INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 18, 71–105 (1994)

TOOLS FOR SIMULATING NON-STATIONARY INCOMPRESSIBLE FLOW VIA DISCRETELY DIVERGENCE-FREE FINITE ELEMENT MODELS

STEFAN TUREK

Institute for Applied Mathematics, University of Heidelberg, INF 294, Heidelberg, Germany

SUMMARY

We develop simulation tools for the non-stationary incompressible 2D Navier–Stokes equations. The most important components of the finite element code are: the fractional step ϑ-scheme, which is of second-order accuracy and strongly A-stable, for the time discretization; a fixed point defect correction method with adaptive step length control for the non-linear problems (stationary Navier–Stokes equations); a modified upwind discretization of higher-order accuracy for the convective terms. Finally, the resulting nonsymmetric linear subproblems are treated by a special multigrid algorithm which is adapted to the quadrilateral non-conforming discretely divergence-free finite elements. For the graphical postprocess we use a fully non-stationary and interactive particle-tracing method. With extensive test calculations we show that our method is a candidate for a 'black box' solver.

KEY WORDS Non-stationary Navier–Stokes equations Upwind Divergence-free finite elements Multigrid Visualization

1. INTRODUCTION

We consider the 'usual' Navier–Stokes equations

$$\mathbf{u}_t - \nu\Delta\mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p = \mathbf{f}, \quad \nabla\cdot\mathbf{u} = 0 \quad \text{in } \Omega \times (0, T)$$

for a given force $\mathbf{f}$ and viscosity ν, with prescribed boundary values on $\partial\Omega$ and an initial condition at $t = 0$. The variables $\mathbf{u}$ and p describe the velocity and pressure respectively of a viscous incompressible flow in a bounded region $\Omega \subset \mathbf{R}^2$.

These fundamental equations are of interest to both more theoretical scientists such as mathematicians or physicists and more applied scientists such as engineers or industrial users. What are the theoretical aspects needed to develop and implement an algorithm to solve these equations? Our mathematical research led us to examine the following problems: efficient treatment of the non-linearity, error estimates for spatial and time discretizations, construction of fast linear solvers, modelling of boundary conditions, and many others. It is very important not only to solve these problems in a theoretical way but also to see the results in 'real life' via computer simulations. It is not sufficient to prove very good convergence rates for a method if the corresponding cost is too high. For instance, direct Gaussian elimination has the best convergence rates (zero) but no one will use it for very large systems because $O(n^3)$ arithmetic operations are needed (with n unknowns). Also, some schemes of (theoretically) first-order

0271–2091/94/010071–35$22.50

Received 1 June 1992
Revised 3 August 1993

accuracy may yield better results than certain second-order methods, since the asymptotic range implicit in the proof is (nearly) never reached. Therefore numerical experiences have a large influence on our theoretical work and should not be neglected.

On the other hand, engineers are also interested in an efficient solution method for these equations. The simulation of various complex fluid structures, the development of new models for 'real' problems and, not least, the saving of money by doing a computer simulation rather than an expensive laboratory test are all requirements which appear every day. Now our aim is to develop and realistically implement a solution method which satisfies both parties; this means we that would like to construct a so-called 'black box' solver. This term is often used, but in many cases its meaning is not clear, so we must define it. What we cannot offer is a 'black universal method'; we only present a method which attempts to give the user a 'fast', 'robust' and 'accurate' solution. Then the problem is to determine whether the computed 'solution' is physically relevant, since our method by itself does not necessarily produce the 'right solution' fulfilling a prescribed accuracy corresponding to the real flow.

In this sense we will develop an FRA method (fast–robust–accurate), with the emphasis on the term method as opposed to algorithm or solver. Thus first we have to explain the key words 'fast', 'robust' and 'accurate'.

Our implemented algorithm should produce the results in a 'short' amount of time operating on a workstation. We prefer this class of machine, because at today's prices (nearly) every institute can afford such computers. Additionally, these machines are very fast, complete as well as robust and the future trend indicates that they will only improve. By 'fast' we mean that we want to compute a fully developed non-stationary flow in hours with an accuracy which is essentially determined by the available RAM memory. Thus, for instance, in our actual computations we computed the fully developed vortex street behind an ellipse with 40 000 'semi-adaptively' distributed grid points in less than 1 day on a SUN IPC.

