INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2004; **46**:1169–1180
Published online 20 October 2004 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.794

Finite element methods based on a new formulation for the non-stationary incompressible Navier–Stokes equations

Ping Lin[1,*,†], Xianqiao Chen[2,‡] and Ming Tze Ong[1,§]

[1] *Department of Mathematics, The National University of Singapore, 2 Science Drive 2, Singapore 117543, Singapore*
[2] *School of Computer Science and Engineering, The Wuhan University of Technology, Yu Jia Tou, Wuhan 430063, China*

SUMMARY

A new formulation of the Navier–Stokes equations is introduced to solve incompressible flow problems. When finite element methods are used under this formulation there is no need to worry whether Babuska–Brezzi condition is satisfied or not. Both velocity and pressure can be obtained separately and the pressure can be simply obtained by a substitution. Moreover, fully explicit time integration can be applied for easy implementation. Implementation issues are discussed and a couple of flow examples are simulated. Parallel implementation based on domain decomposition is incorporated as well. Copyright © 2004 John Wiley & Sons, Ltd.

KEY WORDS: incompressible Navier–Stokes equations; finite element methods; parallel methods; differential-algebraic equations; explicit methods; regularization

1. INTRODUCTION

Consider non-stationary Navier–Stokes equations:

$$\mathbf{u}_t + (\mathbf{u}\cdot \text{grad})\mathbf{u} = \mu\Delta\mathbf{u} - \text{grad}\, p + \mathbf{f} \tag{1}$$

$$\text{div}\,\mathbf{u} = 0 \tag{2}$$

$$\mathbf{u}|_{\partial\Omega} = \mathbf{b},\ \ \mathbf{u}|_{t=0} = \mathbf{a} \tag{3}$$

*Correspondence to: P. Lin, Department of Mathematics, The National University of Singapore, 2 Science Drive 2, Singapore 117543, Singapore.
†E-mail: matlinp@nus.edu.sg
‡E-mail: chenxqgj@public.wh.hb.cn
§E-mail: sci00056@nus.edu.sg

Received 20 June 2003
Revised 16 April 2004

in a bounded domain Ω and the time interval $0 \leqslant t \leqslant T$. Here $\mathbf{u}(\mathbf{x}, t)$ represents the velocity of a viscous incompressible fluid flow, $p(\mathbf{x}, t)$ the pressure, $\mathbf{f}$ the prescribed external force, $\mathbf{a}(\mathbf{x})$ the prescribed initial velocity, and $\mathbf{b}(t)$ the prescribed velocity boundary values.

A huge number of methods have been proposed for the numerical solution of the problem. Direct discretizations include finite difference and finite volume techniques on staggered grids, mixed finite element methods using conformal and non-conformal elements and spectral methods. Initial reformulations and/or regularizations of the equations have also been considered. Examples of such methods include pseudo-compressibility methods, projection and pressure-Poisson reformulations (e.g. References [1–3]). Among them, penalty method is important since its reformulation is very simple (the calculations for $\mathbf{u}$ and p are separated) and artificial boundary values for pressure p is not required. However, an obvious drawback is that it results in a very stiff problem and explicit time discretizations are not possible to be used.

In References [4, 5], a *sequential regularization formulation* or method was proposed and analysed. A variant of the method we are going to use can be seen as an augmented Lagrangian (Uzawa) method applied directly to the non-stationary problem. Unlike the augmented Lagrangian method theoretical justification of convergence based on asymptotic expansion is done for fully non-linear Navier–Stokes equations. We are interested in using the formulation because it keeps the benefits of the penalty but, unlike the penalty method, the regularized problems are more stable or less stiff. Hence, more convenient (non-stiff) methods can be used for time integration, i.e. fully explicit time discretizations satisfying usual time step restrictions. This property is especially attractive when we solve non-linear problems such as Navier–Stokes equations. Also, as indicated in Reference [5] for a simple difference scheme, the time step restriction may be loosened in the case of small viscosity μ for the Navier–Stokes equations. Parallel implementation is also discussed and implemented in Reference [6]. Finite element analysis of the formulation for the model of the displacement in porous media is given in Reference [7].

