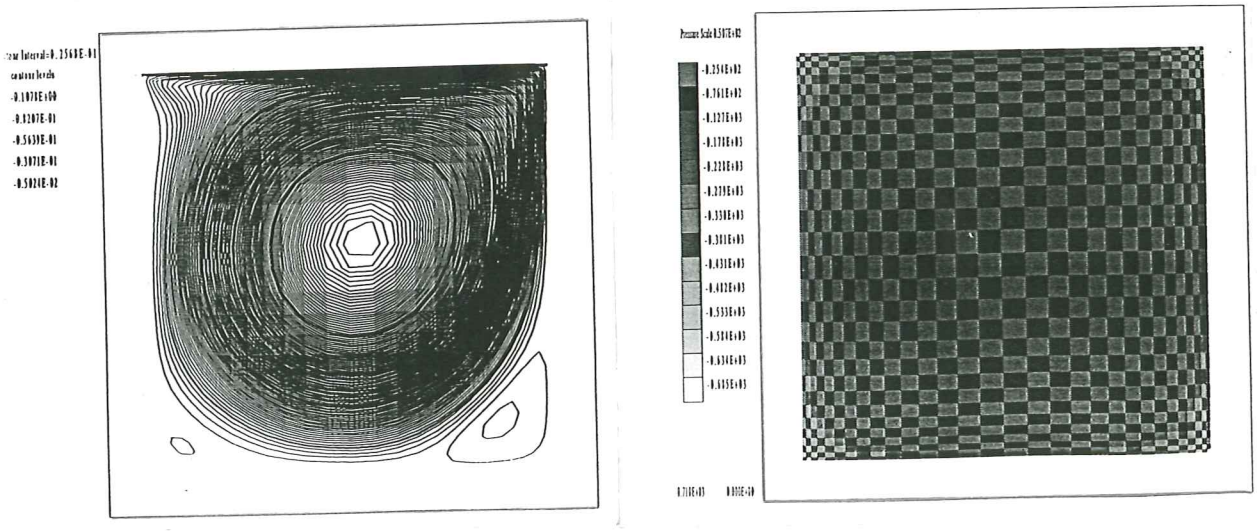
A Fractional-Step Method for the Incompressible Navier-Stokes Equations Related to a Predictor -Multicorrector Algorithm

J. Blasco

R. Codina

A. Huerta



Publication CIMNE Nº 63, March 1995

A Fractional-Step Method for the Incompressible Navier-Stokes Equations Related to a Predictor -Multicorrector Algorithm

J. Blasco
R. Codina
A. Huerta

Publication CIMNE Nº 63, March 1995

International Center for Numerical Methods in Engineering
Gran Capitán s/n, 08034 Barcelona, Spain

ABSTRACT

An implicit fractional–step method for the numerical solution of the time–dependent incompressible Navier–Stokes equations in primitive variables is developed and studied in this paper. The method, which is first order accurate in the time–step, is shown to converge to an exact solution of the equations. By adequately splitting the viscous term, it allows to enforce full Dirichlet boundary conditions on the velocity in all substeps of the scheme, while needing no boundary condition at all for the pressure. It is also shown to be related to an iterative predictor–multicorrector algorithm for evolution equations, when this is applied to the incompressible Navier–Stokes system. A new derivation of the algorithm in a general setting is provided. Two different finite element interpolations are considered for the implementation of the algorithm; numerical results obtained with them for standard benchmark cases are presented.

KEY WORDS: Incompressible Navier–Stokes equations Finite elements Fractional–step methods Predictor–multicorrector algorithm Convergence analysis

1. INTRODUCTION

Since the origin of Computational Fluid Dynamics in the 60's, many numerical schemes have been developed for the approximation of viscous, incompressible flow equations. Among them, several fractional–step projection methods can be found. This category, originated independently by Chorin ([5]) and Temam ([26]), comprises methods developed under different ideas: fractional–step or splitting methods for evolution equations (see [29] for a comprehensive study of them); methods based on a projection onto a space of solenoidal vector fields (see [27]), and others, such as pressure–correction or velocity–correction methods (see [15], for instance), or recently even approximate matrix factorization methods (see [2], [7]).

A typical derivation of a fractional–step method from the original unsteady incompressible Navier–Stokes equations can proceed in two different ways. On the one hand, a time discretization can be performed first, followed by a space discretization. When this approach is adopted, a controversy arises about what boundary conditions are to be imposed at each step, so that the intermediate semidiscrete problems are well–posed. In particular, in most projection methods only the normal component of the velocity boundary condition is imposed in the incompressibility step (see [23]). This fact, together with the need to impose unphysical boundary conditions on the pressure, can generate a numerical boundary layer, so that fine meshes need to be used near the boundary (see [10], [11], [21] and [28]). On the other hand, when a space discretization is performed prior to a fractional–step time discretization of the resulting system of ordinary differential equations, boundary conditions are fixed from the start. However, proceeding this way results in a loss of generality.

Fractional–step projection methods have been used together with different space discretizations: finite difference ([1], [5], [12], [13], [18]), finite element ([6], [10], [15], [17], [20]) and spectral methods ([24]). One important feature of each fractional–step method is the order of the overall scheme with respect to the time discretization. Most methods are first order accurate, but some second order methods have also been developed ([1],

[12]).

An implicit fractional–step method is developed in this paper. A time–discretized form of the equations is presented first, which is shown to be well-posed. Full velocity boundary conditions are imposed in both phases of the scheme, and the pressure does not require of boundary conditions at all. The computed pressure has the meaning of a physical pressure. Besides, discontinuous pressure approximations are allowed.

Convergence of the method to a solution of the continuous problem when the time step tends to zero is proved. Both the intermediate and the end–of–step velocities are shown to converge in the space $H_0^1(\Omega)$, whereas in the original method of Temam the final velocities only converge in $L^2(\Omega)$. The overall algorithm is first order accurate in time for the velocity solution. As will be shown numerically, it is also first order accurate for the pressure.

Although it may look theoretically attractive, straight implementation of the method leads to a scheme in which the second substep of each time–step is equivalent to a Stokes problem, highly undesirable to solve. Based on the method just introduced, a simpler iterative scheme is developed, which is shown to be equivalent to a predictor–multicorrector algorithm designed by T. J. R. Hughes and coworkers (see [4]) and extensively used in the 80's, especially in the context of fluid–structure interaction (see [19]). The present method can actually be understood as a predictor–corrector form of the algorithm, in which only one iteration per time–step is performed.

The classical derivation of the predictor–multicorrector algorithm consists of discretizing the Navier–Stokes equations in space by a finite element interpolation, and then applying a time advancement scheme to the resulting system of ordinary differential equations. A nonlinear system of equations needs being solved at every time-step, and this is achieved by an iterative process. Each iteration of the scheme is classically decomposed into two phases; here it will be shown that this decomposition is performed in a fractional–step way.

As in the above mentioned references, some simplifications are introduced in the

algorithm, which make it a simple and efficient numerical scheme. The effect of these simplifications is cancelled by the iteration procedure.

The paper is organized as follows: Section 2 introduces the problem to solve, the functional setting for it and the notation used in this paper. In Section 3 the fractional–step method, which depends on a free parameter $\theta > 0$, is studied and compared to other existing fractional–step methods. An improvement of the proposed scheme is also considered. Convergence of the method for the fully implicit case $\theta = 1$ is stated in Section 4. The formal derivation of the predictor–multicorrector algorithm as a fractional–step iterative method is given in Section 5. Finally, some numerical results obtained with this algorithm are shown in Section 6, which confirm its accuracy properties and ability to solve both steady and unsteady problems.

2. PROBLEM STATEMENT AND NOTATION

The evolution of viscous, incompressible fluid flow in a bounded domain $\Omega \subset \mathbb{R}^d$ ($d = 2,3$) is governed, in the primitive variable formulation, by the unsteady, incompressible Navier–Stokes equations:

$$\frac{\partial u}{\partial t} + (u \cdot \nabla)u + \nabla p - \nu \nabla^2 u = f \tag{2.1a}$$

$$\operatorname{div} u = 0 \tag{2.1b}$$

on $Q = \Omega \times (0,T)$, where $u(x,t) \in \mathbb{R}^d$ is the fluid velocity at position $x \in \Omega$ and time $t \in (0,T)$ (with $T > 0$ given), $p(x,t) \in \mathbb{R}$ is the fluid kinematic pressure, $\nu > 0$ is the kinematic viscosity, $f(x,t)$ is an external force, ∇ is the gradient operator and ∇^2 is the Laplacian operator.

Equation (2.1a) is formally equivalent to its dimensionless form, provided $\nu = 1/\mathrm{Re}$, Re being the fluid's Reynolds number. Boundary conditions have to be given to complete the equation system (2.1). For the sake of simplicity, only homogeneous Dirichlet type boundary conditions are considered:

$$u(x,t) = 0, \quad (x,t) \in \Gamma \times (0,T) \tag{2.2}$$

where the boundary $\Gamma = \partial\Omega$ is assumed to be sufficiently smooth (see [27] for a precise definition). However, this study can be extended to more general boundary conditions.

An initial condition must also be specified for the velocity:

$$u(x,0) = u^0(x), \quad x \in \Omega \tag{2.3}$$

whereas no boundary or initial conditions need be specified for the pressure.

The study of the above equations of motion requires the following Hilbert spaces (see, for instance, [9]): $L^2(\Omega)$, equipped with the usual scalar product (u,v) and norm $|u| = (u,u)^{1/2}$; the quotient space $L^2_0(\Omega) = L^2(\Omega)/\mathbb{R}$, in the case of Dirichlet type boundary conditions only, since the pressure term p in equation (2.1a) is then determined up to an additive constant; the space $H^1(\Omega)$, whose scalar product and norm are denoted by $(u,v)_1 = (u,v) + (\nabla u, \nabla v)$ and $||u|| = (u,u)_1^{1/2}$, respectively.

The space $H^1(\Omega)$ contains a closed subspace $H^1_0(\Omega)$ made up with functions which vanish at the boundary of Ω. The Poincaré–Friedrich inequality:

$$d_0(\nabla u, \nabla u)^{1/2} \geq (u,u)^{1/2}, \qquad \forall u \in H^1_0(\Omega) \tag{2.4}$$

with $d_0 > 0$ a constant, ensures that $||u||_0 = (\nabla u, \nabla v)^{1/2}$ is a norm on $H^1_0(\Omega)$ equivalent to the norm induced by $H^1(\Omega)$. The scalar product in $H^1_0(\Omega)$ is, then, $((u,v)) = (\nabla u, \nabla v)$. The dual space of $H^1_0(\Omega)$ is denoted by $H^{-1}(\Omega)$; the duality pairing between these spaces is denoted by $< , >$. All these definitions and results carry over to d–dimensional vector valued function spaces.

Due to the incompressibility condition (2.1b), closed subspaces of solenoidal vector fields of these Hilbert spaces are also needed. Thus, in the standard notation, one defines:

$$H = \{u \in (L^2(\Omega))^d \;/\; \exists\, \mathrm{div}\, u = 0 \in L^2(\Omega),\;\; n \cdot u|_\Gamma = 0\} \tag{2.5}$$

$$V = \{u \in (H^1_0(\Omega))^d \;/\; \mathrm{div}\, u = 0 \in L^2(\Omega)\} \tag{2.6}$$

3. FRACTIONAL–STEP METHOD

One way of discretizing equations (2.1) in time is by fractional–step methods, in which the time advancement is decomposed into a sequence of (generally two) steps. The following class of 2–step methods is considered here:

First step

The first step of the method, which includes viscous and convective effects, consists of finding, given $u^n \in V$, an intermediate velocity $u^{n+1/2}$ such that:

$$\frac{u^{n+1/2} - u^n}{\Delta t} - \theta\nu\nabla^2 u^{n+1/2} - (1-\theta)\nu\nabla^2 u^n + (u^n \cdot \nabla)u^{n+1/2} = \overline{f}^n \quad (3.1a)$$

$$u^{n+1/2}|_\Gamma = 0 \quad (3.1b)$$

where $\Delta t > 0$ is the time–step, θ is a parameter such that $0 < \theta \leq 1$ and the superscript n denotes the time level $t_n = n\,\Delta t$. The approximation of the nonlinear term may take other forms; the semi–implicit approximation adopted here is taken from [25]. As for the approximation of the force term, one defines, following [27], $\overline{f}^n = \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} f(t)\,dt$. Equations (3.1) can be written in weak form as: $\tilde{a}_\theta(u^{n+1/2}, v) = l_0(v),\ \forall v \in (H_0^1(\Omega))^d$,

where: $\tilde{a}_\theta(u,v) = (u,v) + \theta\ \Delta t\ \nu\,((u,v)) + \Delta t((u^n \cdot \nabla)u, v)$ is a bilinear, continuous form on $(H_0^1(\Omega))^d$, which is coercive with respect to the norm $||u||_0$ (and hence to $||u||$), and $l_0 \in (H^{-1}(\Omega))^d$ is a known map. Coerciveness of $\tilde{a}$ results from the skew–symmetric character of the approximation of the convective term (which is a consequence of the assumed solenoidal character of u^n and the vanishing of u^n at the boundary) and the presence of the Laplacian term. Existence and uniqueness of $u^{n+1/2}$ is established by the Lax–Milgram theorem.

The fully implicit case $\theta = 1$ can be found in the original method of R.Temam

The Crank–Nicholson case $\theta = 1/2$ is of main importance, since it provides a second order approximation of the viscous term. Examples of such a choice can be found in [1], [10], [12]and [13] (method also used in [24]). The basic difference among these methods is the treatment of the nonlinearity, which is normally second order in time and explicit.

The explicit case $\theta = 0$ has also been considered before (see [6] or [17], for instance). However, the resulting weak problem is not well–posed in $(H_0^1(\Omega))^d$, since the bilinear form $\tilde{a}_0(u, v)$ is not coercive with respect to the norm of $(H_0^1(\Omega))^d$. This case cannot, therefore, be studied under the present ideas.

Second step

As far as the second step of the scheme is concerned, most known methods use the projection idea, which is based on a Hodge decomposition of a given vector field into a solenoidal field with zero normal component on the boundary and the gradient of some scalar function, or more specifically on the Ladyzenskaya theorem (see [16]). The incompatibility of the projection boundary conditions with those of the continuous problem and the need to impose unphysical boundary conditions on the pressure may result in the presence of a numerical boundary layer of size $O(\nu T)^{1/2}$ (see [10] and [21] for further discussion on this subject).

The method proposed herein includes a diffusion term in the incompressibility step, which allows for the imposition of the full boundary conditions for the velocity while needing no boundary condition at all for the pressure. That is, given $u^{n+1/2}$ from equation (3.1), one solves:

$$\frac{u^{n+1} - u^{n+1/2}}{\Delta t} - \theta\nu\nabla^2(u^{n+1} - u^{n+1/2}) + \nabla p^{n+1} = 0 \qquad (3.2a)$$

$$\operatorname{div} u^{n+1} = 0 \qquad (3.2b)$$

$$u^{n+1}|_\Gamma = 0 \qquad (3.2c)$$

Similar ideas to this scheme can be found in the method of [18], developed in a

discrete setting, in the first and third steps of the method of [20] or in the example of a higher order method in Section 7.5 of [23].

The weak form of (3.2) consists of finding $u^{n+1} \in (H_0^1(\Omega))^d$ and $\phi^{n+1} = \Delta t\, p^{n+1} \in L_0^2(\Omega)$ such that:

$$a_\theta(u^{n+1}, v) + b(v, \phi^{n+1}) = l_1(v), \qquad \forall v \in (H_0^1(\Omega))^d \tag{3.3a}$$

$$b(u^{n+1}, q) = 0, \qquad \forall q \in L_0^2(\Omega) \tag{3.3b}$$

where now $a_\theta(u,v) = (u,v) + \theta\, \Delta t\, \nu((u,v))$ is a bilinear, symmetric, continuous form on $(H_0^1(\Omega))^d$, which is also coercive with respect to $||u||_0$; b is a bilinear continuous form defined on $(H_0^1(\Omega))^d \times L_0^2(\Omega)$ by $b(v,q) = -(\mathrm{div}\, v, q)$ and $l_1 \in (H^{-1}(\Omega))^d$ is known. Problem (3.3) is a mixed type problem, in which a_θ is coercive and b satisfies the inf–sup condition or Ladyzenskaya–Babuška–Brezzi (LBB) condition (see [3]):

$$\inf_{q \in L_0^2(\Omega)} \left(\sup_{v \in (H_0^1(\Omega))^d} \frac{b(v,q)}{||v||\, |q|_{L_0^2(\Omega)}} \right) \geq \beta > 0, \tag{3.4}$$

so that existence and uniqueness of u^{n+1} and p^{n+1} follows.

REMARK 1: By adding (3.1a) and (3.2a) one gets:

$$\frac{u^{n+1} - u^n}{\Delta t} - \theta\nu\nabla^2 u^{n+1} - (1-\theta)\nu\nabla^2 u^n + (u^n \cdot \nabla)u^{n+1/2} + \nabla p^{n+1} = \overline{f}^n, \tag{3.5}$$

where the implicit treatment of the viscous term in u^n and u^{n+1}, and not in the intermediate velocity $u^{n+1/2}$, can be observed. Moreover, it is clear from (3.5) that, at least for the linear problem, p^{n+1} keeps its meaning as an end–of–step pressure. One

advantage of using a split scheme like (3.1)–(3.2) rather than (3.5) is the decoupling of the convective effects from incompressibility, which allows the use of suitable approximations for each term. For instance, one may also use a second–order, explicit Adams–Bashforth scheme for the nonlinear term in (3.1), as is costumary practice among some users of fractional steps methods (see [2], [13], [21] or [24]), and choose $\theta = 1/2$ to maintain second order accuracy.

REMARK 2: As in standard projection methods, a Poisson equation can be derived for the pressure to solve (3.2). In fact, taking the divergence of (3.2a) yields:

$$\Delta t\, \nabla^2 p^{n+1} \;=\; (I - \theta\, \Delta t\, \nu \nabla^2)\mathrm{div}\,(u^{n+1/2}) \;\in H^{-1}(\Omega) \tag{3.6}$$

sufficient smoothness of the functions involved been assumed. But in order that (3.6) and (3.2a) imply (3.2b), the incompressibility condition $\mathrm{div}\, u^{n+1} = 0$ must also be enforced on the boundary (see [14]), as in the original method of [5]. Besides, boundary conditions for p^{n+1} cannot be directly derived, and p^{n+1} is subject to integral conditions (see [22]). Therefore, the original grad–div formulation (3.2) is adopted, which has the advantage of allowing discontinuous pressure approximations and requires of no boundary conditions at all for this variable. One drawback of solving (3.2) is the need for the spatial approximation chosen to satisfy the discrete LBB condition (see [3]), and that velocity and pressure unknowns have to be dealt with at the same time.

REMARK 3: A modified scheme with respect to (3.1)–(3.2) is considered next. A pressure update, rather than the pressure itself, is used in the incompressibility step, so that (3.1a) becomes:

$$\frac{u^{n+1/2} - u^n}{\Delta t} \;-\; \theta\nu\nabla^2 u^{n+1/2} \;-(1-\theta)\nu\nabla^2 u^n \;+\; (u^n\cdot\nabla)u^{n+1/2} \;+\nabla p^n \;=\; \overline{f}^n \tag{3.7}$$

whereas (3.2a) turns into:

$$\frac{u^{n+1} - u^{n+1/2}}{\Delta t} - \theta\nu\nabla^2(u^{n+1} - u^{n+1/2}) + \gamma\nabla(p^{n+1} - p^n) = 0 \qquad (3.8)$$

where $0 < \gamma \leq 1$ is another free parameter. Addition of (3.7) and (3.8) shows the implicit treatment of the pressure term; this allows to employ the same approximation for this term as for the other terms by choosing $\gamma = \theta$, whereas (3.1)–(3.2) is necesarilly a fully implicit scheme with respect to the pressure. This idea can also be found in the methods of [25] with $\gamma = 1$ and [12] with $\gamma = 1/2$. The existence and uniqueness of solutions of (3.7) and (3.8) is established the same way as before.

4. CONVERGENCE OF THE METHOD

The fractional–step method introduced in the previous Section is now shown to converge to a solution of the continuous problem, when the parameter θ is chosen equal to unity. The ideas used here follow a similar developement to the proof of convergence of the original fractional–step method given by Temam ([27]). After introducing some notation, a convergence theorem is given for some sequences of functions obtained from the intermediate and the end–of–step velocities of the fractional–step method. The proof of the theorem, as well as the preliminary results needed for it, are given in Appendix A.

Given $r \in [1, \infty)$, $T > 0$ and a Banach space W, the space $L^r(0,T;W)$ consists of functions defined on $(0,T)$ into W which are strongly r–integrable, i.e., such that:

$$||u||_{L^r(0,T;W)} = (\int_0^T ||u(t)||_W^r \, dt)^{1/r} < \infty \tag{4.1}$$

It is also a Banach space with respect to the norm $||u||_{L^p(0,T;W)}$. The space of essentially bounded functions on $(0,T)$ into W is denoted by $L^\infty(0,T;W)$. When W is a Hilbert space with scalar product $(.,.)_W$, the space $L^2(0,T;W)$ is likewise with respect to:

$$(u,v) = \int_0^T (u(t), v(t))_W \, dt \tag{4.2}$$

Let $T = N\Delta t$ and $k = \Delta t$, and consider the scheme consisting of (3.1) and (3.2) with $\theta = 1$. It can be written in variational form as:

I) Find $u^{n+1/2} \in (H_0^1(\Omega))^d$ such that:

$$(\frac{u^{n+1/2} - u^n}{k}, v) + \nu((u^{n+1/2}, v)) + c(u^n, u^{n+1/2}, v) = < \overline{f}^n, v >, \quad \forall v \in (H_0^1(\Omega))^d \tag{4.3}$$

where the trilinear form $c(u,v,w) = \int_\Omega ((u\cdot\nabla)v)\cdot w\, d\Omega$ is continuous on $[(H_0^1(\Omega))^d]^3$ and, when $\operatorname{div} u = 0$ and $n\cdot u|_\Gamma = 0$ (as is the case for $u = u^n$), it is skew–symmetric in its last two arguments, i.e., it satisfies: $c(u,v,v) = 0,\ \forall v \in (H_0^1(\Omega))^d$.

II) Find $u^{n+1} \in (H_0^1(\Omega))^d$ and $p^{n+1} \in L_0^2(\Omega)$ such that:

$$(\frac{u^{n+1}-u^{n+1/2}}{k}, v) + \nu((u^{n+1}-u^{n+1/2}, v)) + b(v, p^{n+1}) = 0, \forall v \in (H_0^1(\Omega))^d \quad (4.4a)$$

$$b(u^{n+1}, q) = 0, \quad \forall q \in L_0^2(\Omega) \quad (4.4b)$$

Some approximating functions are defined next which will be piecewise equal to $u^{n+1/2}$, u^{n+1} or u^n. A linear interpolation between u^{n+1} and u^n is also introduced:

$$u_k^1 : [0,T] \to (L^2(\Omega))^d \ / \ u_k^1(t) = u^{n+1/2}, \ nk \le t < (n+1)k$$

$$u_k^2 : [0,T] \to (L^2(\Omega))^d \ / \ u_k^2(t) = u^{n+1}, \ nk \le t < (n+1)k$$

$$u_k^3 : [0,T] \to (L^2(\Omega))^d \ / \ u_k^3(t) = u^{n}, \ nk \le t < (n+1)k$$

$u_k : [0,T] \to (L^2(\Omega))^d$ / u_k is continuous, linear on t on each interval $[nk,(n+1)k]$ and $u_k(nk) = u^n$, for $n = 0,\ldots,N$.

The main result of this section is summarized in the following theorem:

Theorem 1: let $d \le 4$, $f \in L^2(0,T;H)$ and $u^0 \in V$.

Then, there exists a subsequence k' of k and a solution u of (2.1) such that:

1) $u_{k'}^i$ and $u_{k'}$ converge to u in $L^2(Q)$ strongly, $i = 1,2,3$.

2) $u^i_{k'}$ and $u_{k'}$ converge to u in $L^\infty(0,T;(L^2(\Omega))^d)$ weak–star, $i = 1,2,3$.

3) $u^i_{k'}$ and $u_{k'}$ converge to u in $L^2(0,T;(H^1_0(\Omega))^d)$ weakly, $i = 1,2,3$.

For any other subsequence k'' such that these convergence results hold, u must be a solution of (2.1).

One also has:

Corolary 1: let $d = 2$.

Then, the convergence given in Theorem 1 is of the sequence as a whole.