Another aspect of the term 'fast' involves being able to change ideas and easily implement new ideas by adding subroutines to the existing code. Hence we need a very modular programming structure and very clever data management. The basis of all our programmes is the finite element package FEAT2D which was developed by our numerical group in Heidelberg.[1] This basis enables several people to work and co-operate in the same field.

Finally, another important aspect of 'fast' is the fast graphical presentation of the computed data. Most workstations have very fine graphics features and, for example, enable one to compute and visualize at the same time. Hence we also have to provide the corresponding graphical algorithms and subroutines.

When we say our method is stable, we think of several aspects. First, the implemented algorithm should be independent of given 'data' such as the shape of the domain (not convex, not polygonal, several boundary components), the mesh (not equidistant, refined in some parts) and the size of the force, viscosity and boundary values. Thus by 'robust' we mean that our finite element discretization works for (nearly) all meshes, that our linear solvers (almost) always produce the same convergence rates and that the non-linear convergence rates do not become too bad for higher Reynolds numbers. What we do not claim is that our algorithm works completely independently of all these 'parameters', since, for instance, it is clear that a stationary Navier–Stokes solver may diverge for higher Reynolds numbers if there is no stable stationary solution and it is also clear that the time step size is dependent on the physical flow structure. The aim is to develop a method which requires defining only the domain, the right-hand side, the boundary values and the viscosity; then the algorithm attempts to produce results in an 'optimal' way. We think the proposed method is a step in this direction.

Another aspect is the robustness of the implementation. The programme should be organized in a very modular structure in order to simplify and control possible errors when making changes to the programme. The worst case is to have a code with 30 000 lines and to change a small part, resulting in errors in other routines. This often costs much more time than many simulation processes with the (correctly) modified algorithm.

We also emphasize that 'robustness' includes the aspect of whether the algorithm is analysable and in a certain sense 'deterministic' for the user. Therefore, for instance, variational formulations and discretizations by finite elements are used, because these allow, at least for some simpler model problems, a strong analysis and can be helpful to explain some 'funny' results of the implemented code. Thus only well-known 'standard methods' are used, such as stable finite element pairs for the discretization, fast multigrid algorithms, fixed point defect correction methods for the non-linearity and well-known time discretization schemes for the non-stationary processes. Finally, our graphical support should work (in a robust way) independently of the above-mentioned 'data', something which is not fulfilled by some existing graphics packages.

The question of accuracy is the most critical and difficult point of our method. As mentioned above, we cannot present a 'black universal method' which tells the user the quality of the computed solutions compared with the exact ones. What we can present is a 'solution' with the 'best accuracy' reachable for a given machine and time. Here one has to consider that the more accurate methods are sometimes not the fastest ones. Therefore very simple finite elements (of second order) and only time and space discretization schemes of first or second order are used, since then corresponding solution methods which are very fast can be constructed (divergence-free finite elements, multigrid methods). For the grid construction process the best method would be a fully adaptive method. One reason we do not use this is because at this moment we do not have the right tool, while on the other hand our 'semi-adaptive' method (see next paragraph) leads to a very regular data structure, which gives perhaps not the absolutely most accurate results but leads to a very efficient multigrid solution method.

Now we want to introduce the three main parts of our 'FRA black box' method:

(i) input of all needed 'data'
(ii) fully modular finite element solver
(iii) graphical postprocessing.

The perfect method would be to define only the computational domain (perhaps in a graphical way) and the input data (such as force term, viscosity and boundary values) and then have the algorithm produce the most accurate solution depending on the available memory and proposed time limit. Unfortunately, at this time we are unable to accomplish this goal. At present our realized method needs, besides the description of the domain, also a 'coarse' grid which can be already slightly adapted to the expected solution (e.g. after small test calculations or after comparing with existing theoretical and practical results). Then this 'coarse' grid is systematically refined (corresponding to the available memory) and the computation may begin. The resulting grids are called 'semi-adaptive' in our framework. For non-stationary problems one still has to define the time step size, because (presently) we did not implement an adaptive step size control. However, since our time discretization scheme in unconditionally stable, this parameter can be chosen depending solely on physical reasons and not on (purely) mathematical stability conditions. These non-self-adaptive methods do not guarantee the most accurate 'solution' but do give a nearly optimal 'solution' in a short amount of time.