In this paper we will focus on finite element methods for solving the non-stationary Navier–Stokes equations using this formulation. We will provide theoretical results, discuss implementation issues and compute a few flow problems to demonstrate the method. In a finite element setting of the Navier–Stokes equations the classical Galerkin/variational formulation naturally gives rise to what is termed a mixed method. The success of a formulation of this type was strongly dependent on the particular pair of velocity and pressure interpolations (basis functions) employed. That is, the so-called Babuska–Brezzi condition has to be satisfied. Double mesh implementation of such idea can be found, for example, in the review paper [8]. Although numerous convergent combinations of velocity and pressure elements have been developed, it may be fair to say that most, if not all, involve interpolations (basis functions) which are inconvenient from an implementational standpoint. Three-dimensional elements are particularly hard to pass the Babuska–Brezzi test. In Reference [9] a residual-based stable Petrov–Galerkin formulation for the Stokes problem is presented to circumvent the condition. Rather general C^0 elements can be used in their formulation. In our sequential regularization formulation velocity and pressure are treated separately. Any finite element spaces can be used to solve for the velocity under the formulation. The pressure can then be obtained by an explicit substitution using the velocity obtained earlier. We do not need to worry beforehand whether our elements would pass the Babaska–Brezzi test although it may be automatically satisfied via the formulation.

2. THE SEQUENTIAL REGULARIZATION METHOD AND ITS CONVERGENCE

The importance of the treatment of the incompressibility constraint has long been recognized. A classical approach is the projection method, where one has to solve a Poisson equation for the pressure p with zero Neumann boundary condition which is, however, an unphysical boundary condition. A reinterpretation of the projection method in the context of the so-called pressure stabilization method or, more generally, 'pseudo-compressibility methods' (such as artificial compressibility, penalty method, etc.) has been reviewed in Reference [3]. Our sequential regularization formulation is based on a popular technique—Baumgarte's stabilization from differential-algebraic equation context (cf. References [4, 10]) combined with a modified penalty method. That is, we replace the incompressibility condition by the following:

$$\alpha_1(\operatorname{div}\mathbf{u})_t + \alpha_2 \operatorname{div}\mathbf{u} = 0 \tag{4}$$

where α_1 and α_2 are non-negative constants. Obviously if the initial condition satisfies $\operatorname{div}\mathbf{u}=0$ then (1) coupled with the new equation (4) would have the same solution as that of the original Navier–Stokes equations (1)–(2). We now apply a modified penalty idea to the new equation (4) and obtain

$$-\varepsilon(p - p_0) = \alpha_1(\operatorname{div}\mathbf{u})_t + \alpha_2\operatorname{div}\mathbf{u} \tag{5}$$

where ε is a small penalty constant and p_0 is an initial guess of the pressure, satisfying $\int_\Omega p_0\, \mathrm{d}\mathbf{x}=0$. Coupled (5) with the momentum equation (1) we can solve for both $\mathbf{u}$ and p and then p_0 can be replaced by the newly obtained p to continue the procedure recursively. We thus obtain the sequential regularization formulation: with $p_0(\mathbf{x},t)$ an initial guess (which is usually taken as zero) and $\alpha_1,\alpha_2 \geqslant 0$, for $s=1,2,\ldots,$ solve the problem

$$\begin{aligned}\varepsilon(\mathbf{u}_s)_t - \operatorname{grad}(\alpha_1(\operatorname{div}\mathbf{u}_s)_t + \alpha_2\operatorname{div}\mathbf{u}_s) + \varepsilon(\mathbf{u}_s\cdot\operatorname{grad})\mathbf{u}_s \\ = \varepsilon\mu\Delta\mathbf{u}_s - \varepsilon\operatorname{grad} p_{s-1} + \varepsilon\,\mathbf{f}\end{aligned} \tag{6}$$

$$\mathbf{u}_s|_{\partial\Omega} = \mathbf{b}, \mathbf{u}_s|_{t=0} = \mathbf{a} \tag{7}$$

$$p_s = p_{s-1} - \frac{1}{\varepsilon}(\alpha_1(\operatorname{div}\mathbf{u}_s)_t + \alpha_2\operatorname{div}\mathbf{u}_s) \tag{8}$$

The sequential regularization method is related to augmented Lagrangian method (Uzawa's algorithm) [11, 12] and their difference is explained in Reference [5]. The method is a dynamic iterative method and its convergence can be justified by asymptotic expansion technique. If we use $\mathbf{L}^p(\Omega)$, or simply $\mathbf{L}^p$, to denote the space of functions defined and pth-power integrable in Ω, and

$$\|\mathbf{u}\|_p = \left(\int_\Omega \sum_{i=1}^n u_i^p \,\mathrm{d}\mathbf{x}\right)^{1/p}$$

its norm, where $\mathbf{u}=(u_1,\ldots,u_n)$. We denote the inner product in $\mathbf{L}^2$ by $(\cdot,\cdot)$. $\mathbf{C}^\infty$ is the space of functions continuously differentiable any number of times in Ω, and $\mathbf{C}_0^\infty$ consists of those members of $\mathbf{C}^\infty$ with compact support in Ω. $\mathbf{H}^m$ is its completion in the norm

$$\|\mathbf{u}\|_{\mathbf{H}^m} = \left(\sum_{0 \leqslant |\alpha| \leqslant m} \|D^\alpha \mathbf{u}\|^2 \right)^{1/2}$$