In summary, both the intermediate $u^{n+1/2}$ and the end–of–step velocities u^{n+1} are shown to converge to u in $(H^1_0(\Omega))^d$, through the functions u^1_k and u^2_k respectively. This is an improvement with respect to [27], where u^{n+1} only converges in $(L^2(\Omega))^d$.

5. THE PREDICTOR–MULTICORRECTOR ALGORITHM.

The predictor–multicorrector algorithm of [4] can be understood in the context of fractional–step methods. In fact, the method proposed in the previous Sections can actually be seen as a particular case of this algorithm, in which only one correction per step is performed, at least in the fully implicit case $\theta = 1$. First of all, let us approximate the incompressible Navier–Stokes equations (2.1) by an implicit method of the form:

$$\frac{u^{n+1} - u^n}{\Delta t} - \alpha\, a^{n+1} - (1-\alpha)a^n = 0 \qquad (5.1a)$$

$$\mathrm{div}\,(u^{n+1}) = 0 \qquad (5.1b)$$

$$u^{n+1}|_\Gamma = 0 \qquad (5.1c)$$

where:

$$a^m = f^m - \nabla p^m - (u^m \cdot \nabla)u^m + \nu\nabla^2 u^m \qquad (5.2)$$

$f^m = f(m\,\Delta t)$ and $0 < \alpha \leq 1$. An iterative scheme is introduced for the solution of the nonlinear, coupled problem (5.1). It starts with some predictions u_0^{n+1} and p_0^{n+1} for u^{n+1} and p^{n+1}, respectively. Then, each iteration is split into two steps. The first one accounts for the momentum equation but not for the incompressibility condition, which is dealt with in the second step, in a similar way to the scheme of Section 3. The convective term is linearized for simplicity, and the pressure gradient at the previous iteration is used in the first step, which consists of finding, given the i–th iteration approximations u_i^{n+1} and p_i^{n+1} to u^{n+1} and p^{n+1}, an *intermediate iteration* velocity $u_{i+1/2}^{n+1}$ such that:

$$\frac{u_{i+1/2}^{n+1} - u^n}{\Delta t} - \alpha\nu\nabla^2 u_{i+1/2}^{n+1} = \alpha f^{n+1} + (1-\alpha)a^n - \alpha(u_i^{n+1}\cdot\nabla)u_i^{n+1} - \alpha\nabla p_i^{n+1} \qquad (5.3a)$$

$$u^{n+1}_{i+1/2}|_\Gamma = 0 \qquad (5.3b)$$

(The notation $u^{n+1}_{i+1/2}$ has been chosen deliverately to emphasise that the solution of (5.3) is an intermediate iteration aproximation of the velocity at time t_{n+1}.) In the second step of each iteration, a pressure increment is used to enforce incompressibility, in a similar way to the modified scheme of Remark 3, Section 3. Thus, one looks for an *end-of-iteration* velocity u^{n+1}_{i+1} and pressure p^{n+1}_{i+1} such that:

$$\frac{u^{n+1}_{i+1} - u^{n+1}_{i+1/2}}{\Delta t} - \alpha\nu\nabla^2(u^{n+1}_{i+1} - u^{n+1}_{i+1/2}) + \alpha\nabla(p^{n+1}_{i+1} - p^{n+1}_i) = 0 \qquad (5.4a)$$

$$\text{div}\, u^{n+1}_{i+1} = 0 \qquad (5.4b)$$

$$u^{n+1}_{i+1}|_\Gamma = 0 \qquad (5.4c)$$

As is shown in Appendix B, the iterative scheme (5.3)–(5.4) is a generalized version of the predictor–multicorrector algorithm of [4]. In the fully implicit case $\alpha = 1$, the scheme (3.7)–(3.8) of Section 3, with $\theta = 1$, is also equivalent to this algorithm if only one iteration per time step is performed, since then $u^{n+1}_0 = u^n$ and $p^{n+1}_0 = p^n$, (see Appendix B); however, this algorithm employs an explicit treatment of the convective term in that case. One has, then, that $u^{n+1}_{1/2} = u^{n+1/2}$.

The semidiscrete scheme (5.3)–(5.4) is further discretized in space by a Galerkin finite element interpolation of mixed type. Two different element pair combinations are considered in the numerical examples presented next, both of which employ discontinuous pressure approximations. The first one is the well known bilinear–velocity constant–pressure element (Q1P0), which does not satisfy the discrete LBB condition and suffers from the *checkerboarding* phenomenon. The second element used is the biquadratic–velocity linear–pressure element (Q2P1), which satisfies the discrete LBB condition. In what follows, the following standard notation is used: U and P represent nodal velocity and elemental pressure vectors, respectively; M is the mass matrix, K

is the viscous–stiffness matrix, $C(U)$ is the convection matrix and G represents the discrete gradient operator (see [19] for precise definitions of these matrices). The finite element approximation of (5.3) yields the following system of linear equations:

$$B\,(U^{n+1}_{i+1/2}) = F_1 \tag{5.5}$$

where $B = M + \alpha\,\Delta t\,\nu K$ and F_1 is a known vector. On the other hand, (5.4) yields:

$$B\,(U^{n+1}_{i+1}) \;-\; \alpha\,G\,(P^{n+1}_{i+1}) = F_2 \tag{5.6a}$$

$$G^T\,(U^{n+1}_{i+1}) = 0, \tag{5.6b}$$

G^T (transpose of G) being the discrete divergence matrix and F_2 another known vector. By isolating U^{n+1}_{i+1} from (5.6a) (the matrix B being positive definite, and thus invertible) and substituting into (5.6b), one gets:

$$(G^T B^{-1} G)\,(P^{n+1}_{i+1}) \;=\; F_3 \tag{5.7}$$

which, followed by (5.6a), is equivalent to solving (5.6). It is now clear that a system like (5.6) will eventually suffer from the same restrictions as a discretization of the steady Stokes equations, or, more generally, an arbitrary mixed problem: compatibility conditions between the interpolations of velocity and pressure need be satisfied. The matrix $(G^T B^{-1} G)$ is positive semidefinite; but it may have a non–trivial kernel if the mixed interpolation introduces spurious pressure modes, which satisfy: $GP = 0,\ \ P \neq 0$. This is the case of the Q1P0 element when all Dirichlet–type boundary conditions are considered, even when non–uniform meshes are used, as will be seen in the numerical examples.

Upon implementation of the predictor–multicorrector algorithm, the matrix B is usually approximated by M in all its appearances, the difference between the two being of order Δt, which is the expected error of the overall scheme. Moreover, the matrix M is further simplified by a *lumping* process, resulting in a diagonal matrix M^L.

Given the implicit character of the approximation of the viscous and convective terms, no Δt limitations are expected for the stability of the algorithm in a wide range of Reynolds numbers. However, the iterative nature of the scheme and the simplifications introduced in it (such as the linearization of the convective term) impose restrictions on Δt for the stability (and convergence) of the iterative process, as will be seen in the next Section.

6. NUMERICAL RESULTS

The predictor–multicorrector algorithm described in the previous Section was applied to the following three standard flow problems.

6.1 Analytical transient flow

As a numerical check for the accuracy properties of the method, the test case introduced by van Kan (see [12]) was considered. It consists of the Navier–Stokes flow on a unit square cavity in which an inflow velocity profile is prescribed at the top wall defined by $u((x,1),t) = (0, -\sin(\pi(x^3 - 3x^2 + 3x))e^{(1-1/t)})$ for $0 \leq x \leq 1$ and $t > 0$, the bottom and left walls are no slip walls and natural boundary conditions are enforced on the right, outlet wall. As in [12], a Reynolds number of 10 was selected, and the fluid was at rest at the start. A uniform mesh consisting of 6×6 elements was used for the Q1P0 case; in order to compare the results from both interpolations, the same mesh points were used to define a 3×3 mesh for the Q2P1 element. The latter case requires of less pressure degrees of freedom than the former one, thus reducing the computational cost (recall that the pressure system (5.7) is the only one with a consistent matrix, having replaced B with M^L).

Let's denote by $\kappa_i(\Delta t)$ the quotient: $\dfrac{||U_i(\Delta t) - U_i(\frac{1}{2}\Delta t)||}{||U_i(\frac{1}{2}\Delta t) - U_i(\frac{1}{4}\Delta t)||}$, where U_i $(i = 1,2)$ contains the i-th component of the nodal velocities obtained at $t = 1$ with the indicated time–step. Euclidean norms are used for these vectors. Similarly, $\kappa_p(\Delta t)$ denotes the same quotient for the elemental pressure (and eventually, pressure spatial derivative) values. Tables 1 to 6 summarize the most outstanding results obtained for this problem.

The backward Euler scheme ($\alpha = 1$) is clearly first order accurate both for only one correction per time–step (i.e., with the scheme of Section 3) and when iteration to convergence is performed. In the latter case, it took an average of 10 iterations per time–step to reduce the initial residuals by 7 orders of magnitude, for both 4–noded and 9–noded elements. As can be observed, linear elements admitted larger time-steps. It

can also be seen that in all cases the pressure approximation is also first order accurate.

The Crank–Nicholson case $\alpha = 1/2$ is a second order method. Nevertheless, iterating to convergence is compulsory in this case to maintain the second order accuracy of the scheme. If a fixed number of 2 iterations per time step is chosen, second order accuracy is lost, but the quotients obtained are still larger than 2 (indicating a higher order than 1).

6.2 Cavity flow

The second problem studied was the classical lid–driven cavity flow problem. The leaky–lid case, in which the unit horizontal velocity of the top lid decreases linearly on a small region of both side walls, was considered, so as to remove the pressure singulary at the top–right corner. A regular, non–uniform mesh, which is finer near the boundaries, was used; once again, the same mesh points define the grid for both elements. It consists of 30×30 of the 4–noded elements. A Reynolds number of 1000 was chosen. Computations were carried out starting from the fluid at rest until a steady state was reached. A fixed number of 2 iterations per time–step was used, and the parameter α was set equal to 1, so as to get a converged solution fastest.

Figure 1 shows the steady streamlines obtained with both elements. Secondary bottom left and right vortices can be observed, but no top left vortex was found. This fact is in good agreement with other numerical results obtained for the same Reynolds number (see [6], [8], [13], [15]). The element pressures computed with the Q1P0 element are shown in Figure 2. A checkerboarding phenomenon becomes apparent, which invalidates the pressure approximation without affecting the velocities. On the other hand, the Q2P1 element gave satisfactory pressure results; the pressure contours obtained after a nodal interpolation process through a least–squares method can be seen in Figure 3, and compare well with those of the above mentioned references.

Finally, Figure 4 shows the horizontal velocity profile at the centerline of the cavity ($x = 0.5$). As can be seen, these results compare well with the reference data of [8],

specially for linear elements.

6.3 Plane jet simulation

The third example considered is a purely unsteady case, consisting of a plane jet simulation. The same conditions and mesh as in [17] were taken. Once again, 2 iterations per time step of the algorithm were performed. The streamline and pressure contours at different times are shown in Figures 5 and 6 respectively; they are in good agreement with the results of [17]. The presence of outlet boundary conditions on part of the boundary prevents the appearance of spurious checkboard modes for the Q1P0 element. This example shows the capability of the algorithm to reproduce purely unsteady situations.

7. CONCLUSIONS

In this paper a fractional–step method for the approximation of the incompressible Navier–Stokes equations in primitive variables has been developed, which depends on a free parameter $\theta \in (0,1]$. For $\theta = 1$, both the intermediate and the end–of–step velocities have been shown to converge to an exact solution in the space $(H_0^1(\Omega))^d$. By introducing a viscous term in the second step of the scheme, full boundary conditions are imposed on the velocity in all steps, whereas no boundary conditions at all are imposed on the pressure. Thus, this method does not introduce the numerical boundary layer present in some other fractional–step methods. One drawback of this formulation is the need for the space interpolations of velocity and pressure to satisfy the compatibility conditions met when solving the Stokes problem in primitive variables.

Moreover, this method has been related to an existing predictor–multicorrector algorithm. A new derivation of the algorithm, in a more general context than the classical one, has been given, showing in what sense it can be considered as a fractional–step iterative method. The present method can actually be understood as a predictor–corrector form of the algorithm, in which no iteration occurs. Two finite element implementations of the algorithm, using the Q1P0 element and the Q2P1 element respectively, have been considered. The latter is particularly well suited for this algorithm since it satisfies the discrete LBB condition.

Numerical results confirm the theoretical properties of these methods. First, a numerical convergence study is provided, which shows that first order accuracy for the velocity can be obtained for $\alpha = 1$ with only one correction per time–step, which attains first order accuracy for the pressure too. Also, a second order method is found by setting $\alpha = 1/2$ and iterating to convergence in each time–step. Then, the performance of the scheme on standard benchmark tests for Navier–Stokes solvers, both for steady state and unsteady cases, has also been studied. A comparison between the two finite element interpolations shows that linear elements impose less severe restrictions on the time–step for the stability of the iterative scheme, whereas quadratic elements are usable in

all circumstances (even when velocity is fixed on all the boundary) and require a smaller number of pressure unknowns, thus reducing computational cost.

APPENDIX A

In the notation used here, given $f \in L^2(0,T;(H^{-1}(\Omega))^d)$, the weak form of (2.1) consists of finding $u \in L^2(0,T;(H_0^1(\Omega))^d)$ and $p \in L^2(0,T;L_0^2(\Omega))$ such that:

$$\frac{d}{dt}(u(t),v) + ((u\cdot\nabla)u,v) + \nu((u,v)) - (\operatorname{div} v, p) =< f(t), v >, \forall v \in (H_0^1(\Omega))^d \quad (A.1a)$$

$$(\operatorname{div} u, q) = 0, \qquad \forall q \in L_0^2(\Omega) \qquad (A.1b)$$

A solution of (A.1) exists, which is unique for ν sufficiently large (see [27]). In the 2–dimensional case, the solution is always unique. These solutions are characterized by satisfying (see [27]) $u \in L^2(0,T;V)$ and:

$$\frac{d}{dt}(u(t),v) \; + \; ((u(t)\cdot\nabla)u(t),v) \; + \; \nu((u(t),v)) \; =< f(t), v >, \quad \forall v \in V \qquad (A.2)$$

Assume $f \in L^2(0,T;(L^2(\Omega))^d$. Then, one has:

Lemma 1: a priori estimate.

$$|u^N|^2 + \sum_{n=0}^{N-1}(|u^{n+1} - u^{n+1/2}|^2 + |u^{n+1/2} - u^n|^2) \qquad (A.3)$$

$$+ k\nu \sum_{n=0}^{N-1} ||u^{n+1}||^2 + k\nu \sum_{n=0}^{N-1} ||u^{n+1} - u^{n+1/2}||^2 \le d_2$$

where $d_2 = |u^0|^2 + \frac{d_0^2}{\nu}\int_0^T |f(s)|^2\, ds$ and d_0 was introduced in (2.4).

The proof is similar to that of Lemma 7.1.2 in [27]. Notice the last term appearing in the left–hand side of (A.3), which is not present in [27].

Lemma 2: for every $m = 0, \ldots, N-1$:

1) $|u^{m+i/2}|^2 \leq d_2, \;\; i = 1, 2.$

2) $k||u^{m+1/2}||^2 \leq d_2/\nu$

3) $\sum_{n=0}^{N-1} |u^{n+1} - u^{n+1/2}|^2 \leq d_2$

4) $\sum_{n=0}^{N-1} |u^{n+1/2} - u^n|^2 \leq d_2$

5) $k \sum_{n=0}^{N-1} ||u^{n+1} - u^{n+1/2}||^2 \leq d_2/\nu$

6) $k \sum_{n=0}^{N-1} ||u^{n+1}||^2 \leq d_2/\nu$

The proof, again, is similar to that of [27]. Notice that the bound 5) was not obtained in [27].

The approximating functions u_k^i and u_k introduced in Section 4 satisfy, for decreasing k:

Lemma 3: 1) u_k^i and u_k are bounded in $L^\infty(0, T; (L^2(\Omega))^d), \;\; i = 1, 2, 3.$

2) u_k^i and u_k are bounded in $L^2(0, T; (H_0^1(\Omega))^d), \;\; i = 1, 2, 3$

3) $(u_k^2 - u_k^1)$ and $(u_k^2 - u_k^3)$ are bounded in $L^2(0, T; (H_0^1(\Omega))^d)$

These results are a consequence of Lemma 2 and the definitions of the functions. The main novelty with respect to [27] is now the boundedness of u_k^2 and u_k in $L^2(0, T; (H_0^1(\Omega))^d$, together with that of the differences $(u_k^2 - u_k^1)$ and $(u_k^2 - u_k^3)$. Moreover:

Lemma 4: 1) $||u_k^2 - u_k^1||_{L^2(0,T;(L^2(\Omega))^d)} \leq \sqrt{k d_2}$

2) $||u_k^2 - u_k^3||_{L^2(0,T;(L^2(\Omega))^d)} \leq \sqrt{4 k d_2}$

3) $||u_k - u_k^2||_{L^2(0,T;(L^2(\Omega))^d)} \leq \sqrt{\frac{4 k d_2}{3}}$

Proof: part 1) follows from Lemma 2, part 3); 2) results from Lemma 2, parts 3) and 4) and the triangle inequality. Finally, 3) is a consequence of the definition of u_k

and Lemma 2, parts 3) and 4).

Lemma 5:

$$\begin{aligned} \frac{d}{dt}(u_k(t),v) &= -\nu((u_k^2(t),v)) - c(u_k^3(t),u_k^1(t),v) + (f_k(t),v) \\ &\equiv < g_k(t),v >, \qquad \forall v \in V,\ \forall t \in (0,T), \end{aligned} \tag{A.4}$$

with g_k bounded in $L^2(0,T;V')$. (For the definition of f_k, see [27]). In particular, u_k is a.e. equal to a continuous function from $[0,T]$ into V.

Proof: by adding (4.3) and (4.4a) for $v \in V$, one gets:

$$(\frac{u^{n+1}-u^n}{k},v)+\nu((u^{n+1},v))+c(u^n,u^{n+1/2},v) = < \overline{f}^n,v > \quad \forall v \in (H_0^1(\Omega))^d \tag{A.5}$$

so that (A.4) follows from the above definitions. Besides:

$$||g_k(t)||_{V'} \leq \nu||u_k^2(t)|| + C\,||u_k^3(t)||\,||u_k^1(t)|| + |f_k(t)|, \tag{A.6}$$

where $C > 0$ is a constant related to the continuity of the trilinear form c; the remaining statements are a consequence of Lemma 3 and Lemma III.1.1 of [27].

The proof of the theorem is now ready:

Since u_k^i $(i = 1,2,3)$ and u_k are bounded in $L^\infty(0,T;(L^2(\Omega))^d$, there exists a subsequence k' (which can be taken the same for all 4 sequences) and u^i $(i = 1,2,3)$, $u^* \in L^\infty(0,T;(L^2(\Omega))^d$ such that:

$$\begin{aligned} u_{k'}^i &\longrightarrow u^i \text{ in } L^\infty(0,T;(L^2(\Omega))^d \quad \text{weak} - \text{star} \ (i=1,2,3) \\ u_{k'} &\longrightarrow u^* \text{ in } L^\infty(0,T;(L^2(\Omega))^d \quad \text{weak} - \text{star} \end{aligned}$$

Since $u^i_{k'}$ $(i = 1,2,3)$ and $u_{k'}$ are bounded in $L^2(0,T;(H^1_0(\Omega))^d$, there exists a subsequence of k' (which is also denoted by k') such that:

$$u^i_{k'} \longrightarrow u^i \text{ in } L^2(0,T;(H^1_0(\Omega))^d \quad \text{weakly} \quad (i = 1,2,3)$$
$$u_{k'} \longrightarrow u^* \text{ in } L^2(0,T;(H^1_0(\Omega))^d \quad \text{weakly}$$

This convergence also holds in $L^2(0,T;(L^2(\Omega))^d$. Since, by Lemma 4:

$$(u^2_{k'} - u^3_{k'}),\ (u^2_{k'} - u^1_{k'}),\ (u^2_{k'} - u_{k'}) \longrightarrow 0 \text{ in } L^2(0,T;(L^2(\Omega))^d \quad \text{strongly,}$$

it must be $u^1 = u^2 = u^3 = u^*$.

Since $u^2_{k'} \in L^\infty(0,T;H) \cap L^2(0,T;V)$, one has that $u^*(t) \in V$ a.e. in $(0,T)$, and $u^* \in L^\infty(0,T;H) \cap L^2(0,T;V)$.

The proof of strong convergence in $L^2(Q)$ is the same as in [27], and is therefore omitted. It only remains to show that u^* is a solution of (2.1). The same argument as in [27] is used, so that the convergence results already proved imply, by taking (A.4) to the limit when k' tends to 0, that:

$$\frac{d}{dt}(u^*,v) + \nu((u^*,v)) + c(u^*,u^*,v) = (f,v) \quad \forall v \in V \qquad (A.7)$$

in distribution sense in $(0,T)$, i.e., u^* satisfies (A.2). This, in turn, implies (see [27]) that $\frac{du^*}{dt} \in L^1(0,T;V')$, $u^*(0) = u^0$ weakly in V and u^* is a.e. equal to a continuous function from $(0,T)$ into V. These results ensure that u^* is a weak solution of (2.1), and the theorem is thus proved.

The proof of the Corolary is a consequence of the uniqueness of the solution u in the two–dimensional case.

APPENDIX B

It is shown here that the scheme (5.3)–(5.4) is equivalent to the predictor–multicorrector algorithm of [4].

Assume the velocity u^n, acceleration a^n, pressure p^n and pressure temporal variation $\dot{p}^n$ are known at time $t_n = n\Delta t$, satisfying the Navier–Stokes equation (5.2) and the incompressibility condition. The iterative predictor–multicorrector procedure starts with the following predictions:

$$u_0^{n+1} = u^n + (1-\alpha)\,\Delta t\, a^n \qquad (B.1a)$$

$$a_0^{n+1} = 0 \qquad (B.1b)$$

$$p_0^{n+1} = p^n + (1-\alpha)\,\Delta t\, \dot{p}^n \qquad (B.1c)$$

$$\dot{p}_0^{n+1} = 0 \qquad (B.1d)$$

Assume, further that after each correction phase the approximation of velocity and pressure may be written as:

$$u_i^{n+1} = u^n + (1-\alpha)\,\Delta t\, a^n + \alpha\,\Delta t\, a_i^{n+1} \qquad (B.2a)$$

$$p_i^{n+1} = p^n + (1-\alpha)\,\Delta t\, \dot{p}^n + \alpha\,\Delta t \dot{p}_i^{n+1} \qquad (B.2b)$$

where a_i^{n+1} and $\dot{p}_i^{n+1}$ are the corrected values at the end of the i–th iteration. Note that equations (B.2) are also valid for the initial prediction ($i = 0$).