The most important components of our finite element code are the following (we leave a more detailed description to the subsequent sections):

(i) the finite element package FEAT2D (in Fortran 77) with a pseudodynamic memory concept and basic finite element tools for grid generation, stiffness matrices, right-hand sides and boundary conditions
(ii) a corresponding package with basic linear algebra tools and linear solvers (e.g. Jacobi, Gauss–Seidel or PCG methods (preconditioned conjugate gradient), several multigrid components) for the linearized Navier–Stokes equations
(iii) upwind discretizations and a fixed point defect correction method for the non-linearity for the stationary Navier–Stokes solver
(iv) time discretization schemes for the fully non-stationary Navier–Stokes equations.

We use standard graphics packages such as MOVIE.BYU (for 2D applications such as streamlines, vector plots and 3D pressure plots) and AVS (for 3D applications). Additionally, we developed some modules for non-stationary flows, for example a fully interactive particle tracing with the corresponding data structure. These 'movies' can also be transferred directly from screen to video tape. We think this non-stationary postprocessing module is the most important one, since only with its help is one really able to examine non-stationary flows, because these graphical results differ greatly from the streamline or vector plot snapshots.

After the realization of all these aims a typical 'application' with the 'FRA black box' method follows the structure depicted in Figure 1, with all these tasks running on only one workstation.

This work is divided into three sections. Section 2 concerns the non-conforming finite element spaces and states some stability and approximation results concerning the Stokes equations. We describe the process of constructing the corresponding discretely divergence-free subspaces and some of their properties. Then we explain our multigrid algorithm with the divergence-free grid transfer routines and give some results concerning efficiency and robustness. Section 3 describes the upwind discretizations used for the convective terms and the fixed point defect correction method for the non-linearity. With these modules we have a solver for stationary Navier–Stokes equations. Then, in conjunction with an appropriate time discretization scheme (implicit Euler, Crank–Nicolson, fractional step method), we get our final solution method. In Section 4 we show some of our test calculations and examine some questions such as a comparison of several discretization errors in space and time, the influence of the 'semi-adaptive' grids and of the

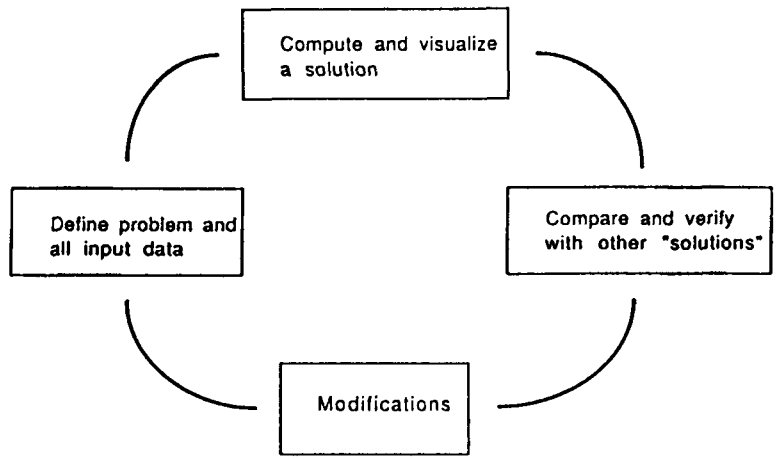

Figure 1. Typical application diagram

boundary conditions. With all our extensive results we hope to demonstrate that the developed method fulfils the conditions of an 'FRA black box' solver. The final section contains conclusions concerning the present results and also gives an outlook to possible future research.

$L^2(\Omega)$ and $H^m(\Omega)$ are the usual Lebesgue and Sobolev spaces for the domain $\Omega \subset \mathbf{R}^2$ with the usual norms $\|\cdot\|_0$ and $\|\cdot\|_m$. The inner product of $L^2(\Omega)$ is $(\cdot, \cdot)$. The space $H_0^1(\Omega)$ is the completion in $H^1(\Omega)$ of the space of test functions $C_0^\infty(\Omega)$ and $H^{-1}(\Omega)$ is its dual space. By $L_0^2(\Omega)$ we denote the subspace of all $L^2(\Omega)$-functions over Ω having mean value zero. Vector-valued functions and spaces are denoted by boldface type, such that $\mathbf{L}^2(\Omega) := L^2(\Omega) \times L^2(\Omega)$, $\mathbf{H}^m(\Omega) := H^m(\Omega) \times H^m(\Omega)$ and $\mathbf{H}_0^1(\Omega) := H_0^1(\Omega) \times H_0^1(\Omega)$. The corresponding norms and inner products are denoted like the analogous scalar ones and no distinction is made. For discrete mesh-dependent constructs such as triangulations, discrete spaces, norms, inner products, etc. we utilize a subscript h. Normal vectors are denoted by $\mathbf{n}$ and tangential vectors by $\mathbf{t}$. All other special notations are introduced and described as needed.