The following convergence estimates are proved in Reference [5].

$$\|\mathbf{u} - \mathbf{u}_s\|_{\mathbf{H}^1} \leqslant M\varepsilon^s$$

$$\left(\int_0^{\mathrm{T}} \|p - p_s\|_2 \, \mathrm{d}t \right)^{1/2} \leqslant M\varepsilon^s$$

where $s = 1, 2, \ldots$. From the result we can see we only need to assume the penalty parameter $\varepsilon < 1$ but not necessary to choose it very small since by iterations the error $O(\varepsilon^s)$ can reach any accuracy that we want. Since we can choose ε not to be very small the formulation is less stiff or more stable and thus the formulation makes it possible to have a fully explicit time discretization (noting that if we apply a explicit discretization to the original equations then we still need to solve a Poisson equation for pressure p thus we do not really have an explicit discretization). From the estimate the convergence rate of the method is explicitly given as ε. We can easily control the convergence rate by choosing appropriate parameter ε.

3. DISCRETIZATION AND IMPLEMENTATION ISSUES

In this paper we will use finite element methods for spatial discretization and explicit schemes for temporal discretization. The variational form of the regularized problem for any fixed s can be easily formulated:

Find $\mathbf{v} \in \mathbf{H}^1$ and $\mathbf{v}|_{\mathbf{x} \in \Omega} = \mathbf{a}$ such that

$$\begin{aligned} &\varepsilon((\mathbf{v})_t, \mathbf{w}) + \alpha_1((\operatorname{div} \mathbf{v})_t, \operatorname{div} \mathbf{w}) + \alpha_2(\operatorname{div} \mathbf{v}, \operatorname{div} \mathbf{w}) + \varepsilon((\mathbf{v} \cdot \operatorname{grad})\mathbf{v}, \mathbf{w}) \\ &\quad = -\varepsilon\mu(\operatorname{grad} \mathbf{v}, \operatorname{grad} \mathbf{w}) - \varepsilon(p_{s-1}, \operatorname{div} \mathbf{w}) + \varepsilon(\mathbf{f}, \mathbf{w}) \quad \forall \mathbf{w} \in \mathbf{H}_0^1 \end{aligned} \tag{9}$$

where $\mathbf{v} = \mathbf{u}_s$ for simplicity of notation. The pressure p_s can be explicitly updated and computed by formula (8). So any standard interpolation patterns (finite elements) can be used to compute $\mathbf{v}$ or $\mathbf{u}_s$. There is no need to consider Babuska–Brezzi condition to match the interpolation patterns used for the velocity and the pressure. The interpolation pattern for p_s will be automatically obtained from the interpolation pattern for $\mathbf{v}$ and from formula (8). We do not need to worry if the automatically obtained interpolation pattern would satisfy the Babuska–Brezzi condition although the condition may possibly be satisfied under the formulation.

From now on we will always take $\alpha_1 = 0$ (and $\alpha_2 = 1$ without loss of generality). It will make the formulation simple. In this case the sequential regularization formulation can be seen as an augmented Lagrangian method (Uzawa's algorithm) applied directly to the non-stationary fully non-linear problem without linearization. Only in this case a fully explicit time discretization is possible.

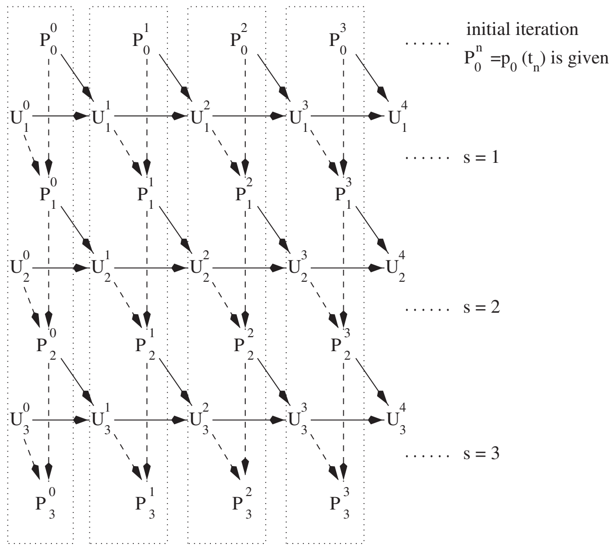


Figure 1. Implementation of the SRM using the forward Euler scheme.