The objective of each iteration is to compute new approximations u_{i+1}^{n+1} and p_{i+1}^{n+1} by computing corrected values of a_{i+1}^{n+1} and $\dot{p}_{i+1}^{n+1}$. Since each iteration is split into two steps, an intermediate velocity $u_{i+1/2}^{n+1}$ and acceleration $a_{i+1/2}^{n+1}$ are first calculated. The intermediate velocity is expressed as:

$$u_{i+1/2}^{n+1} = u^n + (1-\alpha)\,\Delta t\, a^n + \alpha\,\Delta t\, a_{i+1/2}^{n+1} \qquad (B.3)$$

while the intermediate acceleration is defined as:

$$a_{i+1/2}^{n+1} = a_i^{n+1} + \Delta a_1 \qquad (B.4)$$

then the following relation is deduced from (B.2a):

$$u_{i+1/2}^{n+1} = u_i^{n+1} + \alpha\,\Delta t\,\Delta a_1 \qquad (B.5)$$

The intermediate velocity and acceleration are computed from the first split step, namely equations (5.3a) and (5.3b), which can thus be rewritten as:

$$\Delta a_1 - \alpha\nu\Delta t(\nabla^2\Delta a_1) = f^{n+1} - a_i^{n+1} - \nabla p_i^{n+1} + \nu\nabla^2 u_i^{n+1} - (u_i^{n+1}\cdot\nabla)u_i^{n+1} \qquad (B.6a)$$

$$\Delta a_1|_\Gamma = 0 \qquad (B.6b)$$

The end–of–step velocity u_{i+1}^{n+1} is expressed, using (B.2a), as:

$$u_{i+1}^{n+1} = u^n + (1-\alpha)\,\Delta t\, a^n + \alpha\,\Delta t\, a_{i+1}^{n+1} \qquad (B.7)$$

and can be further simplified in terms of the intermediate velocity, using equation (B.3), and the end–of–step acceleration, which is defined as:

$$a_{i+1}^{n+1} = a_{i+1/2}^{n+1} + \Delta a_2, \qquad (B.8)$$

Thus:

$$u_{i+1}^{n+1} = u_{i+1/2}^{n+1} + \alpha\, \Delta t\, \Delta a_2 \qquad (B.9)$$

Likewise, from (B.2b), the end–of–step pressure p_{i+1}^{n+1} is expressed as:

$$p_{i+1}^{n+1} = p^n + (1-\alpha)\, \Delta t\, \dot{p}^n + \alpha\, \Delta t\, \dot{p}_{i+1}^{n+1} \qquad (B.10)$$

where the new pressure variation is determined by:

$$\dot{p}_{i+1}^{n+1} = \dot{p}_i^{n+1} + \Delta\dot{p} \qquad (B.11)$$

and consequently one gets:

$$p_{i+1}^{n+1} = p_i^{n+1} + \alpha\, \Delta t\, \Delta\dot{p} \qquad (B.12)$$

With the previous expressions of the end–of–step velocity and pressure, equations (B.9) and (B.12) respectively, the second split step defined by the equation system (5.4) can be written as:

$$\Delta a_2 - \alpha\nu\Delta t(\nabla^2 \Delta a_2) + \alpha\Delta t\nabla(\Delta\dot{p}) = 0 \qquad (B.13a)$$

$$\mathrm{div}\,(\Delta a_2) = \frac{-1}{\alpha\Delta t}\mathrm{div}\,(u_i^{n+1} + \alpha\Delta t\, \Delta a_1) \qquad (B.13b)$$

$$\Delta a_2|_\Gamma = 0 \qquad (B.13c)$$

Given the time–discretization scheme defined by (B.2), equations (B.6) and (B.13) are the discretized version of the two split–step equations (5.3) and (5.4) defined previously. After they have been solved, the corresponding corrections are performed, namely equations (B.7) and (B.10).

The fully discrete algorithm obtained by a finite element approximation of the predictions (B.1) and the multicorrector scheme (B.6), (B.13), (B.7) and (B.10), with the approximation of the matrix B by M^L, is, then:

$$M^L \ \Delta A_1 = R_1 \qquad (B.14a)$$

$$\alpha^2 \ (\Delta t)^2 \ (G^T (M^L)^{-1} G) \ (\Delta \dot{P}) = G^T \ (U_i^{n+1} + \alpha \Delta t (\Delta A_1)) \qquad (B.14b)$$

$$M^L \ \Delta A_2 = \alpha \ \Delta t \ G \ (\Delta \dot{P}) \qquad (B.14c)$$

where:

$$R_1 = F^{n+1} - M A_i^{n+1} - \nu K U_i^{n+1} - C(U_i^{n+1}) U_i^{n+1} + G P_i^{n+1}, \qquad (B.16)$$

the vector A represents the nodal accelerations and the rest of the matrices and vectors are defined in Section 5. This is the same algorithm as used in [4].

REFERENCES.

1.- J. B. Bell, P. Colella, H. M. Glaz: " A Second–Order Projection Method for the incompressible Navier–Stokes equations." *Journal of Computational Physics,* Vol. **85**, p. 257–283, 1989.

2.- J. Blair Perot: " An Analysis of the fractional–step projection method." *Journal of Computational Physics,* Vol. **108**, p. 51–58, 1993.

3.- F. Brezzi, M. Fortin: *Mixed and Hybrid finite element methods.* Springer Series in Computational Mathematics, 15. Springer–Verlag, 1991.

4.- A. N. Brooks, T. J. R. Hughes: " Streamline upwind Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations." *Computer Methods in Applied Mechanics and Engineering,* Vol. **32**, p. 199–259, 1982.

5.- A.J.Chorin: " Numerical solution of the Navier–Stokes equations." *Mathematics of Computation,* Vol. **22**, p. 745–762, 1968.

6.- J. Donea, S. Giuliani, H. Laval, L. Quartapelle: " Finite element solution of the unsteady Navier–Stokes equations by a fractional step method." *Computer Methods in Applied Mechanics and Engineering,* Vol. **30**, p. 53–73, 1982.

7.- J. K. Dukovicz, A. S. Dvinsky: " Approximate Factorization as a High Order Splitting for the Implicit Incompressible Flow Equations." *Journal of Computational Physics,* Vol. **102**, p. 336–347, 1992.

8.- U. Ghia, K. N. Ghia, C. T. Shin: " High–*Re* solutions for incompressible flow using the Navier–Stokes equations and a multigrid method." *Journal of Computational Physics,* Vol. **48**, p. 387–411, 1982.

9.- V. Girault, P. A. Raviart: *Finite Element Approximation of the Navier–Stokes Equation.* Springer–Verlag, New York, 1986.

10.- P. M. Gresho: " On the theory of semi–implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix. Part I: theory." *International Journal for Numerical Methods in Fluids,* Vol. **11**, p. 587–620, 1990.

11.- P. M. Gresho, R. Sani: " On pressure boundary conditions for the incompressible Navier–Stokes equations." *International Journal for Numerical Methods in Fluids,* Vol. **7**, p. 1111–1145, 1987.

12.- J. van Kan: " A second–order accurate pressure correction scheme for viscous incompressible flow." *SIAM Journal of Sci. Stat. Comp.,* Vol. **7**, N. 3, p. 870–891, 1986.

13.- J. Kim, P. Moin: " Application of a Fractional–Step Method to incompressible Navier–Stokes equations." *Journal of Computational Physics,* Vol. **59**, p. 308–323, 1985.

14.- L. Kleiser, U. Shumann: *Treatment of incompressibility and boundary conditions in 3–D numerical spectral simulation of plane channel flows.* Hirschel E. H. ed., Proc. 3rd GAMM Conference on Numerical Methods in Fluid Mechanics, Vieweg–Verlag Braunschweig, 1980.

15.- A. Kovacs, M. Kawahara: " A finite element scheme based on the velocity–correction method for the solution of the time–dependent incompressible Navier–Stokes equations." *International Journal for Numerical Methods in Fluids,* Vol. **13**, p. 403–423, 1991.

16.- O. A. Ladyzhenskaya: *The mathematical theory of viscous incompressible flow.* Gordon and Breach, New York, 1969.

17.- H. Laval, L. Quartapelle: " A fractional–step Taylor–Galerkin method for unsteady incompressible flows." *International Journal for Numerical Methods in Fluids,* Vol. **11**, p. 501–513, 1990.

18.- H. Le, P. Moin: " An improvement of Fractional–Step Methods for the incompressible Navier–Stokes equations." *Journal of Computational Physics,* Vol. **92**, p. 369–379, 1991.

19.- W. K. Liu, J. Gvildys: " Fluid–Structure interaction of tanks with an eccentric core barrel." *Computer Methods in Applied Mechanics and Engineering,* Vol. **58**, p. 51–77, 1986.

20.- R. Natarajan: " A Numerical Method for Incompressible Viscous Flow Simulation." *Journal of Computational Physics,* Vol. **100**, p. 384–395, 1992.

21.- S. A. Orzag, M. Israeli, M. O. Deville: " Boundary conditions for incompressible flows." *Journal of Scientific Computing,* Vol. **1**, N. 1, p. 75–111, 1986.

22.- L. Quartapelle, M. Napolitano: " Integral conditions for the pressure in the computation of incompressible viscous flows." *Journal of Computational Physics,* Vol. **62**, p. 340–348, 1986.

23.- L. Quartapelle: *Numerical Solution of the Incompressible Navier-Stokes Equations.* Birkhäuser, International Series in Numerical Mathematics Vol. 113, 1993.

24.- J. Shen: " Hopf bifurcation of the unsteady regularized driven cavity flow." *Journal of Computational Physics,* Vol. **95**, p. 228–245, 1991.

25.- J. Shen: " On error estimates of projection methods for Navier–Stokes equations: first–order schemes." *SIAM Journal Numerical Analysis,* Vol. **29**, p. 57–77, 1992.

26.- R. Temam: " Sur l'approximation de la solution des équations de Navier–Stokes par la méthode des pas fractionaires (I)." *Arch. Rac. Mech. Anal.,* Vol. **32**, N. 2, p. 135–153, 1969.

27.- R. Temam: *Navier–Stokes equations. Theory and Numerical Analysis.* North–

Holland, Amsterdam, 3rd. Edition, 1984.

28.- R. Temam: " Remark on the Pressure Boundary Condition for the Projection Method." *Theoretical and Computational Fluid Dynamics,* Vol. **3**, p. 181–184, 1991.

29.- N. N. Yanenko: *The Method of Fractional Steps.* Springer–Verlag, Berlin, 1971.

LIST OF TABLES:

FIGURE CAPTIONS:

Figure 1: Cavity flow, streamlines. a) Q1P0 element.

b) Q2P1 element.

Figure 2: Q1P0 element, element pressure values.

Figure 3: Q2P1 element, nodal pressure contours.

Figure 4: Horizontal velocity profile through cavity centerline $x = 0.5$:

—— $Q1P0$; - - - $Q2P1$; · Ghia et al.

Figure 5: Jet flow, streamlines: a) Q1P0 element, $t = 1.2$.

b) Q1P0 element, $t = 2.5$

c) Q1P0 element, $t = 4.0$

d) Q2P1 element, $t = 1.2$

e) Q2P1 element, $t = 2.5$

f) Q2P1 element, $t = 4.0$

Figure 6 Pressure contours: a) Q1P0 element, $t = 1.2$.

b) Q1P0 element, $t = 2.5$.

c) Q1P0 element, $t = 4.0$.

d) Q2P1 element, $t = 1.2$.

e) Q2P1 element, $t = 2.5$.

f) Q2P1 element, $t = 4.0$.

Table 1: Q1P0 element, $\alpha = 1$, 1 iteration per time–step.

Δt	$\kappa_1(\Delta t)$	$\kappa_2(\Delta t)$	$\kappa_p(\Delta t)$
1/16	2.04	2.02	2.03
1/32	2.02	2.01	2.02
1/64	2.01	2.01	2.01
1/128	1.99	1.99	2.00

Table 2: Q1P0 element, $\alpha = 1$, iterating to convergence.

Δt	$\kappa_1(\Delta t)$	$\kappa_2(\Delta t)$	$\kappa_p(\Delta t)$
1/16	2.00	1.99	2.02
1/32	1.97	2.02	2.00
1/64	1.97	1.95	2.00
1/128	2.01	1.83	2.01

Table 3: Q1P0 element, $\alpha = 1/2$, iterating to convergence.

Δt	$\kappa_1(\Delta t)$	$\kappa_2(\Delta t)$
1/7	4.06	4.08
1/8	4.04	4.04
1/10	4.03	4.01
1/12	3.75	3.99
1/14	4.00	3.79
1/16	4.05	3.92
1/20	3.92	3.75

Table 4: Q2P1 element, $\alpha = 1$, 1 iteration per time–step.

Δt	$\kappa_1(\Delta t)$	$\kappa_2(\Delta t)$	$\kappa_p(\Delta t)$
1/16	2.24	2.27	2.04
1/32	2.01	1.99	2.02
1/64	2.00	1.99	2.01
1/128	1.97	1.99	2.01

Table 5: Q2P1 element, $\alpha = 1$, iterating to convergence.

Δt	$\kappa_1(\Delta t)$	$\kappa_2(\Delta t)$	$\kappa_p(\Delta t)$
1/32	1.98	1.99	2.01
1/64	1.99	2.00	2.01
1/128	2.00	1.95	1.99

Table 6: Q2P1 element, $\alpha = 1/2$, iterating to convergence.

Δt	$\kappa_1(\Delta t)$	$\kappa_2(\Delta t)$
1/16	3.82	4.15
1/20	3.96	4.04
1/24	3.86	3.92

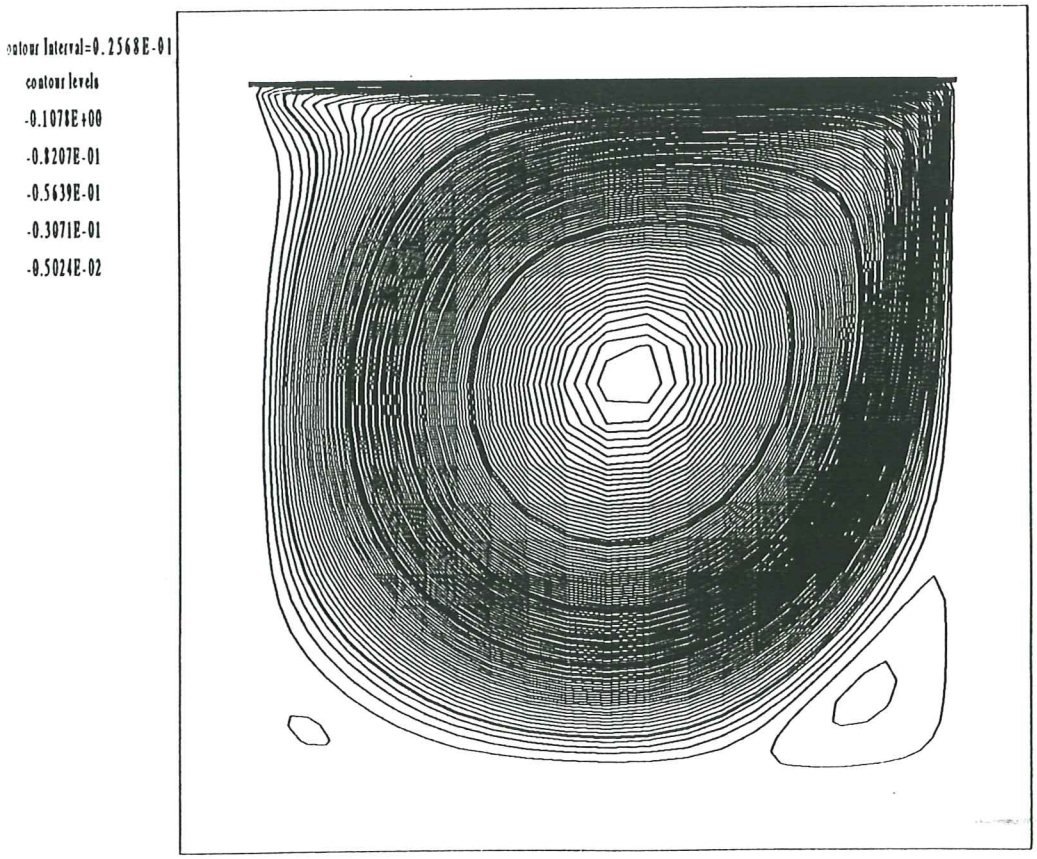

ontour Interval=0.2568E-01
contour levels
-0.1078E+00
-0.8207E-01
-0.5639E-01
-0.3071E-01
-0.5024E-02

Fig. 1a

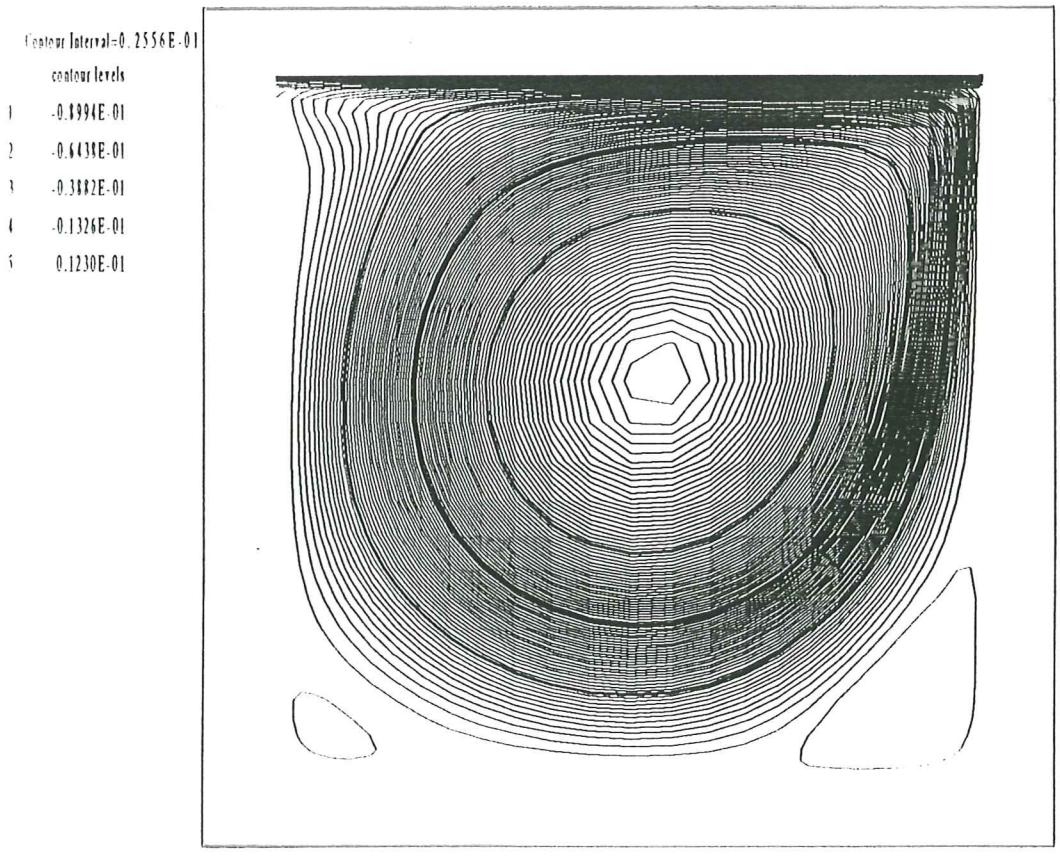

Contour Interval=0.2556E-01
contour levels
1 -0.1994E-01
2 -0.6438E-01
3 -0.3882E-01
4 -0.1326E-01
5 0.1230E-01

Fig. 1b

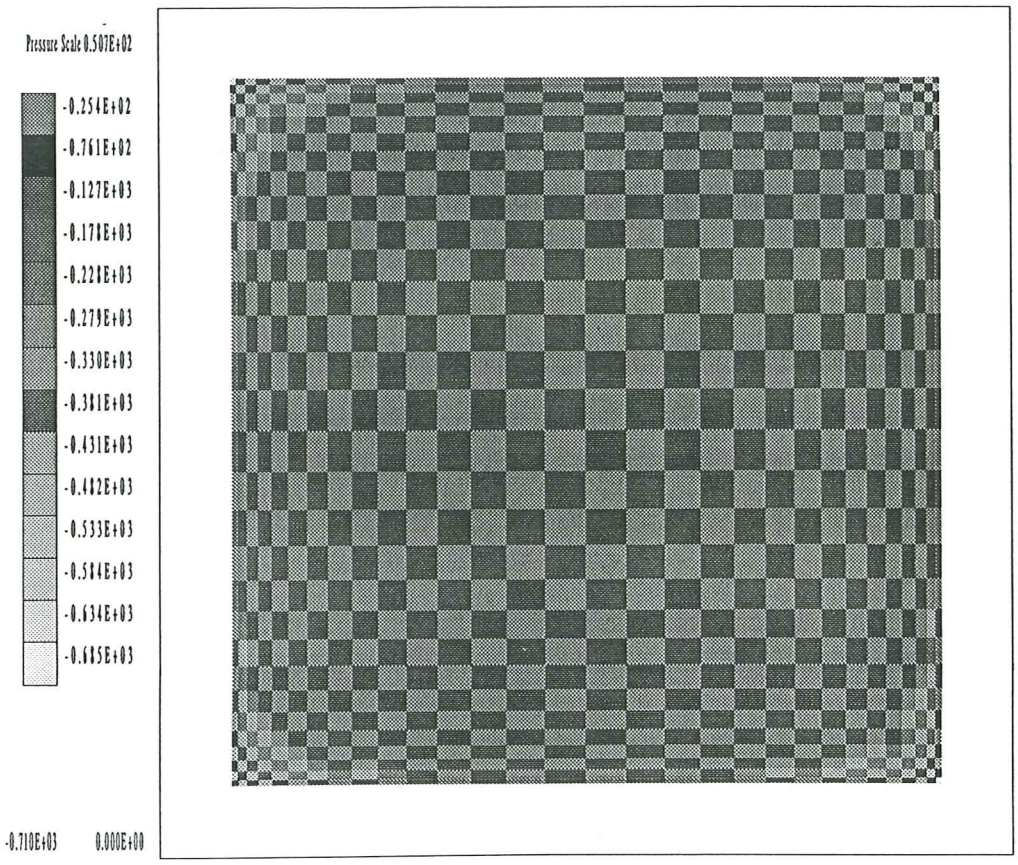

Pressure Scale 0.507E+02
-0.254E+02
-0.761E+02
-0.127E+03
-0.178E+03
-0.228E+03
-0.279E+03
-0.330E+03
-0.381E+03
-0.431E+03
-0.482E+03
-0.533E+03
-0.584E+03
-0.634E+03
-0.685E+03
-0.710E+03
0.000E+00

Fig. 2

Fig. 3

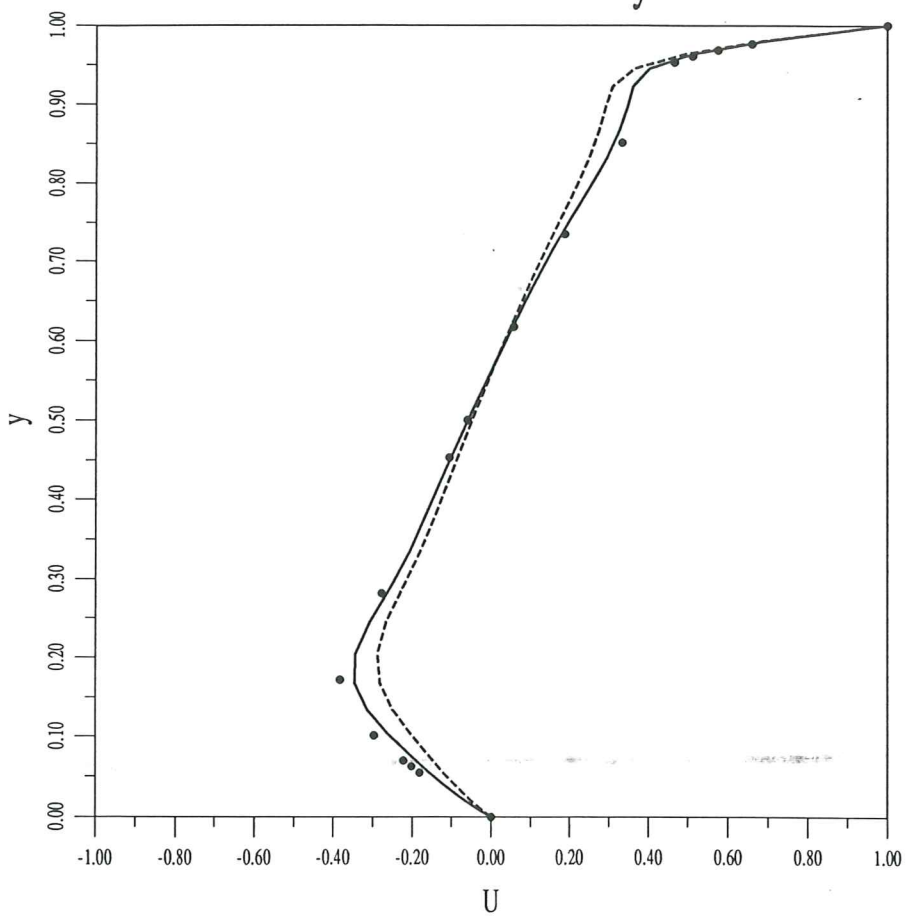

Horizontal velocity
y
U
0.00
0.10
0.20
0.30
0.40
0.50
0.60
0.70
0.80
0.90
1.00
-1.00
-0.80
-0.60
-0.40
-0.20
0.00
0.20
0.40
0.60
0.80
1.00

Fig.

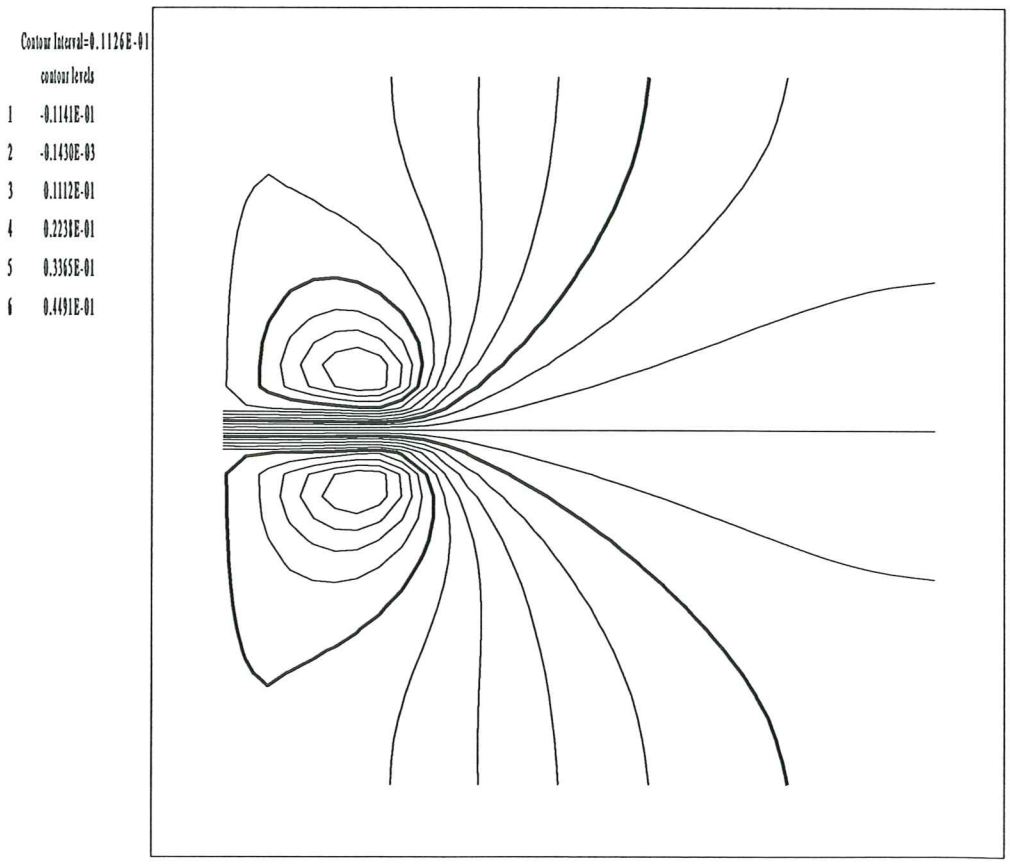

Contour Interval=0.1126E-01
contour levels
1 -0.1141E-01
2 -0.1430E-03
3 0.1112E-01
4 0.2238E-01
5 0.3365E-01
6 0.4491E-01

Fig. 5a

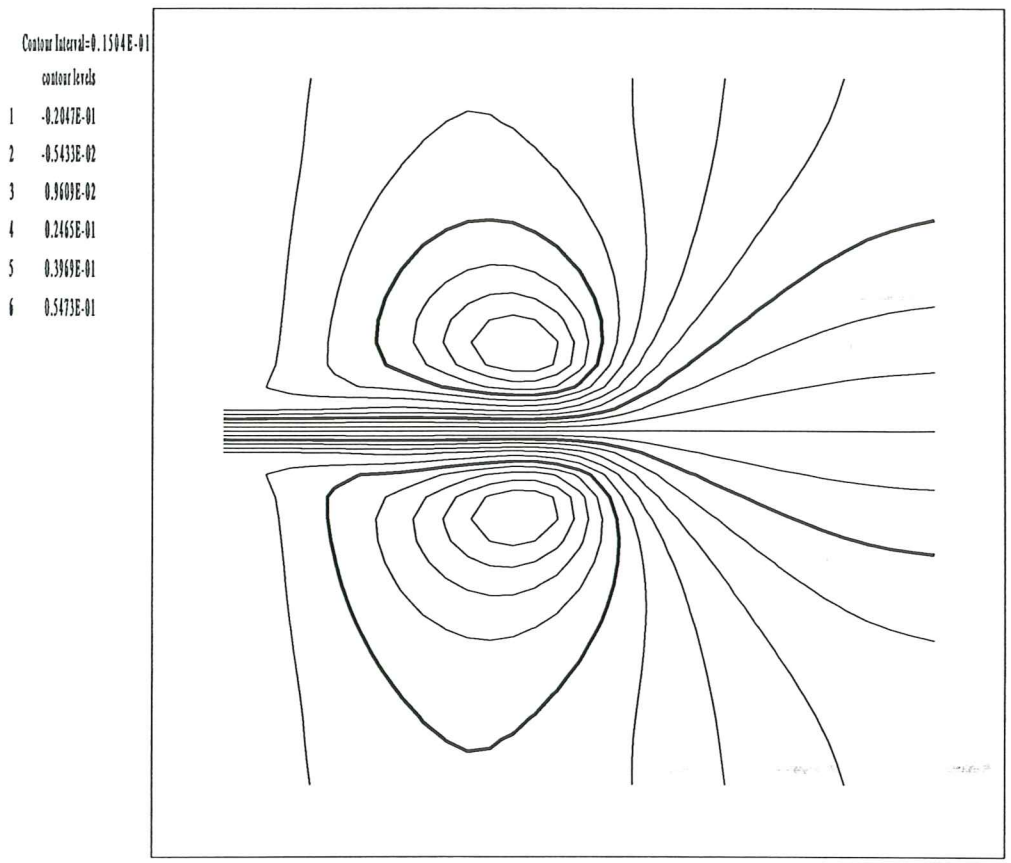

Contour Interval=0.1504E-01
contour levels
1 -0.2047E-01
2 -0.5433E-02
3 0.9609E-02
4 0.2465E-01
5 0.3969E-01
6 0.5473E-01

Fig. 5b

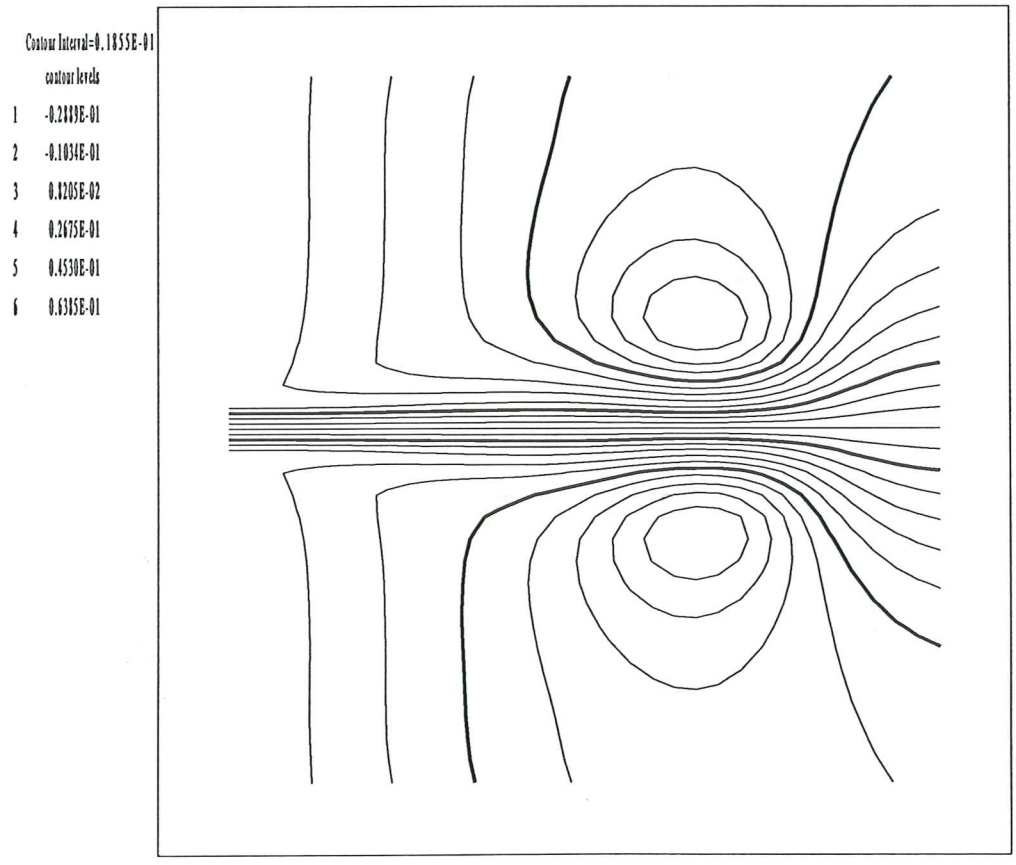

Contour Interval=0.1855E-01
contour levels
1 -0.2489E-01
2 -0.1034E-01
3 0.8205E-02
4 0.2675E-01
5 0.4530E-01
6 0.6385E-01

Fig. 5c

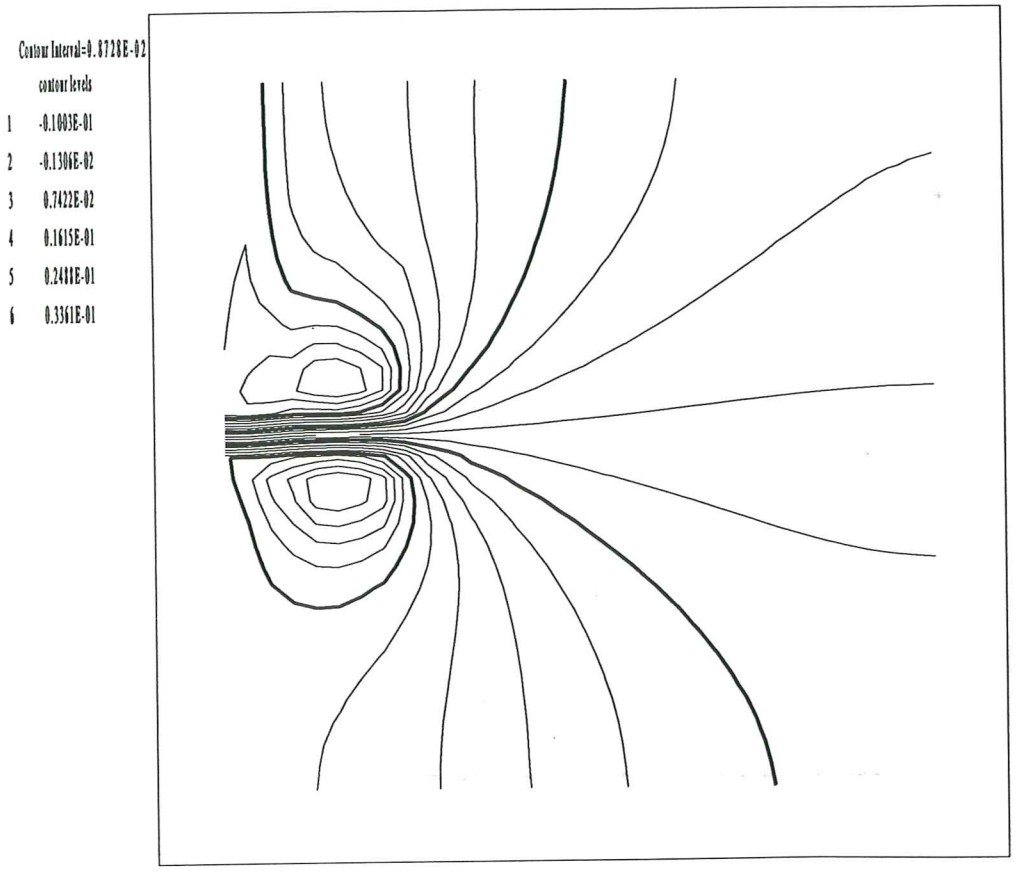

Contour Interval=0.8728E-02
contour levels
1 -0.1003E-01
2 -0.1306E-02
3 0.7422E-02
4 0.1615E-01
5 0.2488E-01
6 0.3361E-01

Fig. 5d

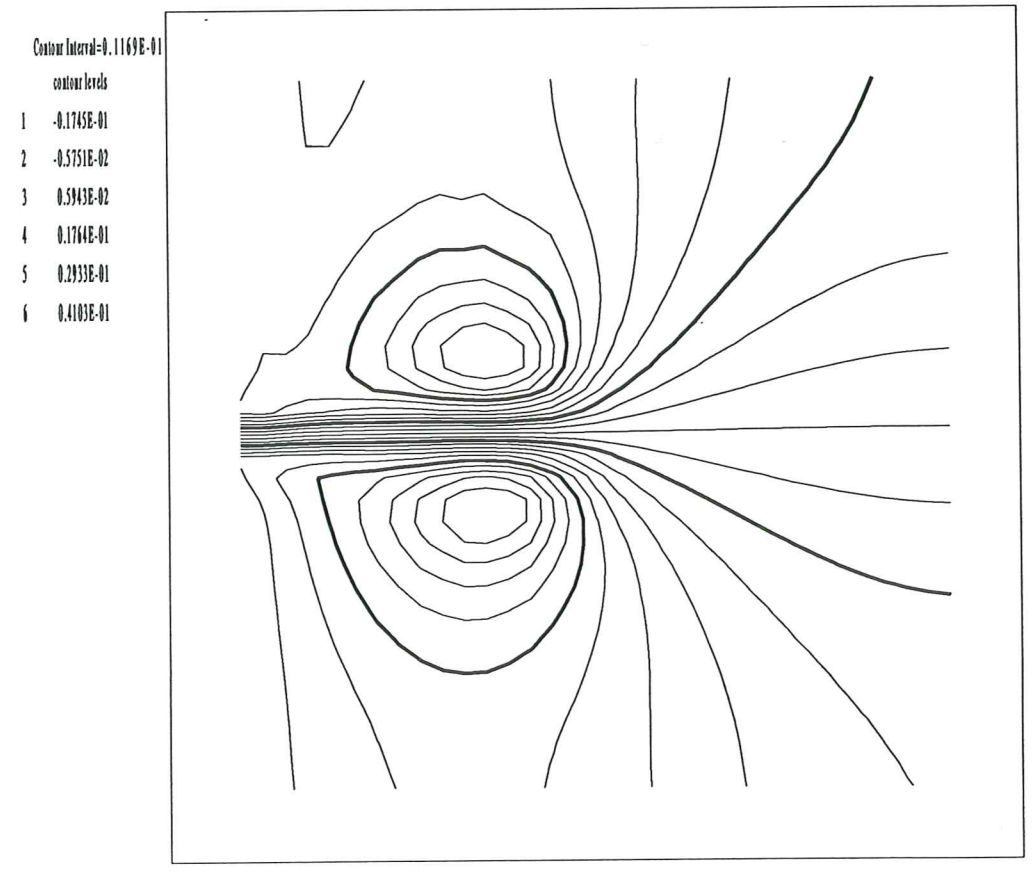

Contour Interval=0.1169E-01
contour levels
1 -0.1745E-01
2 -0.5751E-02
3 0.5943E-02
4 0.1764E-01
5 0.2933E-01
6 0.4103E-01

Fig. 5e

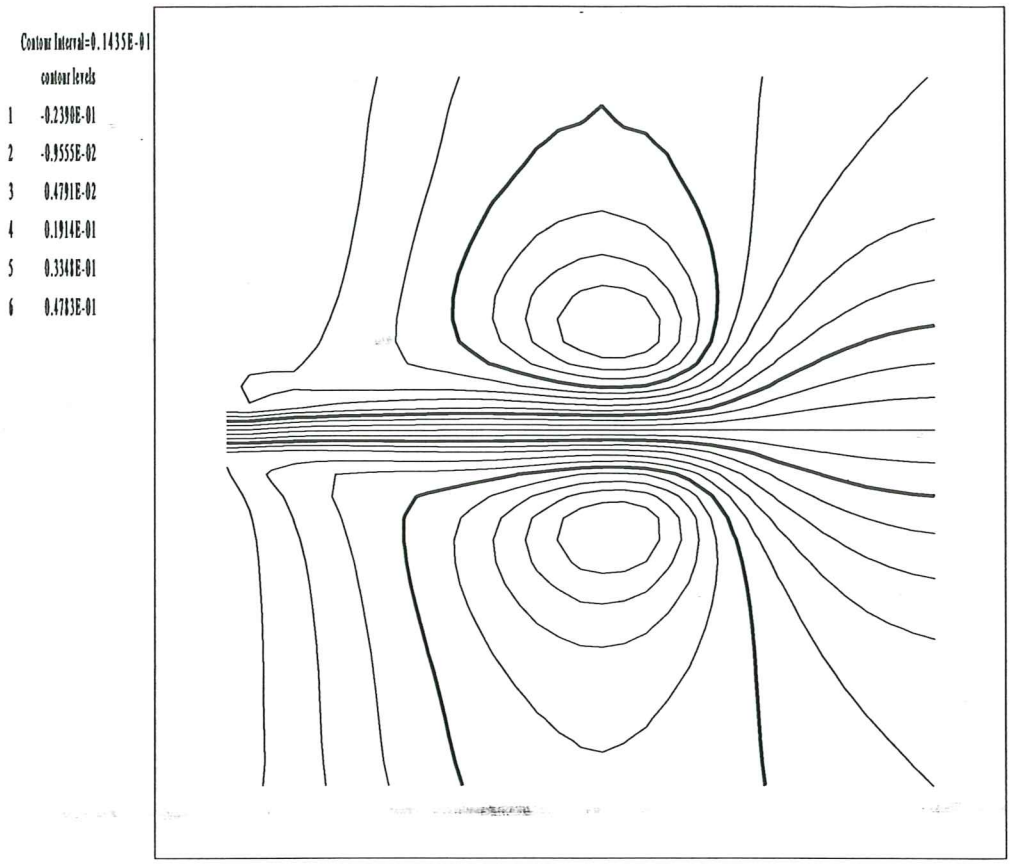

Contour Interval=0.1435E-01
contour levels
1 -0.2390E-01
2 -0.9555E-02
3 0.4791E-02
4 0.1914E-01
5 0.3341E-01
6 0.4783E-01

Fig. 5g

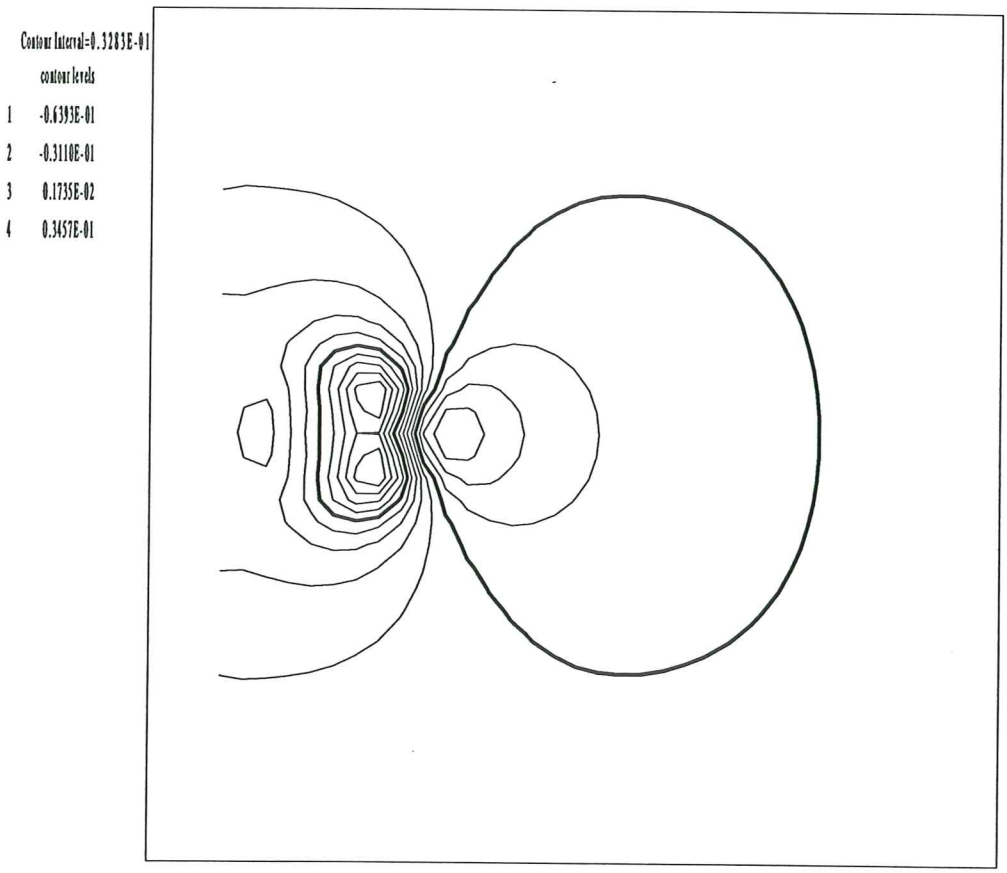

Contour Interval=0.3283E-01
contour levels
1 -0.6393E-01
2 -0.3110E-01
3 0.1735E-02
4 0.3457E-01

Fig. 6a

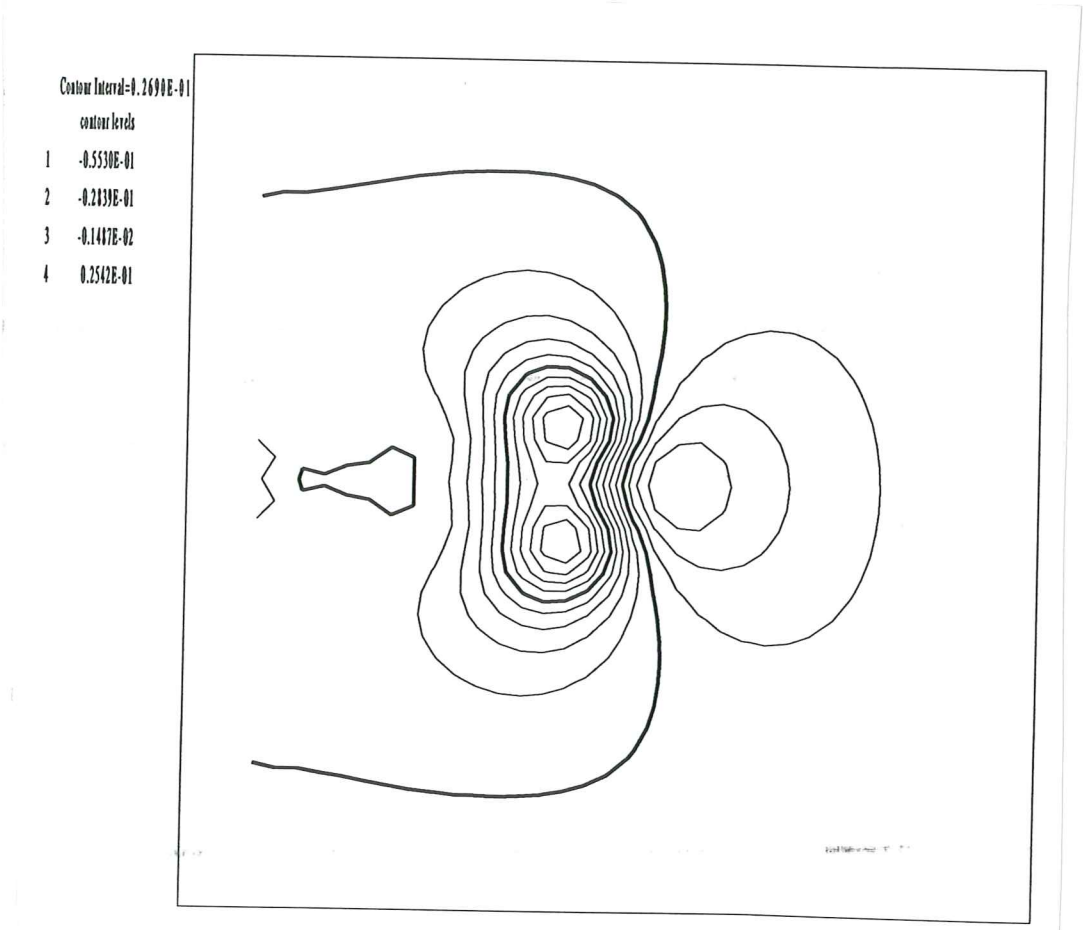

Contour Interval=0.2690E-01
contour levels
1 -0.5530E-01
2 -0.2839E-01
3 -0.1447E-02
4 0.2542E-01