Below we will describe the method using two-dimensional terms. There is certainly no problem to use it to higher dimensional cases. Let $h > 0$ be the mesh size of a finite decomposition of the domain Ω. We introduce a standard finite element space $\mathbf{W}_h \subset \mathbf{H}_0^1$ for Galerkin methods associated with a quasi-regular subdivision of Ω into triangles or rectangles of diameter less than h. Concretely, let $\Pi_h = \{\tau_i^h\}_{i=1}^N$ be a finite decomposition of Ω into a family of edge-to-edge triangles or quadrilaterals of $\Omega_h \subseteq \Omega$ with the parameter $h \approx N^{-1/2}$ uniformly comparable to $\max_i(\text{diam}(\tau_i^h))$ and with $\text{diam}(\tau_i^h) \leqslant K_1 h \leqslant K_2 \rho(\tau_i^h)$, where $\rho(\tau)$ denotes the diameter of the largest inscribed disc of τ, and

$$\mathbf{W}_h = \{\mathbf{w} \in \mathbf{C}^0, \mathbf{w} = 0 \text{ on } \partial\Omega, \ \mathbf{w}|_{\tau_i^h} \text{ polynomials of degree} \leqslant k\}$$

Given a partition of $[0, T]$, $0 = t_0 < t_1 < \cdots < t_N = T$, we denote $\Delta t_n = t_{n+1} - t_n$, and for simplicity let $\Delta t_n = \Delta t$ be a constant. Let $\{P_s^n, \mathbf{V}^n\}$ be the approximation of $\{p_s, \mathbf{v}\}$ at time level t_n.

We have indicated that simple explicit time discretizations can be used to make the computation and its programming easier. Let us just use forward Euler scheme (higher order explicit schemes can be used as well to obtain better accuracy and less time step restriction associated with the explicit scheme) and define the full approximation scheme (combined the Euler scheme with the standard Galerkin method) at time $t_n (n = 0, 1, 2, \ldots)$ by the following.

Given any initial guess $P_0^n, n = 0, 1, \ldots, N$. For $s = 1, 2, \ldots$, iteratively obtain

$$\{\mathbf{V}^n, P_s^n\} \equiv \{\mathbf{U}_s^n, P_s^n\}, \quad n = 0, 1, \ldots, N$$

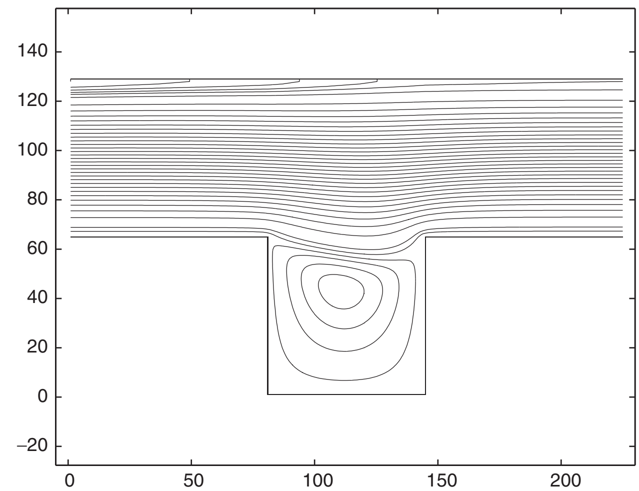


Figure 2. Streamline of the channel flow with a cave at $t = 60$.

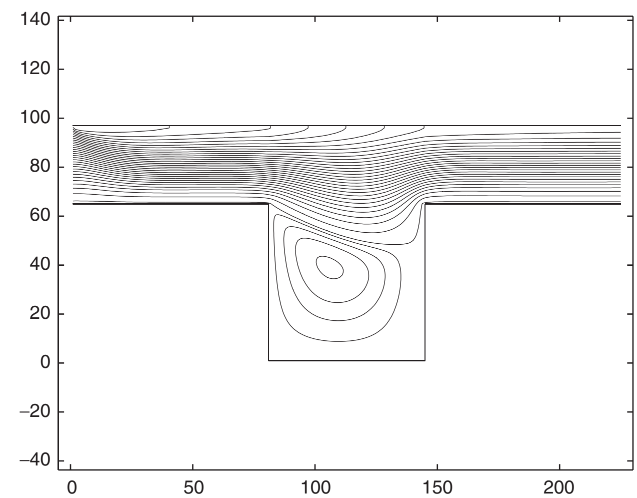


Figure 3. Streamline of the channel flow with a reduced upper channel height of 0.5 at $t = 60$.