Fig. 6b

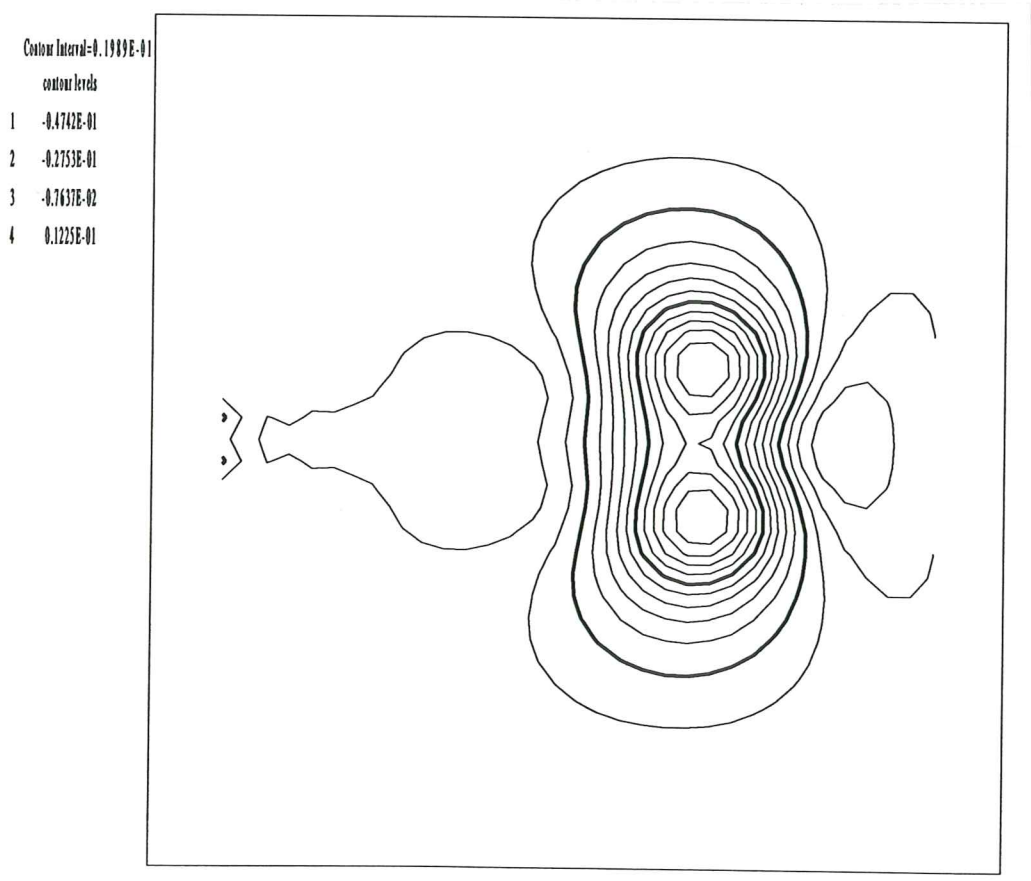

Contour Interval=0.1989E-01
contour levels
1 -0.4742E-01
2 -0.2753E-01
3 -0.7637E-02
4 0.1225E-01

Fig. 6c

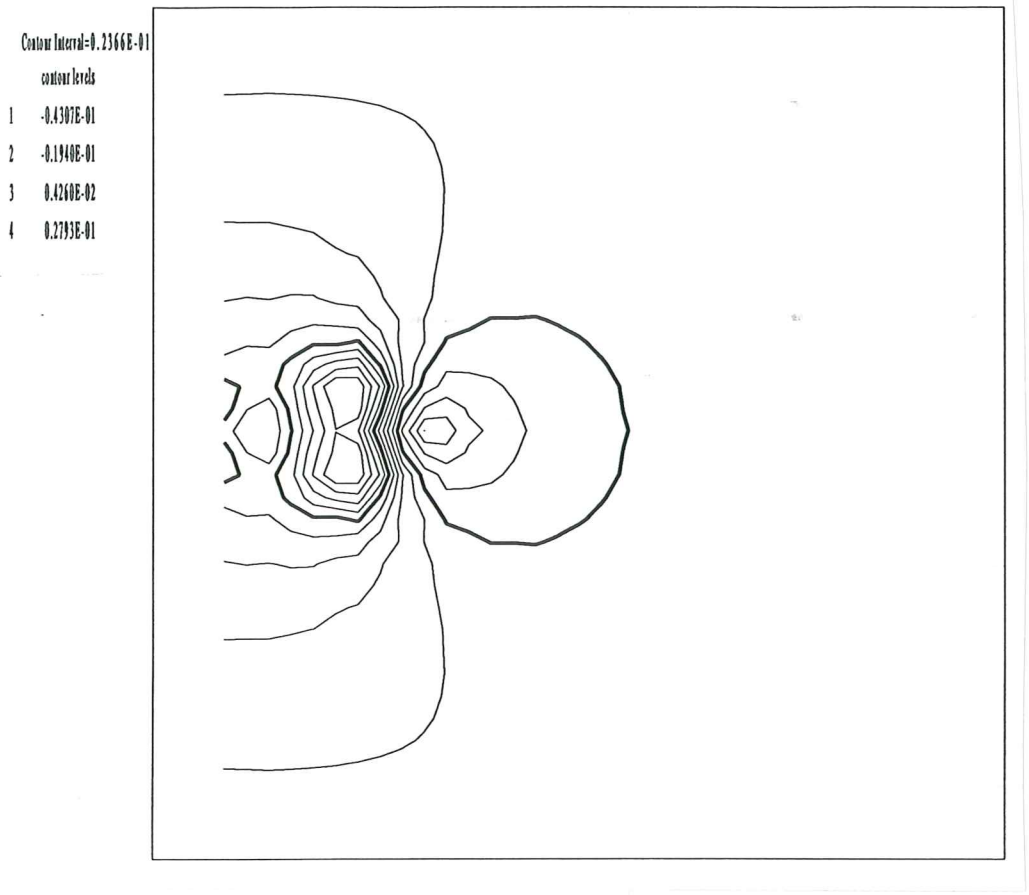

Contour Interval=0.2366E-01
contour levels
1 -0.4307E-01
2 -0.1940E-01
3 0.4260E-02
4 0.2793E-01

Fig. 6d

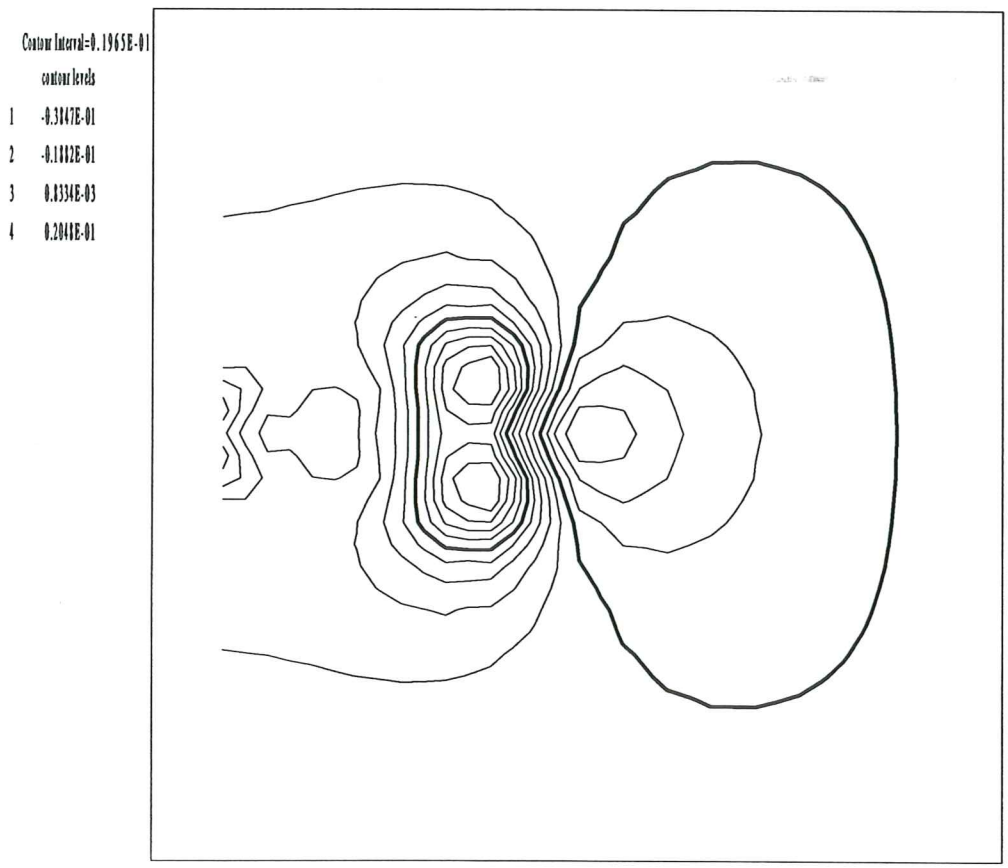

Contour Interval=0.1965E-01
contour levels
1 -0.3147E-01
2 -0.1182E-01
3 0.8334E-03
4 0.2041E-01

Fig. 6e

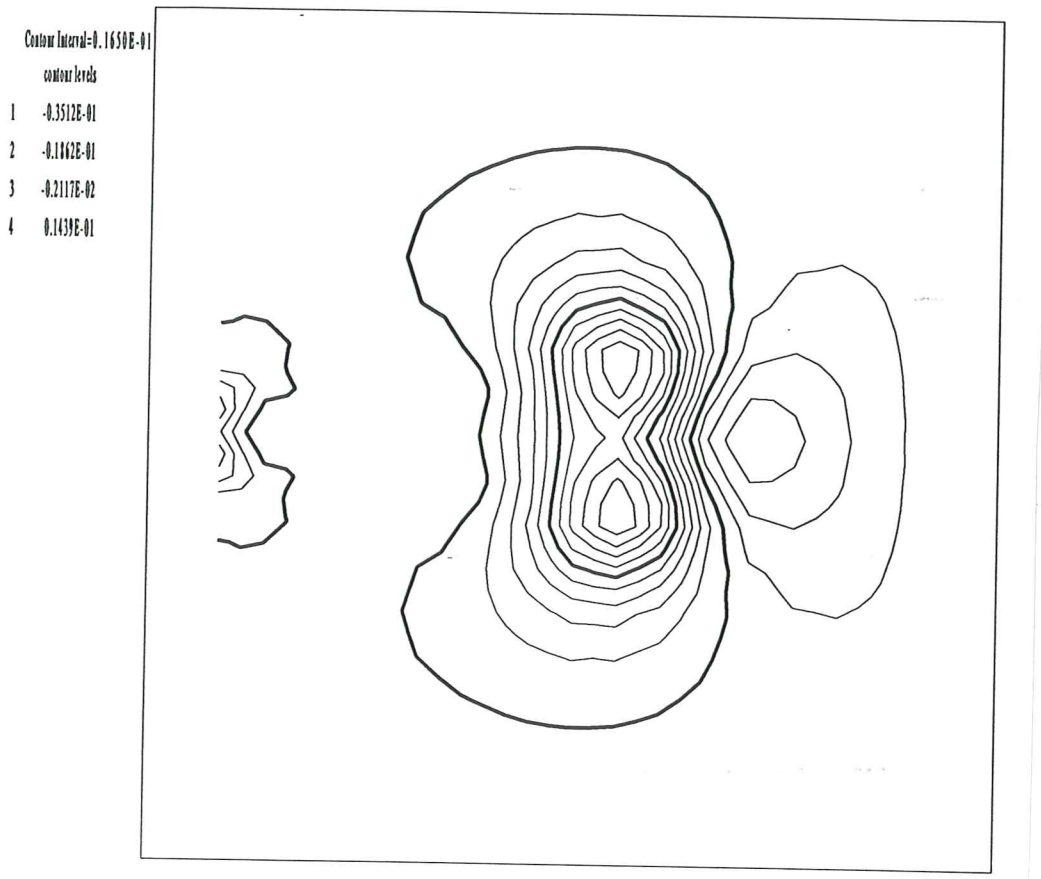

Contour Interval=0.1650E-01
contour levels
1 -0.3512E-01
2 -0.1862E-01
3 -0.2117E-02
4 0.1439E-01

Fig. 6f

Fig 4a

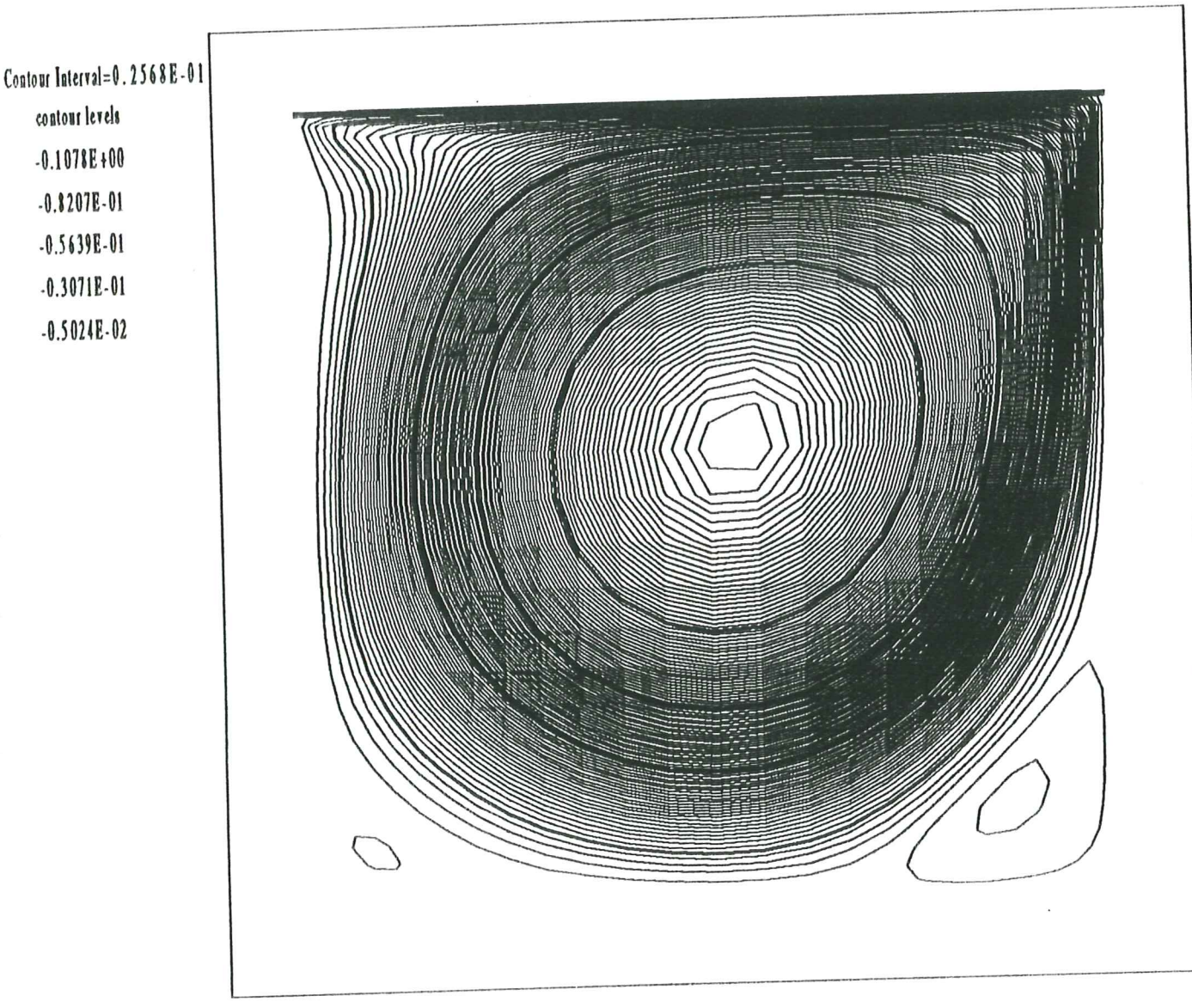

Contour Interval=0.2568E-01
contour levels
-0.1078E+00
-0.8207E-01
-0.5639E-01
-0.3071E-01
-0.5024E-02

Fig. 1 b

Contour Interval=0.2556E-01

contour levels

1 -0.8994E-01

2 -0.6438E-01

3 -0.3882E-01

4 -0.1326E-01

5 0.1230E-01

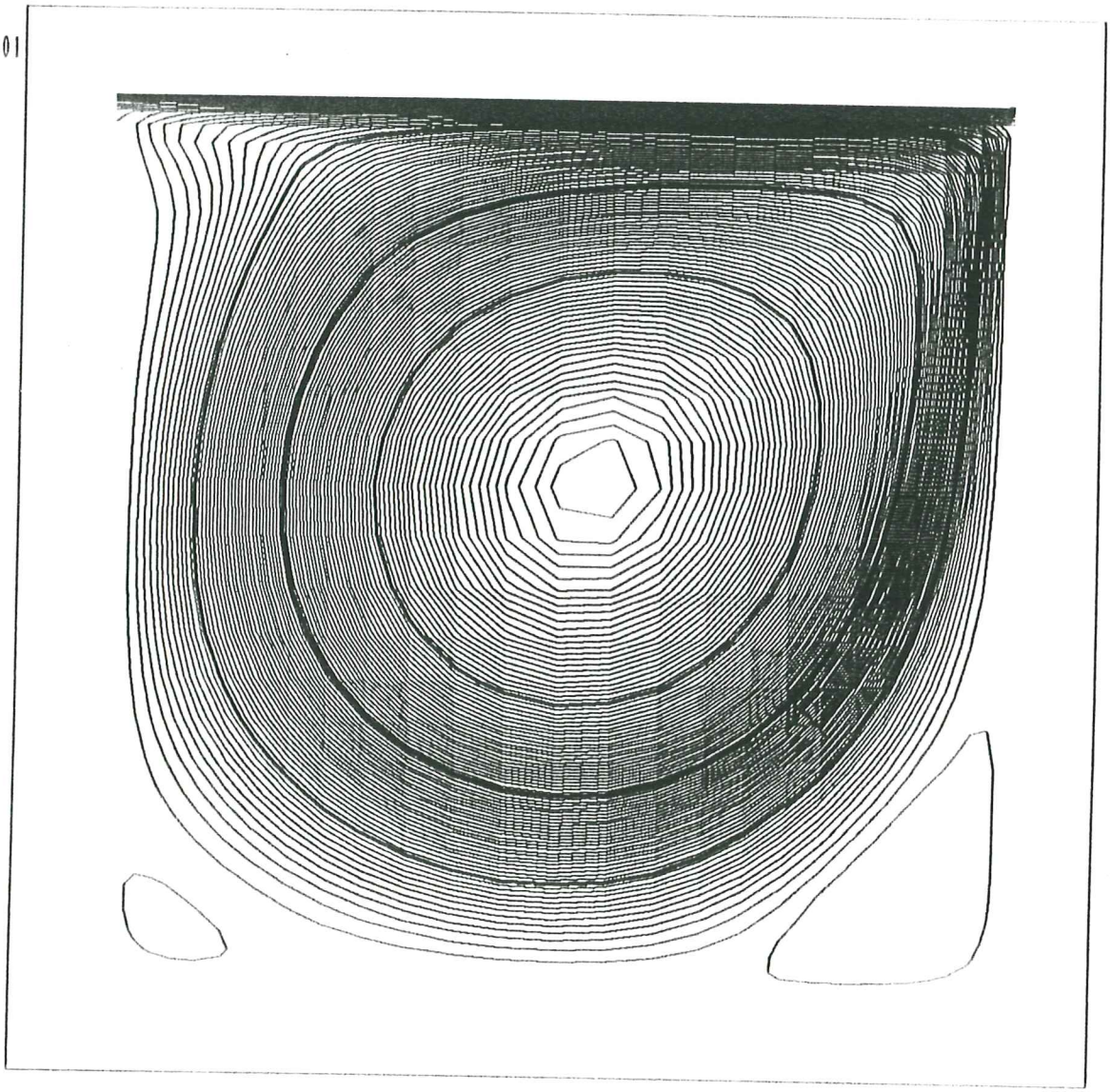


Fig 2

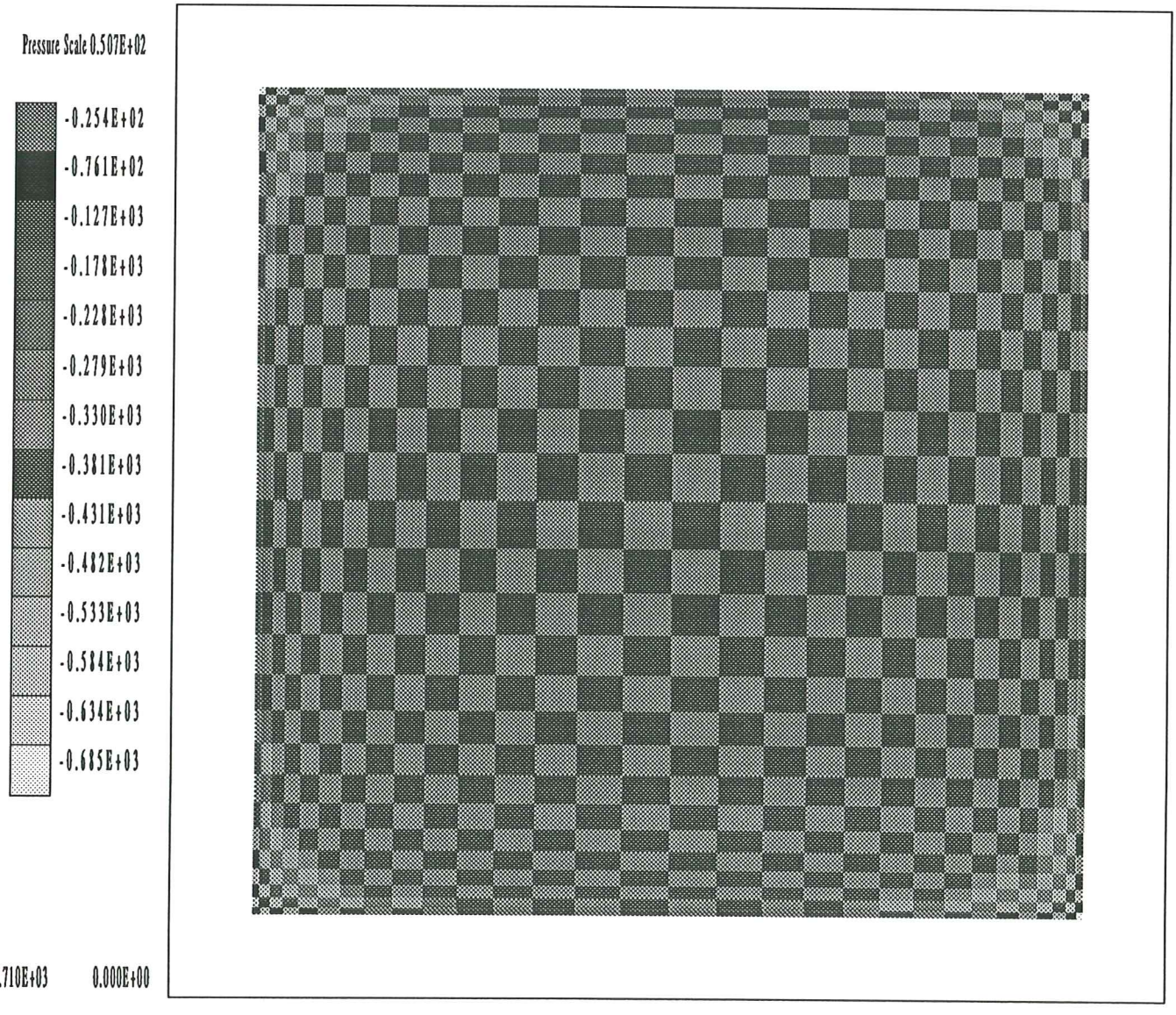

Pressure Scale 0.507E+02
-0.254E+02
-0.761E+02
-0.127E+03
-0.178E+03
-0.228E+03
-0.279E+03
-0.330E+03
-0.381E+03
-0.431E+03
-0.482E+03
-0.533E+03
-0.584E+03
-0.634E+03
-0.685E+03
-0.710E+03
0.000E+00

Fig 3

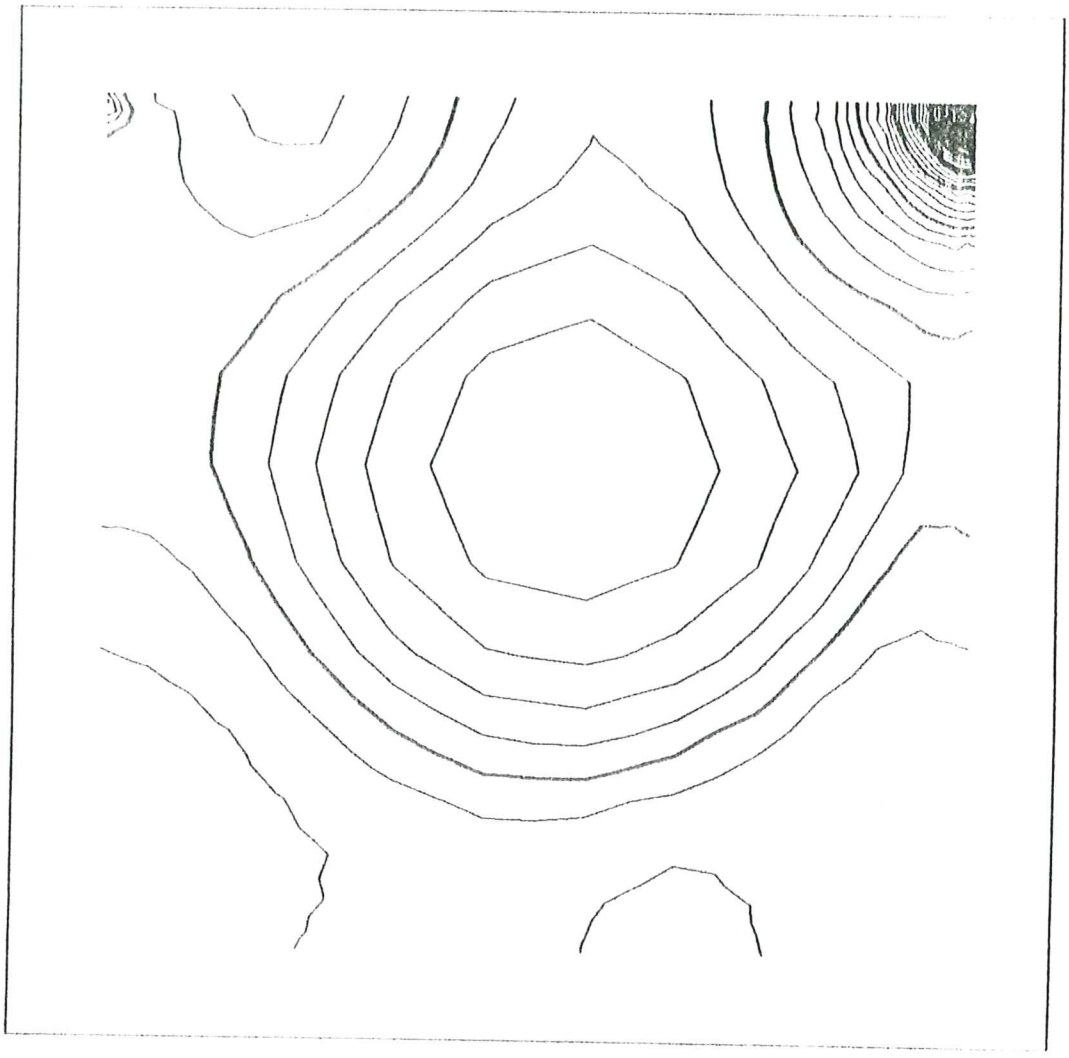


Fig. 4

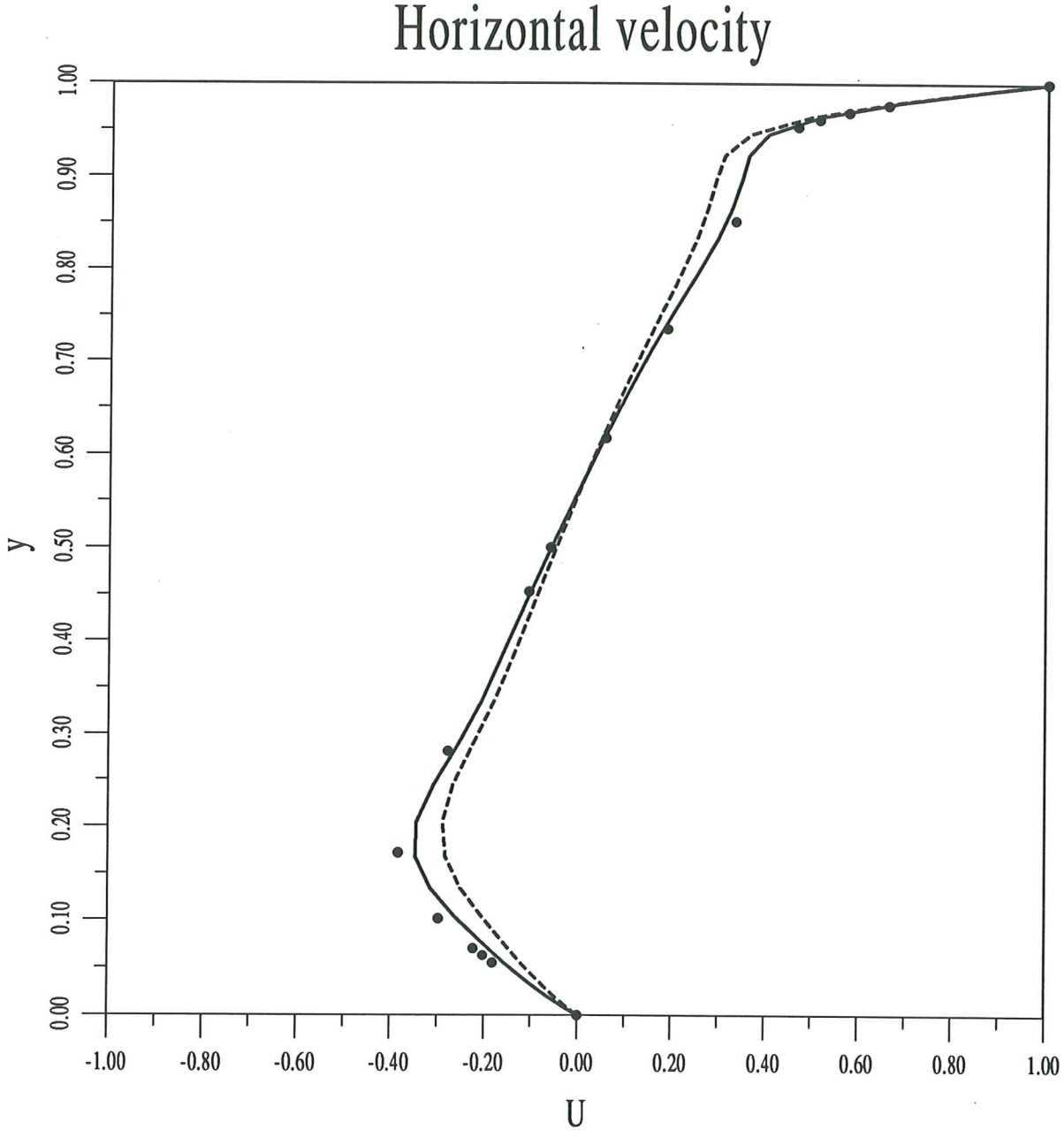

Horizontal velocity
y
U
0.00
0.10
0.20
0.30
0.40
0.50
0.60
0.70
0.80
0.90
1.00
-1.00
-0.80
-0.60
-0.40
-0.20
0.00
0.20
0.40
0.60
0.80
1.00

Fig. 5a

Contour Interval=0.1126E-01

contour levels

1 -0.1141E-01

2 -0.1430E-03

3 0.1112E-01

4 0.2238E-01

5 0.3365E-01

6 0.4491E-01

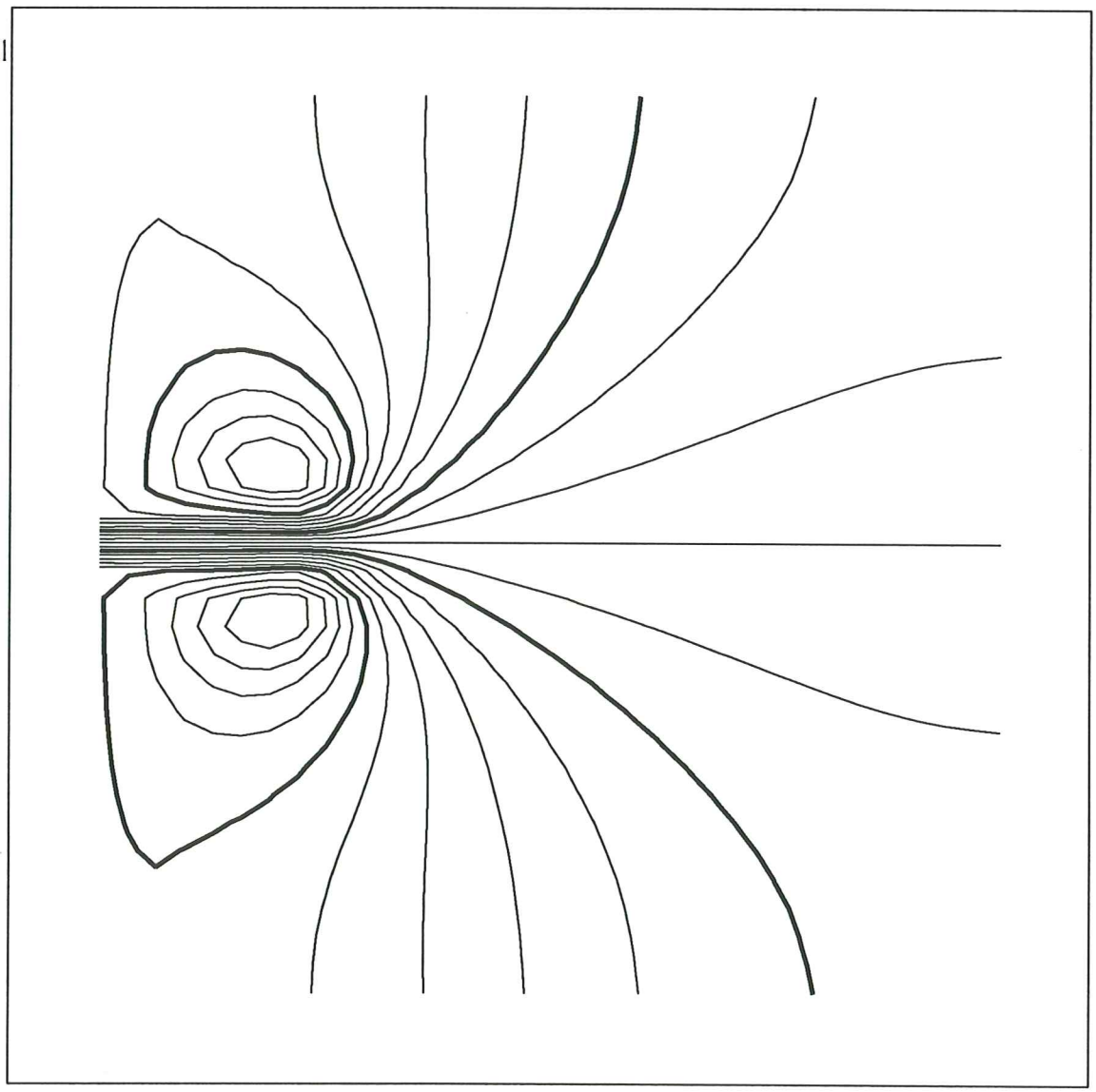


Fig 5b

Contour Interval=0.1504E-01

	contour levels
1	-0.2047E-01
2	-0.5433E-02
3	0.9609E-02
4	0.2465E-01
5	0.3969E-01
6	0.5473E-01

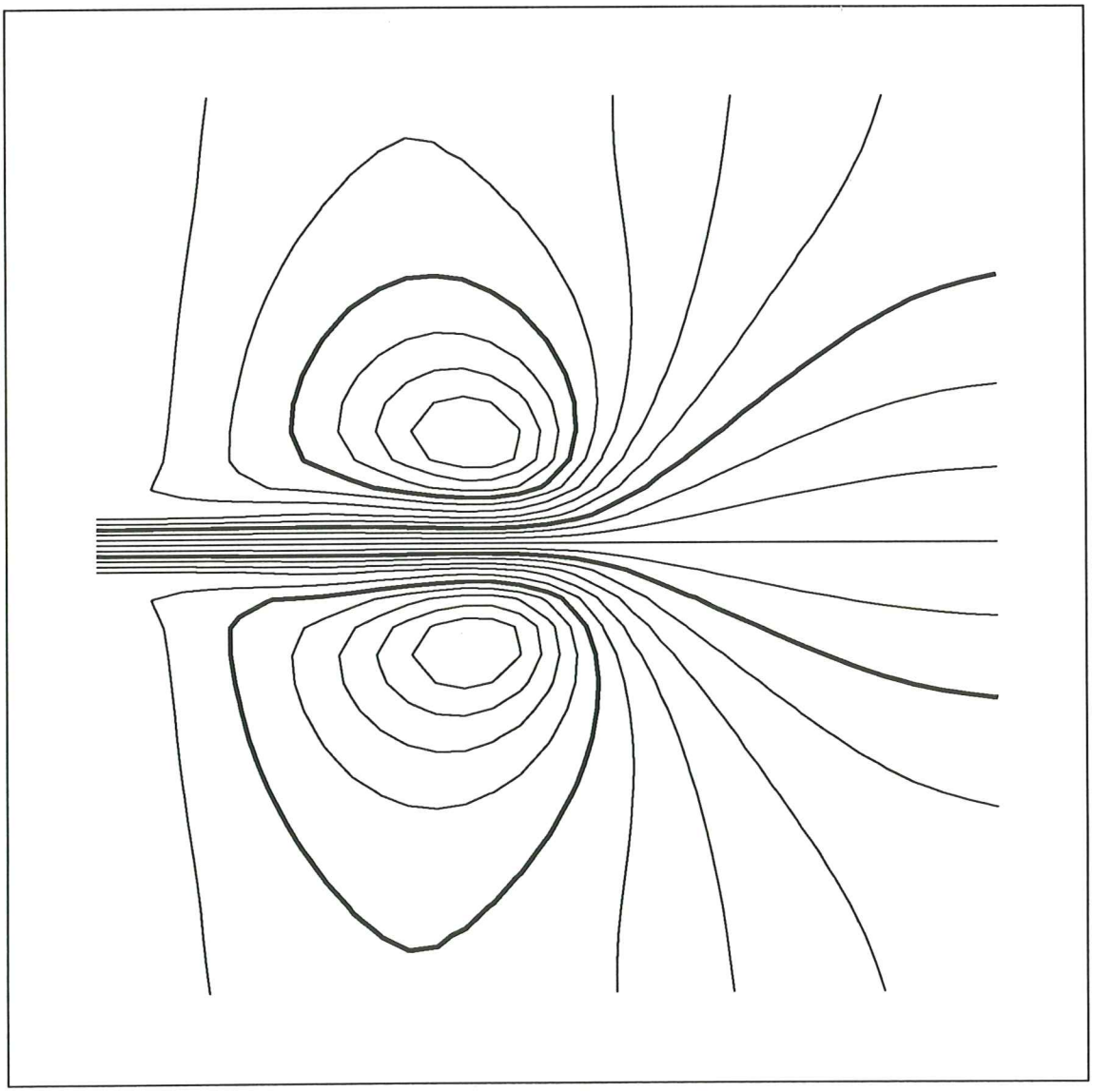


Fig 5c

Contour Interval=0.1855E-01

contour levels

1 -0.2889E-01

2 -0.1034E-01

3 0.8205E-02

4 0.2675E-01

5 0.4530E-01

6 0.6385E-01

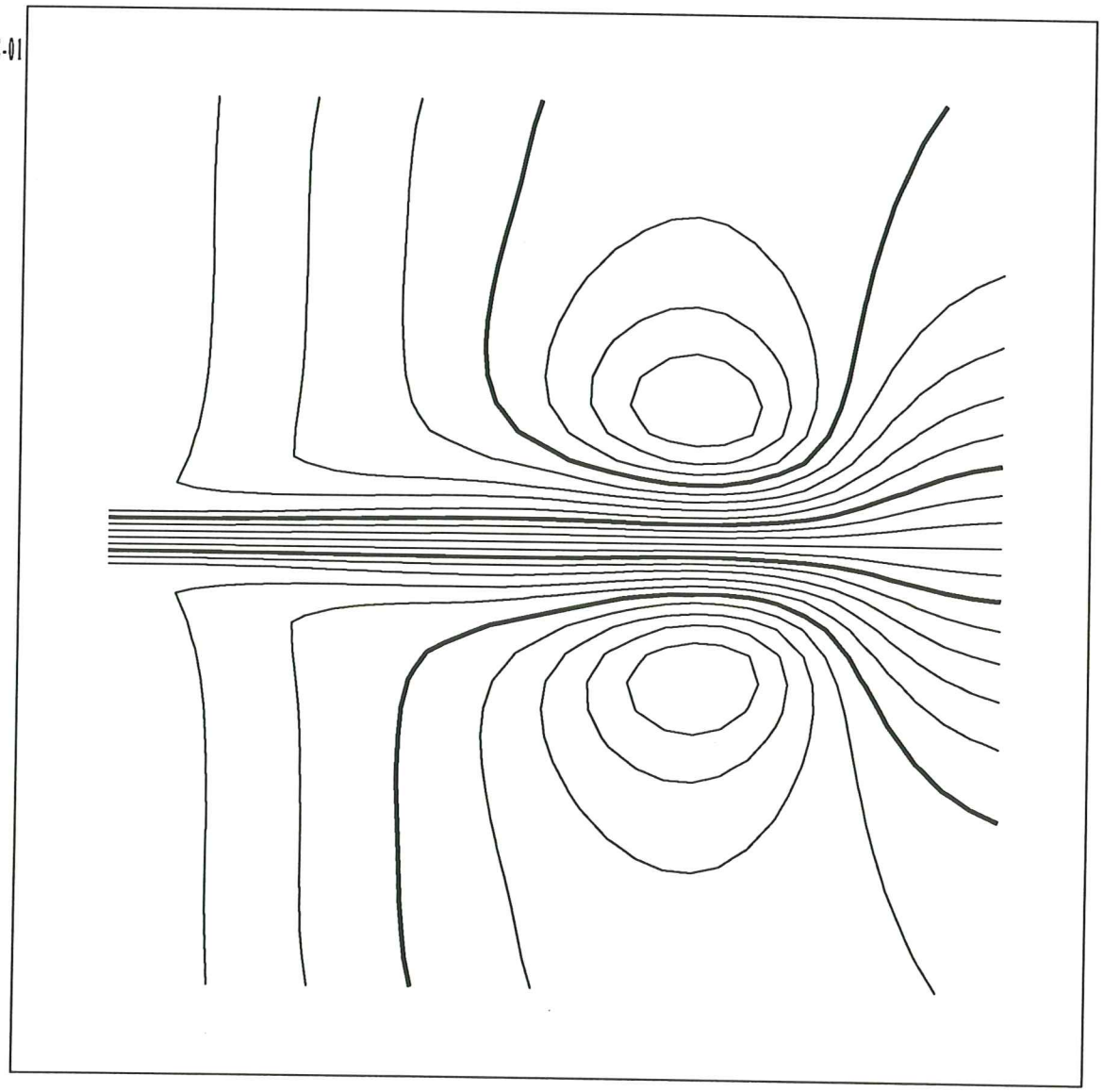


Fig. 5d.