Concretely, at the sth SRM iteration ($P^n_{s-1}, n = 0, 1, \ldots, N$ are known at this iteration), starting from the initial values $\mathbf{V}^0$, find $\mathbf{V}^{n+1} \in \mathbf{W}_h$ such that

$$\left(\frac{\mathbf{V}^{n+1} - \mathbf{V}^n}{\Delta t}, \mathbf{w}\right) + \frac{1}{\varepsilon}(\operatorname{div} \mathbf{V}^n, \operatorname{div} \mathbf{w}) + (\mathbf{V}^n \cdot \operatorname{grad})\mathbf{V}^n, \mathbf{w})$$
$$= -\mu(\operatorname{grad} \mathbf{V}^n, \operatorname{grad} \mathbf{w}) - (P^n_{s-1}, \operatorname{div} \mathbf{w}) + (\mathbf{f}^n, \mathbf{w}) \quad \forall \mathbf{w} \in \mathbf{W}_h \tag{10}$$

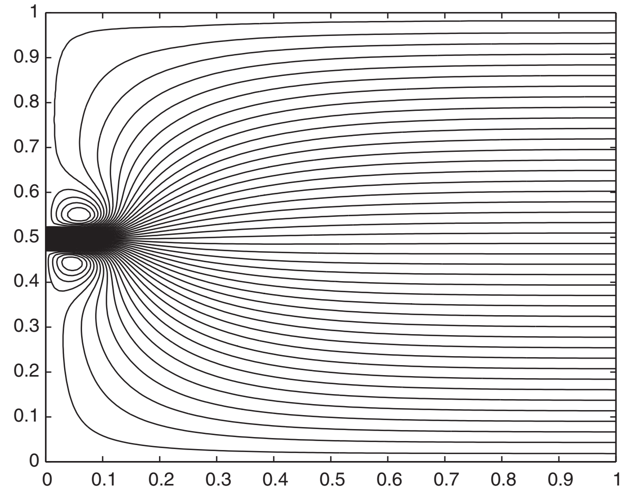


Figure 4. Streamline of the channel jet at $t = 0.1$.

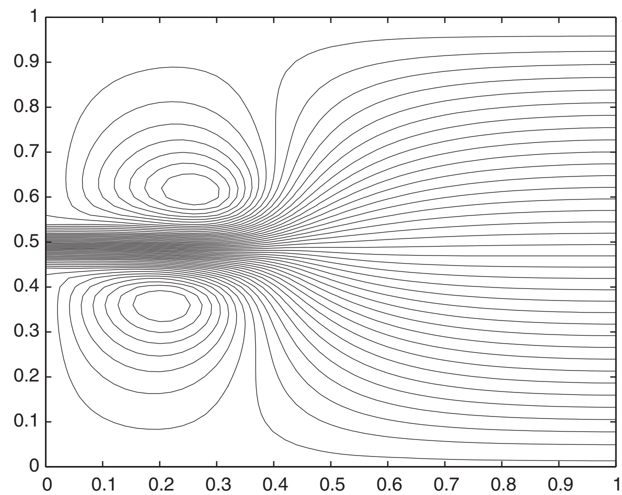


Figure 5. Streamline of the channel jet at $t = 1.0$.

and then obtain P_s^n by the direct substitution:

$$P_s^n = P_{s-1}^n - \frac{1}{\varepsilon} \operatorname{div} \mathbf{V}^n \tag{11}$$

for $n = 0, 1, \ldots, N-1$.

We call the above procedure a complete explicit procedure because in other formulations of incompressible Navier–Stokes equations a linear system arising from the pressure Poisson equation has to be solved even if an explicit temporal scheme is used. Of course, in the case of high Reynolds number, streamline diffusion finite element term or other type of stabilization

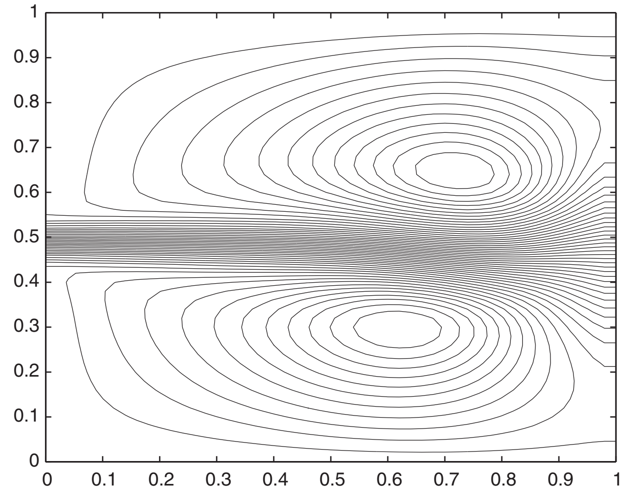


Figure 6. Streamline of the channel jet at $t = 2.9$.

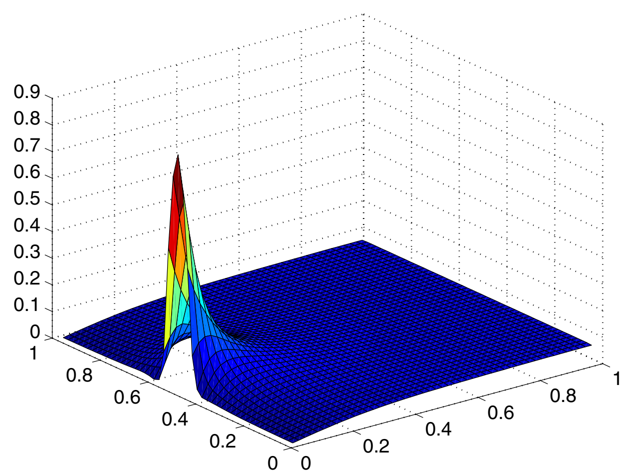


Figure 7. Horizontal velocity of the channel jet at $t = 0.1$.

term should be added to deal with the dominant convection term in order to achieve stability (cf. Reference [13]). Our computational experience shows that usually four or five iterations would be enough for $\varepsilon = 0.5$.

The order of the sequential regularization iteration is: starting from $s = 0$ (i.e. given initial guess P_0^n for all $t_n \in [0, T]$), find $\mathbf{U}_1^n, P_1^n$ for all $t_n \in [0, T]$, then find $\mathbf{U}_2^n, P_2^n$ for all n, etc. That is, to compute $\mathbf{U}_s^n, P_s^n$ at the current iteration s we need to store $\mathbf{U}_{s-1}^n, P_{s-1}^n$ at the previous iteration $s-1$ for all time n. Since usually time step size is small and the number of time steps would be huge. It is not realistic to store all values of $\mathbf{U}_{s-1}^n, P_{s-1}^n$, $n = 0, 1, 2, \ldots$ in order to compute the current iteration $\mathbf{U}_s^n, P_s^n$. In the sequential regularization method because the

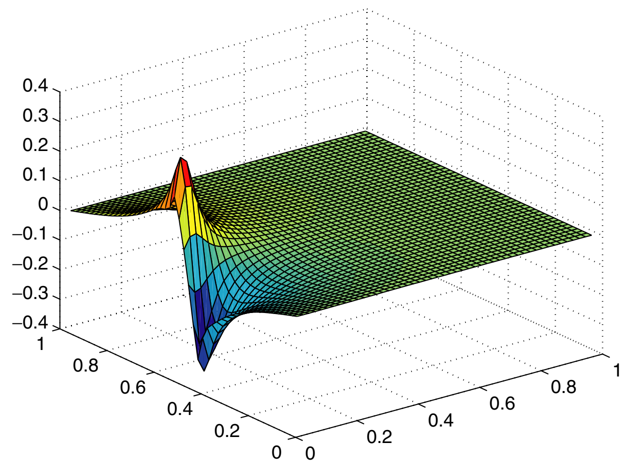


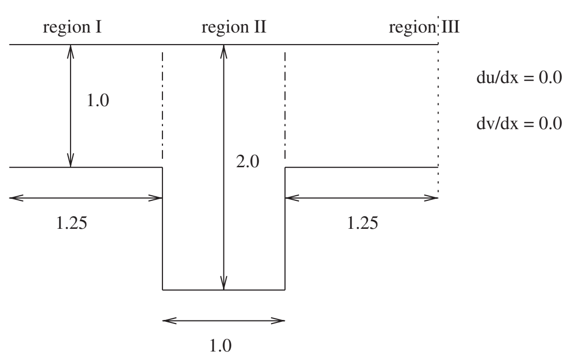
Figure 8. Vertical velocity of the channel jet at $t = 0.1$.

number of iterations is fixed (usually four or five) and is determined in advance (according to the accuracy of the discretization used) we can design an order of the iterations to avoid this large storage problem. In Figure 1 the implementation order of the SRM is depicted for forward Euler time discretization. Similar diagram can be drawn for other explicit schemes. From the diagram we see that our implementation can go (dotted) box by (dotted) box in the temporal direction. Since we usually do four or five iterations we only need to store four or five values of $\mathbf{U}_s^n$ and P_s^n in the algorithm.

4. NUMERICAL SIMULATION OF SOME INCOMPRESSIBLE VISCOUS FLOWS

The main goal of this section is to present the results of numerical experiments concerning the simulation of two-dimensional incompressible viscous flows.