Contour Interval=0.8728E-02

contour levels

1 -0.1003E-01

2 -0.1306E-02

3 0.7422E-02

4 0.1615E-01

5 0.2488E-01

6 0.3361E-01

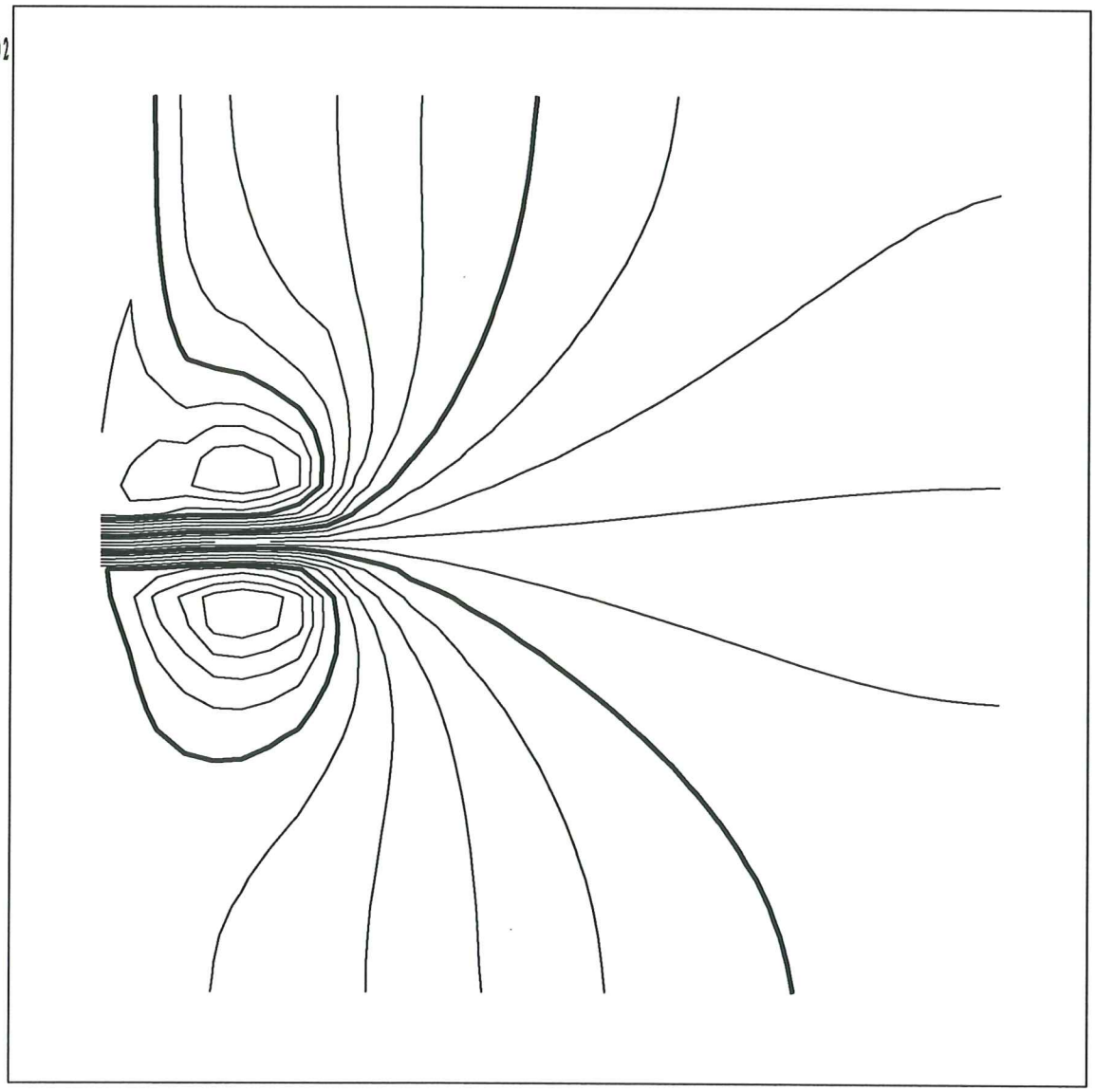


Fig 5e

Contour Interval=0.1169E-01

contour levels

1	-0.1745E-01
2	-0.5751E-02
3	0.5943E-02
4	0.1764E-01
5	0.2933E-01
6	0.4103E-01

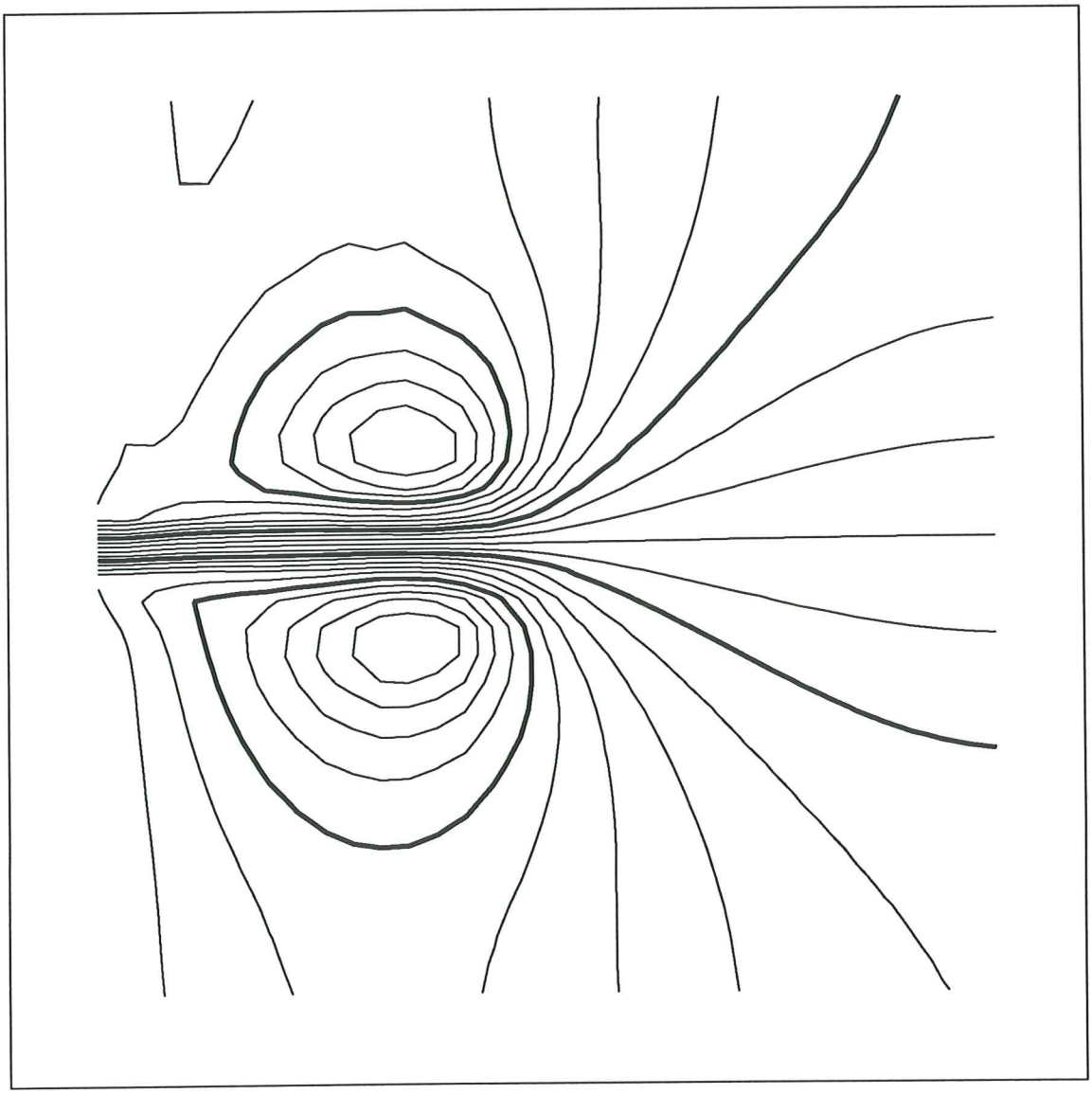


Fig 58

Contour Interval=0.1435E-01

contour levels

1 -0.2390E-01

2 -0.9555E-02

3 0.4791E-02

4 0.1914E-01

5 0.3348E-01

6 0.4783E-01

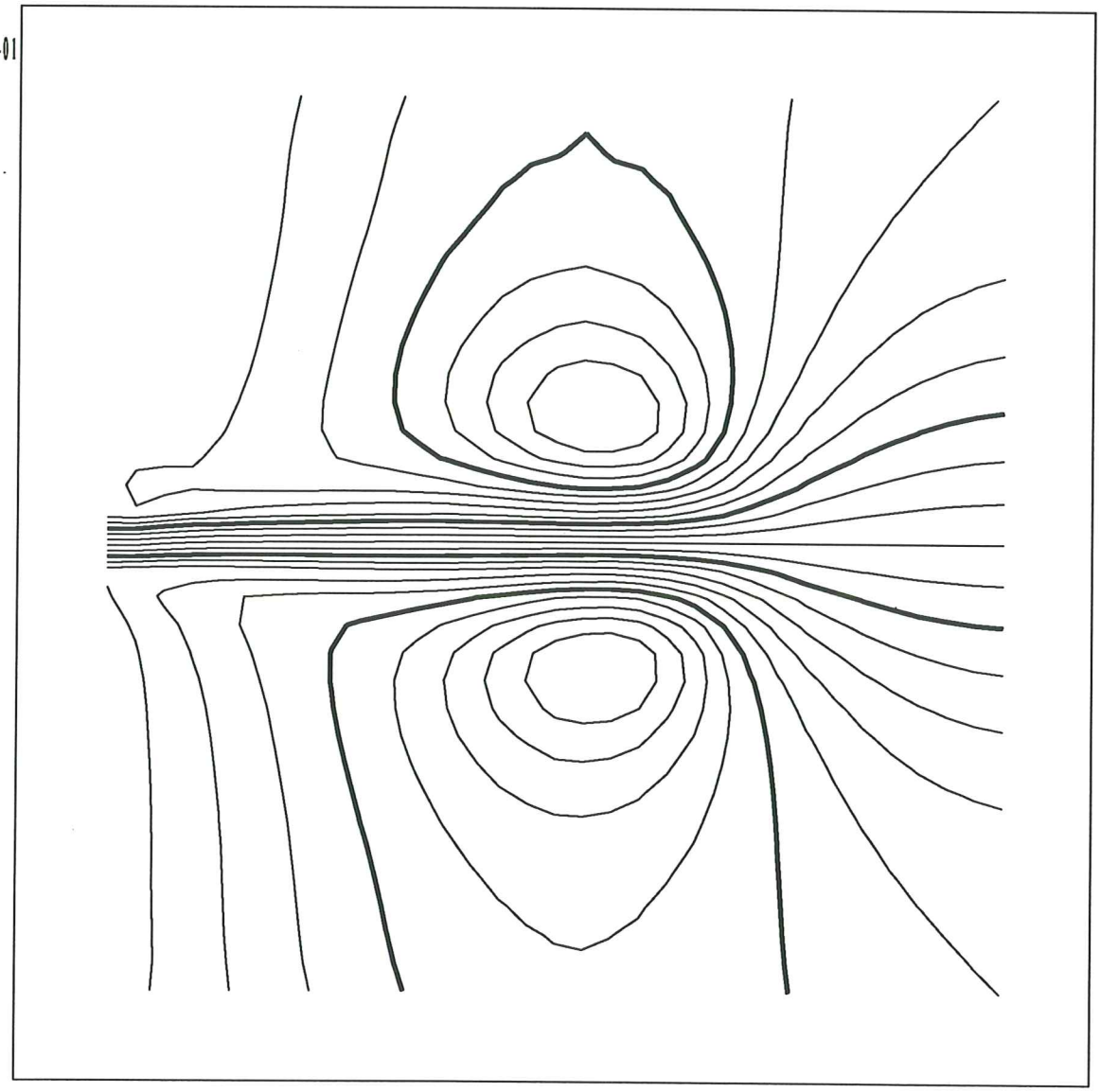


Fig 6a

Contour Interval=0.3283E-01

contour levels

1 -0.6393E-01

2 -0.3110E-01

3 0.1735E-02

4 0.3457E-01

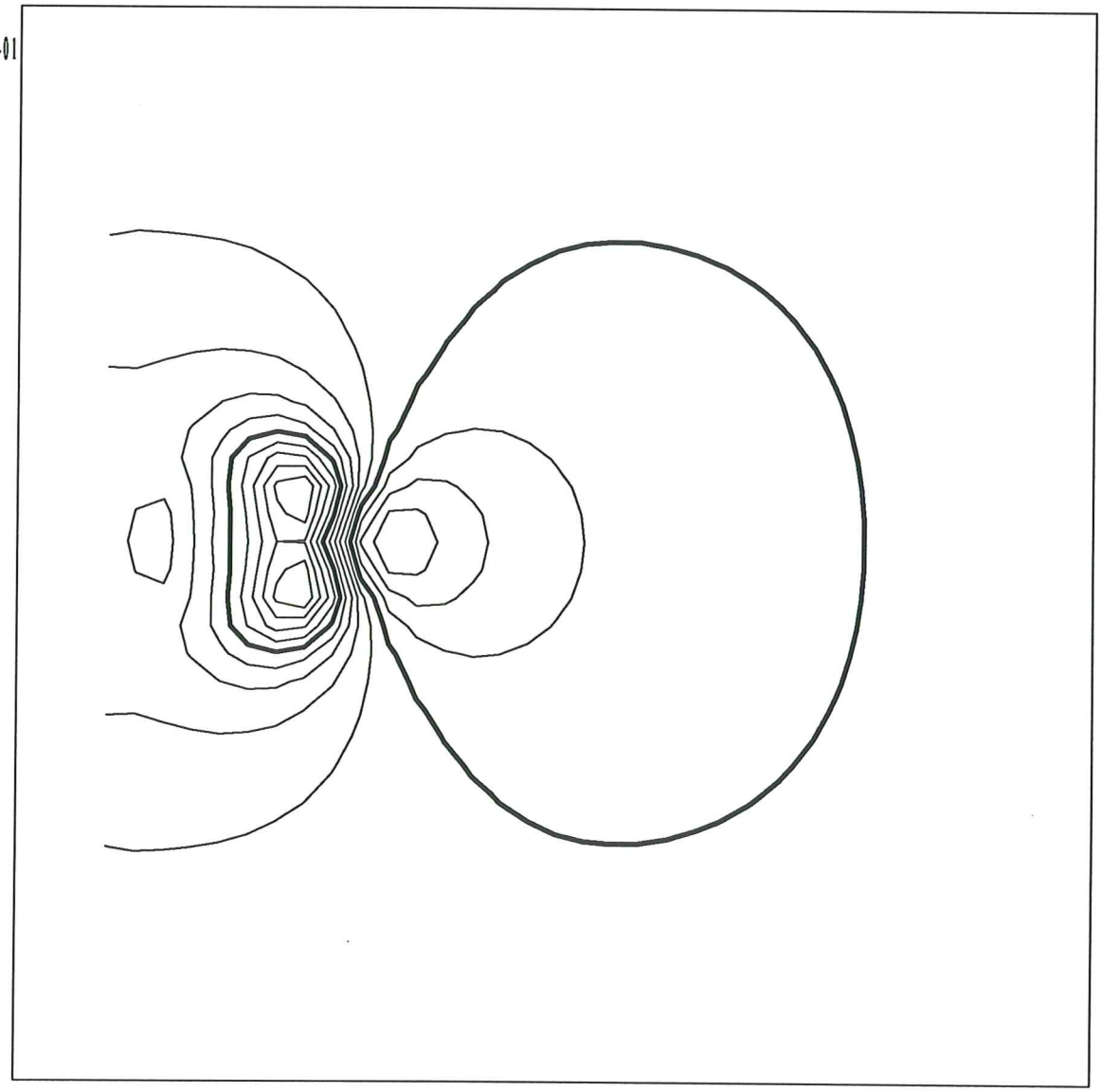


Fig. 6b

Contour Interval=0.2690E-01

contour levels

1 -0.5530E-01

2 -0.2839E-01

3 -0.1487E-02

4 0.2542E-01

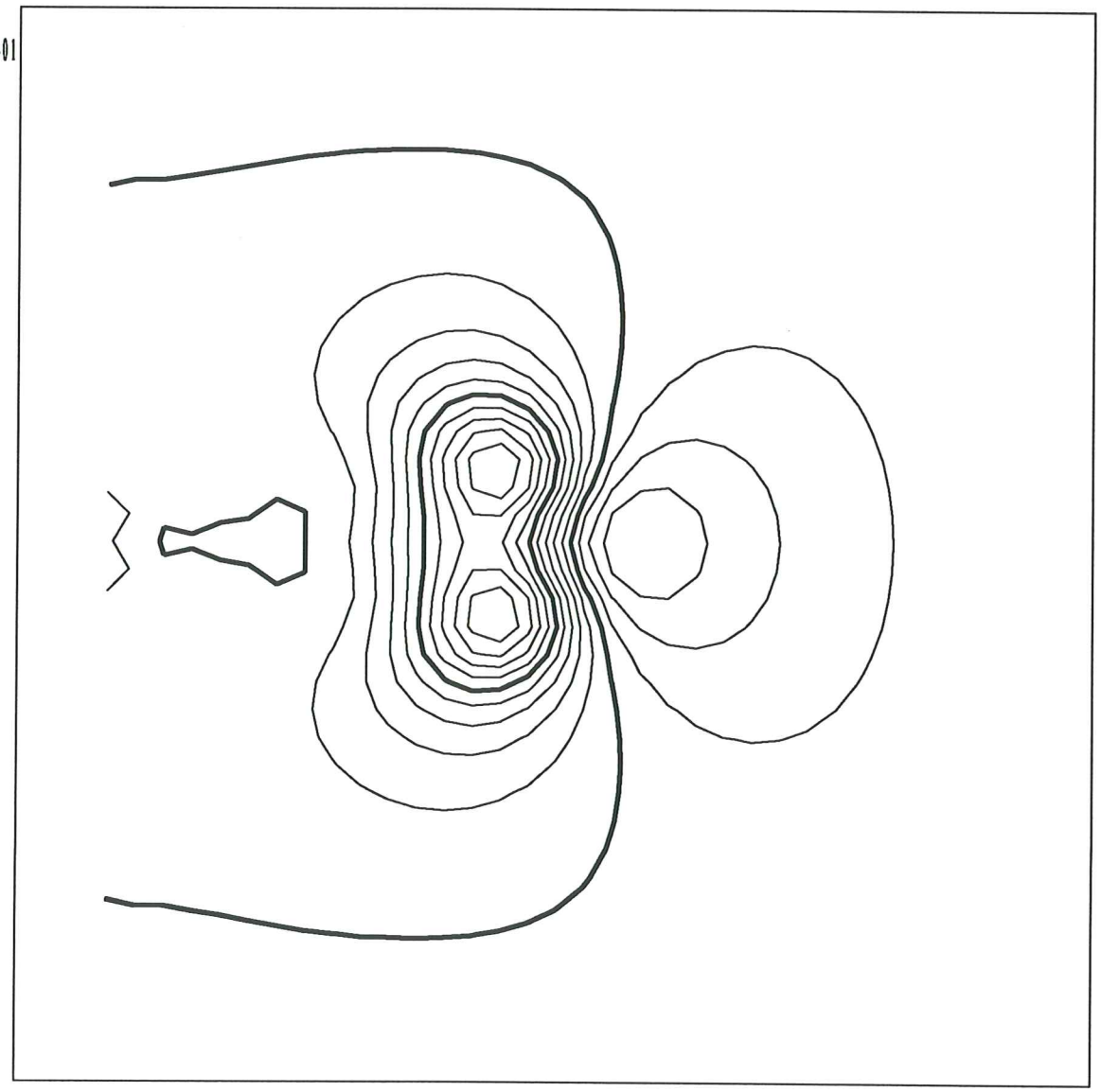


Fig. 6c.

Contour Interval=0.1989E-01

contour levels

1 -0.4742E-01

2 -0.2753E-01

3 -0.7637E-02

4 0.1225E-01

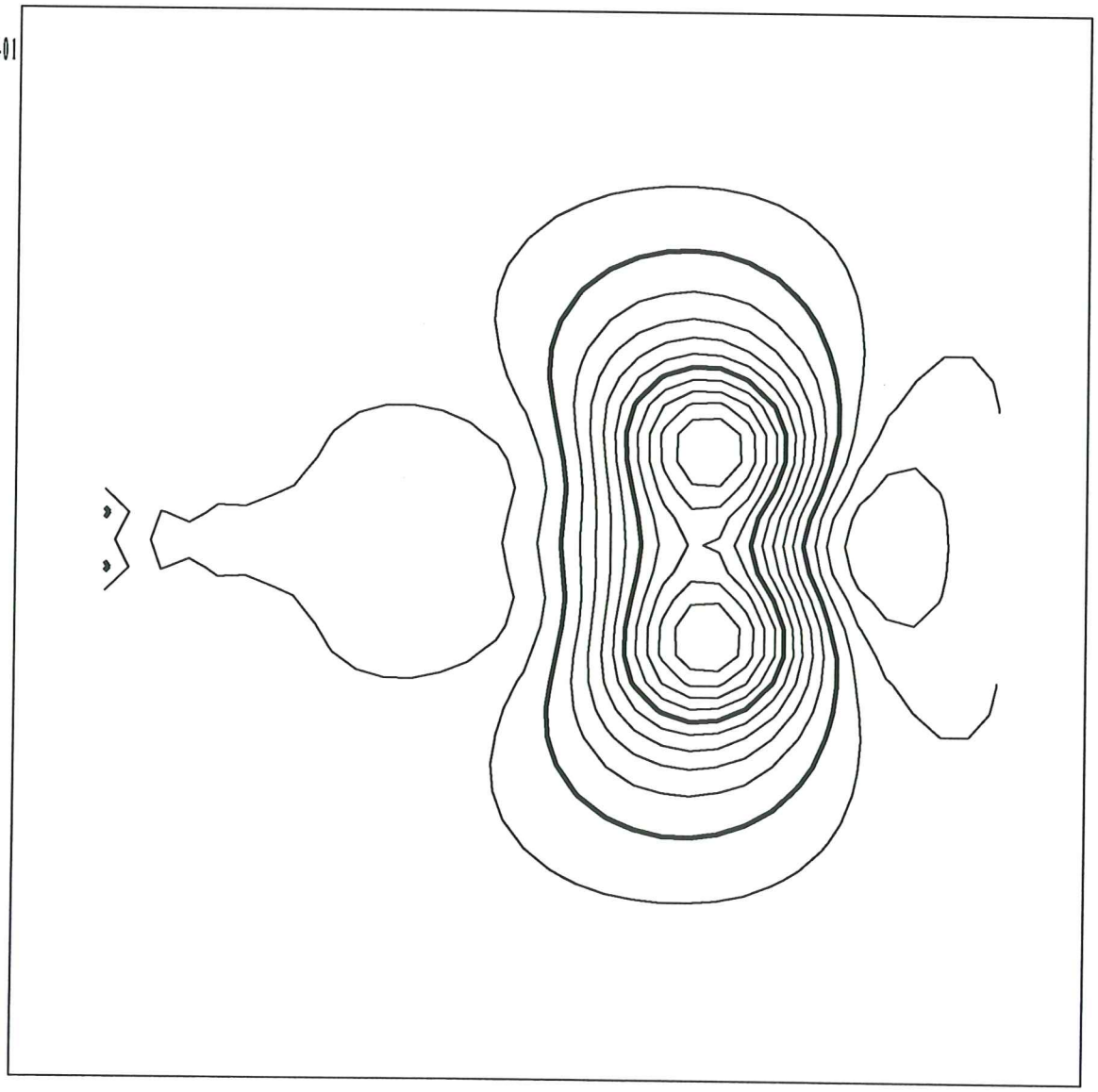


Fig. 6d.

Contour Interval=0.2366E-01

contour levels

1 -0.4307E-01

2 -0.1940E-01

3 0.4260E-02

4 0.2793E-01

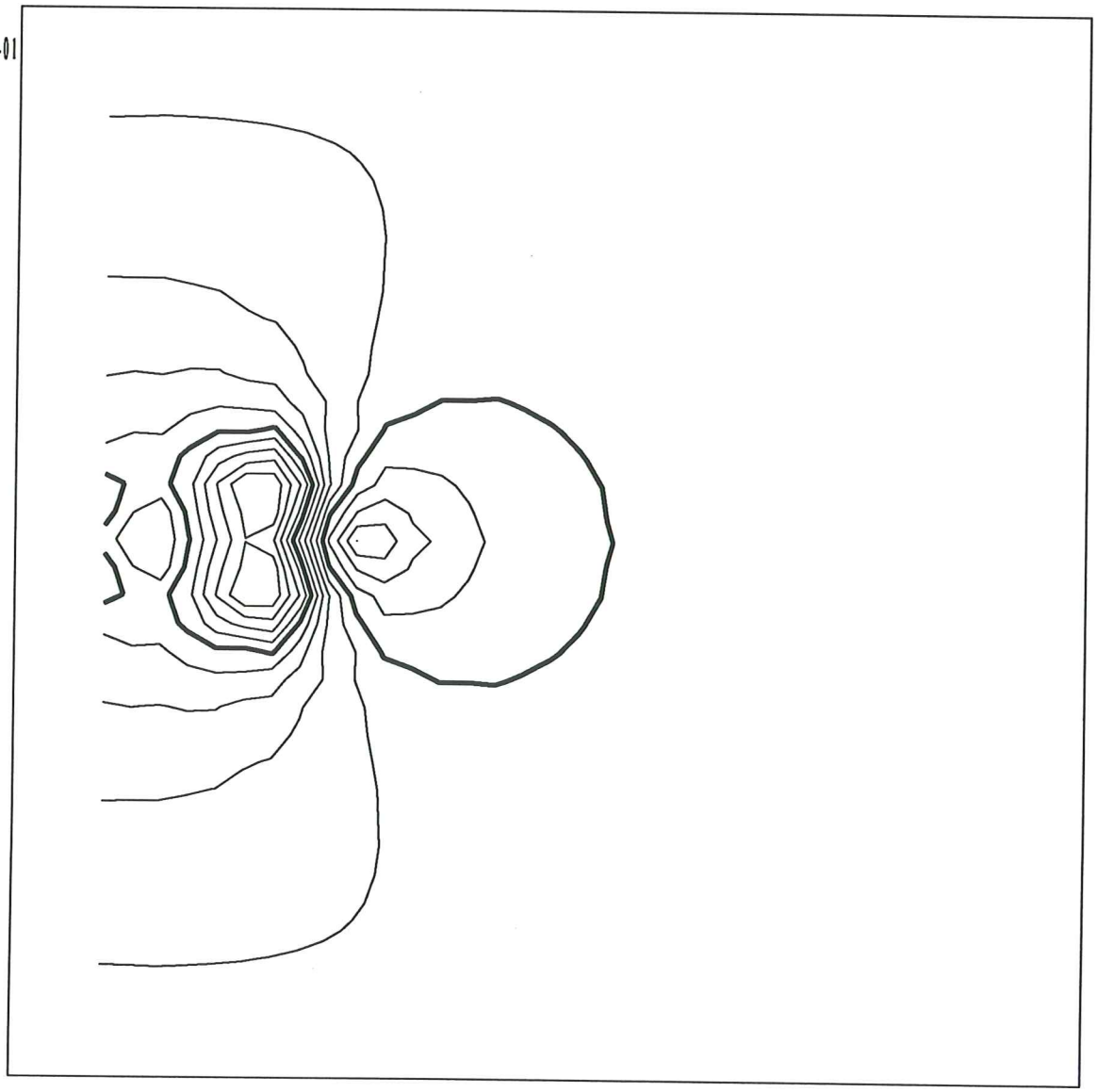


Fig. 6e

Contour Interval=0.1965E-01

contour levels

1 -0.3847E-01

2 -0.1882E-01

3 0.8334E-03

4 0.2048E-01

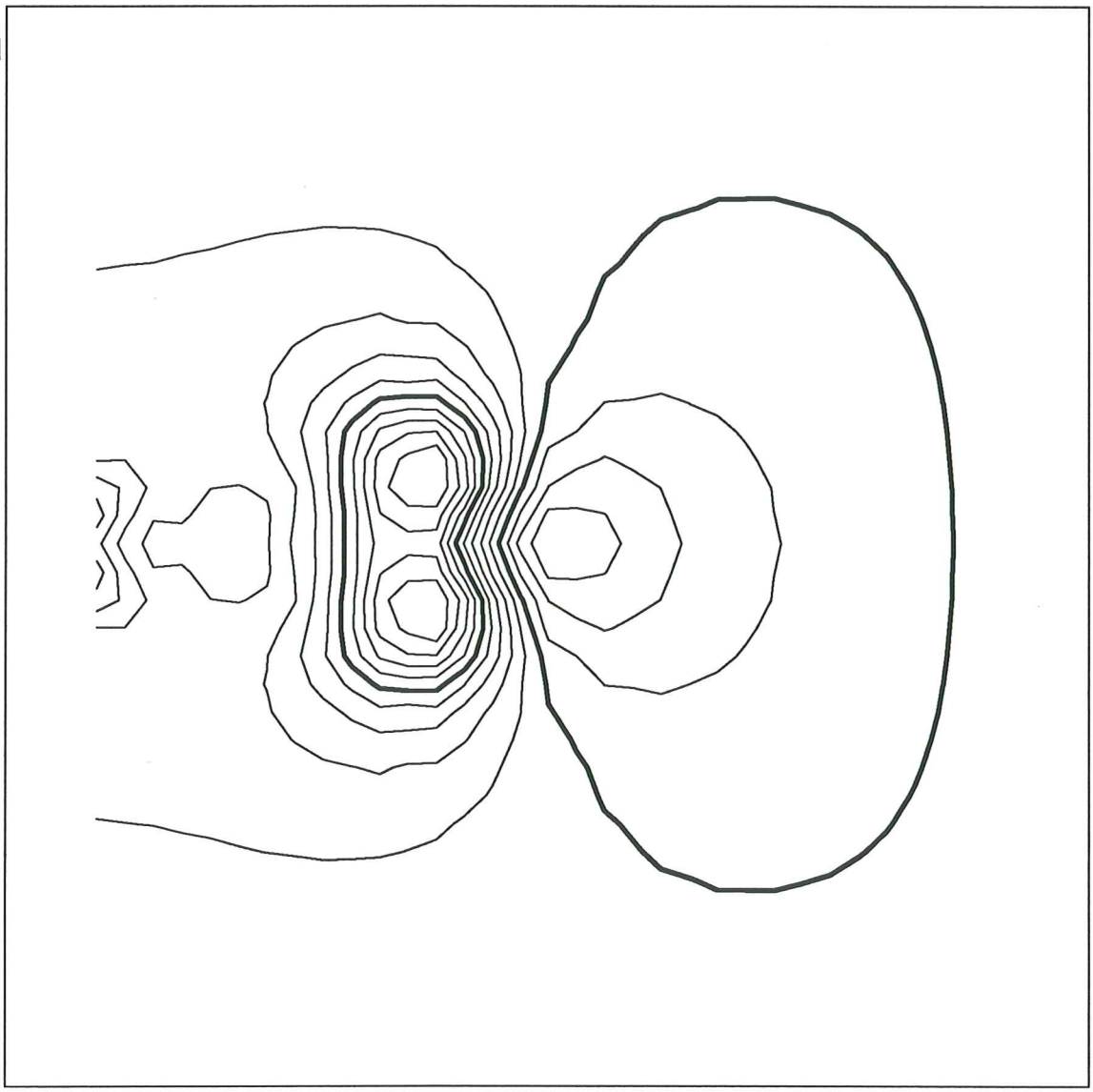


Fig. 68.

Contour Interval=0.1650E-01

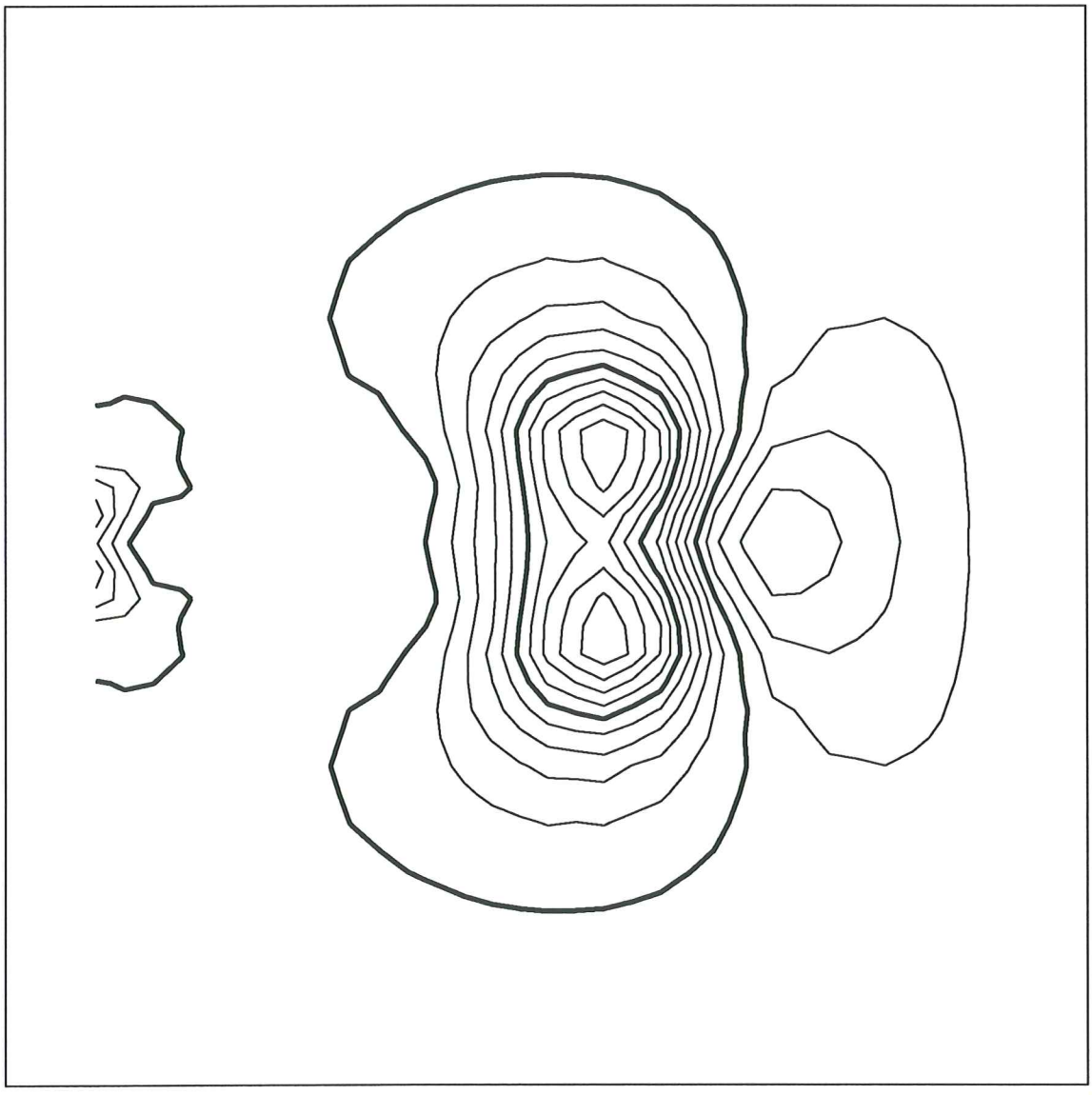
contour levels

1 -0.3512E-01

2 -0.1862E-01

3 -0.2117E-02

4 0.1439E-01



A STRATEGY TO COMBINE OPTIMUM STRUCTURAL SHAPE DESIGN WITH AUTOMATIC MESH ADAPTION

G. BUGEDA and E. OÑATE

CIMNE; Universitat Politècnica de Catalunya
Módul C1; Campus Nord UPC; Gran Capità s/n;08034 Barcelona;Spain

SUMMARY

This paper presents a methodology based on the use of automatic adaptive mesh refinement (AMR) techniques in the context of structural shape optimization problems analyzed by the Finite Element Method (FEM). The computation of the sensitivities of the finite element mesh coordinates and the error estimator allow their projection from one design to the next, giving an "a priori knowledge" of the error distribution on the new design. This allows to build up a finite element mesh for the new design with a specified and controlled level of error. The robustness and reliability of the proposed methodology is checked out with the optimization of the central section of a double curvature arch dam.

1. INTRODUCTION

Extensive research has been performed in recent years to develop efficient methodologies and codes for optimization of structural shapes using the finite element method. However, some problems still remain unsolved in this context, e.g. the inclusion of robust parametrization procedures for definition of each design and the control of the error associated with finite element computations and its influence on the solution of the optimization problem. Usually, once the optimization process is finished there is no guarantee of the accuracy of the final design. Sometimes a more accurate analysis would reveal that the final design is unfeasible, as one or more of the constraints imposed are violated.

With a view towards solving this problem a general finite element methodology for structural shape optimization problems should include the following features:

- general parametrization procedures.
- easy treatment of boundary conditions.
- easy and general definition of the objective function and constraints.
- automatic, robust and flexible mesh generation.
- accurate and inexpensive estimation of the discretization errors.
- effective, reliable and not too expensive sensitivity analysis.
- efficient optimization procedures.
- Cost-efficient automatic adaptive remeshing procedures.
- control over the quality of the meshes used for each design.

In this paper we present a general methodology for optimization problems including all the above mentioned features. In the following sections we describe the parametrization of the optimum design problem, the procedure for automatic mesh generation and error estimation, the

$$f(\boldsymbol{x}^{k+1}) = f(\boldsymbol{x}^k + \theta^k \boldsymbol{s}^k) \approx f(\boldsymbol{x}^k) + \theta^k \frac{\partial f}{\partial \boldsymbol{s}^k} \tag{1}$$

3. PARAMETRIZATION OF THE PROBLEM

Each design geometry is represented by using "definition points" which specify some interpolation curves. The curves used here are parametric B-splines. The general expression of a closed B-spline for q points is [3]:

$$\boldsymbol{r}(t) = \sum_{l=0}^{q} \boldsymbol{r}_l N_{4,l+1}(t) \tag{2}$$

where $\boldsymbol{r}(t)$ is the position vector depending on a parametric variable t (see Figure 2). The curve is expressed as a linear combination of $q+1$ normalized fourth order (cubic) B-splines [3]. The $\boldsymbol{r}_l$ coefficients are the coordinates of the so called polygon definition points. By using eq. (2), the coordinates of the definition points and some additional conditions about slopes and curvatures, the following equation system can be derived:

$$\boldsymbol{v} = \boldsymbol{N}\boldsymbol{r}^g \tag{3}$$

where $\boldsymbol{v}$ is a vector containing the imposed conditions at the definition points, $\boldsymbol{N}$ is a matrix containing some terms corresponding to the values of the polynomials that define each B-spline, and the $\boldsymbol{r}^g$ vector contains the coefficients $\boldsymbol{r}_i$ to be computed. Details of this process can be found in [3].

Figure 2. Definition and polygon definition points of a B-spline.

The first order sensitivities of $\boldsymbol{r}^g$ along a direction $\boldsymbol{s}$ in the design variable space are given by:

$$\frac{\partial \boldsymbol{r}^g}{\partial \boldsymbol{s}} = \boldsymbol{N}^{-1}\left(\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{s}} - \frac{\partial \boldsymbol{N}}{\partial \boldsymbol{s}}\boldsymbol{r}\right) \tag{4}$$

design variables space, $\boldsymbol{s}$, are obtained by differentiating eq. (6) with respect to $\boldsymbol{s}$ for each cycle, i.e.

$$\frac{\partial \boldsymbol{r}_i}{\partial \boldsymbol{s}} = \frac{\sum_{j=1}^{m_i} \frac{\partial \boldsymbol{r}_j}{\partial \boldsymbol{s}}}{m_i} \tag{7}$$

5. FINITE ELEMENT ANALYSIS

In this paper we will consider only the solution of structural problems governed by the standard elliptic equations of elasticity with appropriate boundary conditions [7]:

$$\boldsymbol{Lu} \equiv \boldsymbol{S}^T \boldsymbol{DSu} = \boldsymbol{b} \text{ in } \Omega \tag{8}$$

Discretization of eq. (8) leads to the standard linear system of equations:

$$\boldsymbol{Ka} = \boldsymbol{q} \quad with \quad \begin{cases} \boldsymbol{K} = \sum_e \boldsymbol{K}_e \\ \boldsymbol{K}_e = \int_{\Omega_e} \boldsymbol{B}^T \boldsymbol{DB} d\Omega \\ \boldsymbol{q} = \sum_e \boldsymbol{q}_e \\ \boldsymbol{q}_e = \boldsymbol{q}_{\Omega_e} + \boldsymbol{q}_{\Sigma_e} + \boldsymbol{q}_{P_e} \\ \boldsymbol{q}_\Omega = \int_{\Omega_e} \boldsymbol{N}^T \boldsymbol{b} d\Omega \\ \boldsymbol{q}_{\Gamma_e} = \int_{\Gamma_e} \boldsymbol{N}^T \boldsymbol{t} d\Gamma \\ \boldsymbol{q}_{P_e} = \sum \boldsymbol{N}^T \boldsymbol{p} \end{cases} \tag{9}$$

where $\boldsymbol{K}$, $\boldsymbol{a}$ and $\boldsymbol{q}$ denote, as usual, the stiffness matrix, the nodal displacement and the equivalent nodal force vectors. Vectors $\boldsymbol{b}$, $\boldsymbol{t}$ and $\boldsymbol{p}$ correspond to the body, surface and point loads respectively. Matrix $\boldsymbol{B} = \boldsymbol{SN}$ is used to obtain the strains at each point as $\boldsymbol{\varepsilon} = \boldsymbol{Ba}$ and the constitutive matrix $\boldsymbol{D}$ relates strains with stresses as $\boldsymbol{\sigma} = \boldsymbol{D\varepsilon}$.

In our work, nodal stresses $\bar{\boldsymbol{\sigma}}^*$ are recovered using a global least squares smoothing technique [8]:

$$\begin{cases} \boldsymbol{\sigma}^* = \sum N_i \bar{\boldsymbol{\sigma}}_i^* = \bar{\boldsymbol{N}}^T \bar{\boldsymbol{\sigma}}^* \\ \bar{\boldsymbol{\sigma}}^* = \boldsymbol{M}^{-1} \boldsymbol{\Phi} \\ \boldsymbol{\Phi} = \sum_e \int_{\Omega_e} \bar{\boldsymbol{N}}^T \boldsymbol{\sigma} d\Omega \\ \boldsymbol{M} = \sum_e \boldsymbol{M}_e \\ \boldsymbol{M}_e = \int_{\Omega_e} \bar{\boldsymbol{N}} \bar{\boldsymbol{N}}^T d\Omega \end{cases} \tag{10}$$

6. ERROR ESTIMATION

The error associated with each finite element solution is evaluated for each element using the Zienkiewicz and Zhu[8] error estimator as:

$$\|\boldsymbol{e}\|_{E_e}^2 \approx \eta_e^2 = \int_{\Omega_e} (\boldsymbol{\sigma}^* - \boldsymbol{\sigma})^T \boldsymbol{D}^{-1} (\boldsymbol{\sigma}^* - \boldsymbol{\sigma}) d\Omega \tag{11}$$

This technique allows to obtain first-order and higher-order sensitivities of the stiffness matrix $\boldsymbol{K}$, the nodal forces vector $\boldsymbol{q}$ and of any other integral expression involved in the computations. The detailed expressions for the first-order and higher-order sensitivity analysis can be found in [2,4].

Eq. (13) allows to obtain the first order sensitivities of the displacement vector $\boldsymbol{a}$ as:

$$\frac{\partial \boldsymbol{a}}{\partial \boldsymbol{s}} = \boldsymbol{K}^{-1}\left[\frac{\partial \boldsymbol{q}}{\partial \boldsymbol{s}} - \frac{\partial \boldsymbol{K}}{\partial \boldsymbol{s}}\boldsymbol{a}\right] \tag{15}$$

Eqs. (15) show that the inverse of the stiffness matrix is needed for the sensitivity computations. If a direct solver is used this matrix has already been factorized and each new sensitivity analysis involves only a new backsubstitution process. Moreover it is not necessary to assemble the sensitivities of the stiffness matrix because they always appear multiplying a vector and these products can be computed in an element-by-element manner.

The strain and stress sensitivities can be computed as:

$$\frac{\partial \boldsymbol{\varepsilon}}{\partial \boldsymbol{s}} = \frac{\partial \boldsymbol{B}}{\partial \boldsymbol{s}}\boldsymbol{a} + \boldsymbol{B}\frac{\partial \boldsymbol{a}}{\partial \boldsymbol{s}} \quad , \quad \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{s}} = \frac{\partial \boldsymbol{D}}{\partial \boldsymbol{s}}\boldsymbol{\varepsilon} + \boldsymbol{D}\frac{\partial \boldsymbol{\varepsilon}}{\partial \boldsymbol{s}} \tag{16}$$

The sensitivities of the smoothed stresses are computed in terms of the sensitivities of the mass matrix $\boldsymbol{M}$ and the $\boldsymbol{\Phi}$ vector of eq. (10). The techniques discussed above for the integral expressions are also used to compute these sensitivities. Finally, the sensitivities of the smoothed stresses are obtained as:

$$\frac{\partial \bar{\boldsymbol{\sigma}}^*}{\partial \boldsymbol{s}} = \boldsymbol{M}^{-1}\left[\frac{\partial \boldsymbol{\Phi}}{\partial \boldsymbol{s}} - \frac{\partial \boldsymbol{M}}{\partial \boldsymbol{s}}\bar{\boldsymbol{\sigma}}^*\right] \quad , \quad \frac{\partial \boldsymbol{\sigma}^*}{\partial \boldsymbol{s}} = \bar{\boldsymbol{N}}^T\frac{\partial \bar{\boldsymbol{\sigma}}^*}{\partial \boldsymbol{s}} \tag{17}$$

The same comments about the factorization of the stiffness matrix apply now to the mass matrix.

Following a similar procedure the first-order sensitivity of the error estimator is obtained from eq. (11) as:

$$\begin{aligned}\frac{\partial \eta_e^2}{\partial \boldsymbol{s}} = \int_{\Omega_{\boldsymbol{\xi}}} \Bigg[& \left(\frac{\partial \boldsymbol{\sigma}^*}{\partial \boldsymbol{s}} - \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{s}}\right)^T \boldsymbol{D}^{-1}(\boldsymbol{\sigma}^* - \boldsymbol{\sigma})|\boldsymbol{J}| + (\boldsymbol{\sigma}^* - \boldsymbol{\sigma})^T\frac{\partial \boldsymbol{D}^{-1}}{\partial \boldsymbol{s}}(\boldsymbol{\sigma}^* - \boldsymbol{\sigma})|\boldsymbol{J}| \\ & + (\boldsymbol{\sigma}^* - \boldsymbol{\sigma})^T\boldsymbol{D}^{-1}\left(\frac{\partial \boldsymbol{\sigma}^*}{\partial \boldsymbol{s}} - \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{s}}\right)|\boldsymbol{J}| + (\boldsymbol{\sigma}^* - \boldsymbol{\sigma})^T\boldsymbol{D}^{-1}(\boldsymbol{\sigma}^* - \boldsymbol{\sigma})\frac{\partial |\boldsymbol{J}|}{\partial \boldsymbol{s}}\Bigg] d\xi_1 d\xi_2\end{aligned} \tag{18}$$

In order to use an adaptive mesh refinement strategy it is also necessary to compute the element and total strain energy. The values of this strain

energy and its first order sensitivity can be approximated from the finite element solution as:

$$\|\boldsymbol{u}\|^2_{E_e} \approx \boldsymbol{a}^T \boldsymbol{K}_e \boldsymbol{a} + \eta_e^2 \tag{19}$$

$$\frac{\partial \|\boldsymbol{u}\|^2_{E_e}}{\partial \boldsymbol{s}} \approx \frac{\partial \boldsymbol{a}^T}{\partial \boldsymbol{s}} \boldsymbol{K}_e \boldsymbol{a} + \boldsymbol{a}^T \frac{\partial \boldsymbol{K}_e}{\partial \boldsymbol{s}} \boldsymbol{a} + \boldsymbol{a}^T \boldsymbol{K}_e \frac{\partial \boldsymbol{a}}{\partial \boldsymbol{s}} + \frac{\partial \eta_e^2}{\partial \boldsymbol{s}} \tag{20}$$

Should the second order sensitivities of above variables be required it can be computed following similar procedures as shown in [2].

8. DESIGN ENHANCEMENT

The objective function sensitivities are used to get improved values of the design variables by means of a minimization method. Depending on the optimization algorithm it may be necessary to use second order sensitivities. The design variables corresponding to the improved design are found in terms of an advance parameter θ^k as:

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \theta^k \boldsymbol{s}^k \tag{21}$$

The direction of change $\boldsymbol{s}^k$ has been obtained here using a BFGS Quasi-Newton method which only requires first order sensitivities of the objective function. The value of θ^k is obtained by a directional second order sensitivity analysis in the $\boldsymbol{s}^k$ direction. The expressions of the second order sensitivities of the relevant variables can be found in [2,4]. The objective function f can be approximated along this direction using a second order Taylor expansion similar to eq. (1) which minimization provides the value of θ^k.

9. PROJECTION TO THE NEXT DESIGN AND DEFINITION OF THE NEW MESH

Once the new design has been defined, the new values of the error estimator, the "energy" and the coordinates of the mesh can be simply projected from the previous solution into the next design as:

$$\begin{cases} (x,y)^{k+1} = (x,y)^k + \theta^k \left(\frac{\partial x}{\partial \boldsymbol{s}}, \frac{\partial y}{\partial \boldsymbol{s}} \right) \\ \eta^{k+1^k} = \eta^{k^2} + \theta^k \frac{\partial \|e\|^2}{\partial \boldsymbol{s}} \\ \|\boldsymbol{u}\|^{k+1^2}_E = \|\boldsymbol{u}\|^{k^2}_E + \theta^k \frac{\partial \|\boldsymbol{u}\|^2_E}{\partial \boldsymbol{s}} \end{cases} \tag{22}$$

The projected values provide the necessary information to perform a remeshing over the next design, even before any new computation is performed. In that sense, the error estimator computed "a posteriori" is transformed into an "a priori" error estimator.

This projection allows the quality control of the mesh for each design prior to any new computation. The projected values are used to create the background mesh information needed to generate the mesh for the new design. This closes the iterative process which will lead to the "enhanced" optimum design after convergence. Of course second order projection can be used as described in [2].

The generation of every new mesh in the remeshing procedure requires the definition of a "mesh optimality criterion". In this work a mesh is considered as optimal when the error density is equally distributed across the volume, i.e. when $\|\boldsymbol{e}\|_e^2/\Omega_e = \|\boldsymbol{e}\|^2/\Omega$ is satisfied. The justification of this mesh optimality criterion can be found in [5].

The combination of the mesh optimality criterion and the error estimation allows to define the new element sizes. Previously, it is necessary to define the allowable global error percentage γ as:

$$\gamma = 100\frac{\|\boldsymbol{e}\|}{\|\boldsymbol{u}\|} \approx 100\frac{\eta}{\sqrt{\eta^2 + (\boldsymbol{a}^T\boldsymbol{K}\boldsymbol{a})^2}} \tag{23}$$

The target error level for each element is:

$$\|\boldsymbol{e}\|_e^t = \frac{\gamma}{100}\|\boldsymbol{u}\|\sqrt{\frac{\Omega_e}{\Omega}} \tag{24}$$

The new element sizes $\bar{h}_e$ can be computed in terms of the old sizes h_e and the order of the shape function polynomials p using the expression:

$$\bar{h}_e = \frac{h_e}{\xi_e^{1/p}} \qquad with \qquad \xi_e = \frac{\|\boldsymbol{e}\|_e}{\|\boldsymbol{e}\|_e^t} \tag{25}$$

10. EXAMPLE: SHAPE OPTIMIZATION OF THE CENTRAL SECTION OF AN ARCH DAM

Figure 3 shows the parametric description of the geometry of the section studied. The design variables are the x coordinates of the eight points shown in the figure. The objective function is the total wheight of the dam. The material properties are $E = 3,000,000\ T/m^2$, $\nu = 0.2$ and $\rho = 2.3\ T/m^2$.

Three load cases corresponding to two different structural behaviours have been considered:

1) Self weight: the dam has been here analyzed under plane stress conditions assuming no interactions between adjacent blocks as typically occurs during the construction process. Limit values of the compressive and tensile stresses of 2750 T/m^2 and 50 T/m^2 have been taken.
2) Self weight with/without hydrostatic pressure: this two load cases correspond to the operational level of the dam after construction. The

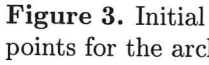


Figure 3. Initial design and definition points for the arch dam problem.

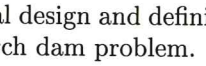


Figure 4. Obtained design and meshes.

structural behaviour has been modeled here using an axisymmetric solid formulation. The maximum restricted values of the compressive and tensile stresses in this case were of 2750 T/m^2 and 20 T/m^2 respectively.

In addition, the minimum top width was limited to $5m$. Also, $119 \leq (x_1 + x_2) \leq 121$ was assumed. This constrains the movement of the base towards the x axes so as to preserve the original distance of 60 m. Simple six noded quadratic triangular elements were used for the analysis and the target percentage of global error was taken as a 5%.

Figure 4 shows some of the designs and meshes obtained during the optimization process. The evolution of the objective function is displayed in Figure 5 whereas Figures 6 shows the evolution of the percentage of global error corresponding to each load case. Note the higher density of elements in the lower part of the dam due to the stress concentration at the base corners. This corresponds well with the mesh optimality criteria chosen (equidistribution of error density) which tends to concentrate more elements in the zones where higher stress gradients occur [5].

Figures 7 to 9 show a repetition of the same example including now the foundation in the analysis and using a target percentage of global error of 10%. Note that the final shape obtained is very similar to that of the

Figure 5. Evolution of the objective function.

Figure 6. Evolution of the error:
• self weight under plane stress.
o self weight under plane strain.
× self weight & water pressure under plane strain conditions.

Figure 7. Design and meshes obtained during the optimization process of the dam with foundation.