The first example is a channel flow with a cave. The flow domain Ω is a rectangular channel plus a square cave as shown in the following diagram. We assume that the



fluid viscosity v is equal to $\frac{1}{100}$ and that the fluid is at rest at time $t = 0$, that is, $\mathbf{u}(\mathbf{x}, 0) = \mathbf{0}$, for all $\mathbf{x} \in \Omega$. The inflow profile is parabolic $\mathbf{u} = (y(1-y), 0)^{\mathrm{T}}$. The outflow boundary condition is

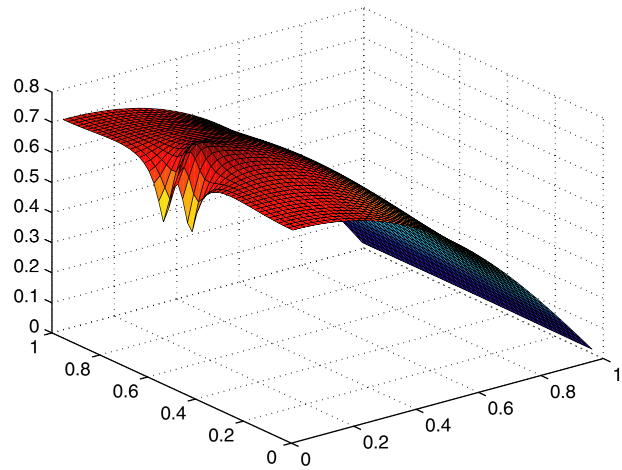


Figure 9. Pressure of the channel jet at $t = 0.1$.

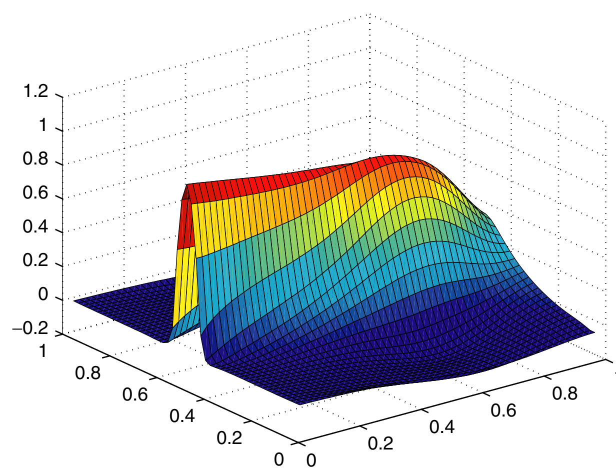


Figure 10. Horizontal velocity of the channel jet at $t = 2.9$.

$\partial \mathbf{u}/\partial x = 0$. On the other boundaries $\mathbf{u} = 0$. We have divided the domain to three subdomains I–III where two processors are assigned to subdomains I and III and four processors are assigned to subdomain II for parallel implementation. We take piecewise linear interpolation on a uniform triangular grid with $h_x = h_y = \frac{1}{64}$ (which is equivalent to certain finite difference discretization) and temporal step size $\Delta t = 2 \times 10^{-5}$. The penalty parameter $\varepsilon = 0.5$ and do four sequential regularization iterations. In Figure 2 we depict the streamline of the solution after three million time steps (reaching steady-state). We also calculate the case where the upper channel boundary moves down to be close to the cave. The computational result is depicted in Figure 3. We see that more interaction takes place between the flow in the channel and the flow inside the cave.

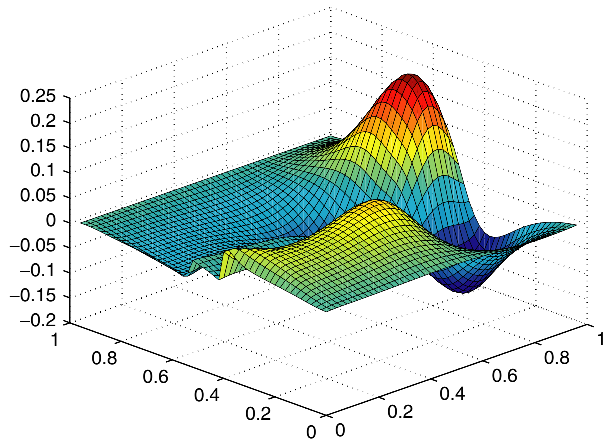


Figure 11. Vertical velocity of the channel jet at $t = 2.9$.

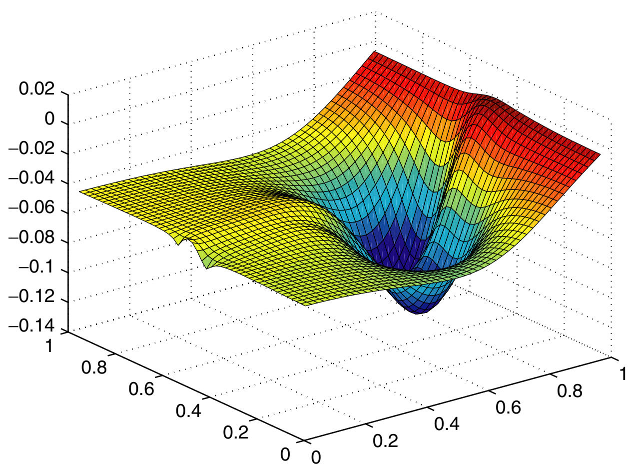


Figure 12. Pressure of the channel jet at $t = 2.9$.

The second example is a channel jet flow. The flow domain Ω is a rectangle. We assume that the fluid viscosity $v = \frac{1}{2000}$, that the fluid is at rest at time $t = 0$, and that the jet aperture is located on the left vertical line, is centred at the middle and is $\frac{1}{16}$ wide. We suppose also that the jet profile is parabolic, with a maximal velocity equal to 1, the jet being horizontal. The velocity at the outflow boundary is zero in the vertical direction and zero normal derivative at the horizontal direction. The velocity at other boundaries is zero. We use again the piecewise linear interpolation on a uniform triangular grid combined with the forward Euler explicit temporal discretization. $h_x = h_y = \frac{1}{50}$ and $\Delta t = 5 \times 10^{-5}$. We also tried finer spatial grids and see no significant difference from the results.

In Figures 4–6 we depict streamlines of the channel jet flow at different times. When $t = 2.9$ the jet is close to the right artificial boundary and some non-symmetry starts to develop. Longer domain is needed to simulate the channel jet flow further. In Figures 7–12 we show velocity and pressure surface at $t = 0.1$ and 2.9.

REFERENCES

1. Chorin AJ. Numerical solution of the Navier–Stokes equations. *Mathematics of Computation* 1968; **22**: 745–762.
2. Temam R. *Navier–Stokes Equations*. North-Holland: Amsterdam, 1977.
3. Rannacher R. On the numerical solution of the incompressible Navier–Stokes equations. *Zeitschrift fur Angewandte Mathematik und Mechnik* 1993; **73**:203–216.
4. Ascher U, Lin P. Sequential regularization methods for higher index DAEs with constraint singularities: linear index-2 case. *SIAM Journal on Numerical Analysis* 1996; **33**:1921–1940.
5. Lin P. A sequential regularization method for time-dependent incompressible Navier–Stokes equations. *SIAM Journal on Numerical Analysis* 1997; **34**:1051–1071.
6. Lin P, Guo QP, Chen XQ. A fully explicit method for incompressible flow computation. *Computer Methods in Applied Mechanics and Engineering* 2003; **192**:2555–2564.
7. Lin P, Yang DQ. An iterative perturbation method for the pressure equation in the simulation of miscible displacement. *SIAM Journal on Scientific Computing* 1998; **19**:893–911.
8. Glowinski R. *Finite element methods for the numerical simulation of incompressible viscous flow: introduction to the control of the Navier–Stokes equations.* Lectures in Applied Mathematics, vol. 28. American Mathematics Society: Providence, RI, 1991; 219–301.
9. Hughes TJR, Franca LP, Balestra M. A new finite element formulation for computational fluid dynamics: V. Circumventing the Babuska–Brezzi condition: a stable Petrov–Galerkin formulation of the Stokes problem accommodating equal-order interpolations. *Computer Methods in Applied Mechanics and Engineering* 1986; **59**:85–99.
10. Ascher U, Lin P. Sequential regularization methods for nonlinear higher index DAEs. *SIAM Journal on Scientific Computing* 1997; **18**:160–181.
11. Fortin M, Glowinski R. *Augmented Lagrangian methods: applications to the numerical solution of boundary-value problems.* Studies in Mathematics and its Applications, vol. 15. North-Holland: Amsterdam, 1983.
12. Arrow K, Hurwicz L, Uzawa H. *Studies in Linear and Nonlinear Programming.* Stanford University Press, Stanford, CA, 1958.
13. Hughes TJR. *The Finite Element Method*: *Linear Static and Dynamic Finite Element Analysis.* Prentice-Hall: Englewood Cliffs, NJ, 1987.
14. Baumgarte J. Stabilization of constraints and integrals of motion in dynamical systems. *Computer Methods in Applied Mechanics and Engineering* 1972; **1**:1–16.