2. THE FINITE ELEMENT SPACES AND THEIR PROPERTIES

We consider the steady Stokes problem

$$-\Delta \mathbf{u} + \nabla p = \mathbf{f}, \quad \nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega, \qquad \mathbf{u} = \mathbf{g} \quad \text{on } \partial\Omega, \tag{St}$$

where the pair $\{\mathbf{u}, p\}$ represents the velocity and pressure respectively in a bounded region $\Omega \subset \mathbf{R}^2$ with prescribed boundary values on $\partial\Omega$ and a given force $\mathbf{f}$. For simplicity we assume that Ω is a convex polygon and that the boundary values $\mathbf{g}$ are homogeneous.

Defining the bilinear forms $a(\mathbf{u}, \mathbf{v}) := (\nabla \mathbf{u}, \nabla \mathbf{v})$ and $b(p, \mathbf{v}) := -(p, \nabla \cdot \mathbf{v})$, the usual weak formulation of the problem (St) reads

$$\begin{aligned} &\text{find a pair } \{\mathbf{u}, p\} \in \mathbf{H}_0^1(\Omega) \times L_0^2(\Omega) \text{ such that} \\ &\qquad a(\mathbf{u}, v) + b(p, \mathbf{v}) + b(q, \mathbf{u}) = (\mathbf{f}, \mathbf{v}) \quad \forall \{\mathbf{v}, q\} \in \mathbf{H}_0^1(\Omega) \times L_0^2(\Omega). \end{aligned} \tag{V}$$

An equivalent 'shorter' formulation with $\mathbf{V}(\Omega) = \{\mathbf{v} \in \mathbf{H}_0^1(\Omega) : \nabla \cdot \mathbf{v} = 0\}$ is

$$\begin{aligned} &\text{find } \mathbf{u} \in \mathbf{V}(\Omega) \text{ such that} \\ &\qquad a(\mathbf{u}, \mathbf{v}) = (\mathbf{f}, \mathbf{v}) \quad \forall \mathbf{v} \in \mathbf{V}(\Omega). \end{aligned} \tag{V^d}$$

Problem (V) has a unique solution for any force $\mathbf{f} \in \mathbf{H}^{-1}(\Omega)$ (see e.g., Reference 2), which is a consequence of the well-known stability estimate

$$\sup_{\mathbf{v} \in \mathbf{H}_0^1(\Omega)} \frac{(q, \nabla \cdot \mathbf{v})}{\|\nabla \mathbf{v}\|_0} \geqslant \beta \|q\|_0 > 0 \quad \forall q \in L_0^2(\Omega), \quad q \neq 0, \tag{BB}$$

and if $\mathbf{f} \in \mathbf{L}^2(\Omega)$, then the solution is in $\mathbf{H}^2(\Omega) \times H^1(\Omega)$ and satisfies the *a priori* estimate

$$\|\mathbf{u}\|_2 + \|p\|_1 \leqslant c \|\mathbf{f}\|_0. \tag{1}$$

For the discretization let $\mathbf{T}_h$ be regular decompositions of the domain Ω into (convex) quadrilaterals denoted by T, where the mesh parameter $h > 0$ describes the maximum diameter of the elements of $\mathbf{T}_h$. $\partial \mathbf{T}_h$ denotes the set of all boundary edges of the elements $T \in \mathbf{T}_h$. Additionally, the family $\{\mathbf{T}_h\}_h$ is assumed to satisfy the usual *uniform shape condition.*[3,4] The common edge between two elements T_i, $T_j \in \mathbf{T}_h$ is denoted by Γ_{ij} with corresponding midpoint m_{ij}. Analogously, we define the boundary edges $\Gamma_{i0} \subset (\partial \mathbf{T}_h \cap \partial\Omega)$ with the midpoints m_{i0}. To obtain the fine mesh $\mathbf{T}_h$ from a coarse mesh $\mathbf{T}_{2h}$, we simply connect opposing midpoints (true

domain boundaries are respected, which means that all boundary nodes are on the true domain boundary). In the new grid $\mathbf{T}_h$ the old midpoints become vertices. As mentioned before, for programming reasons in our multigrid approach, we do not use adaptive techniques in this work.

For the approximation of problem (V) by the finite element method we have to introduce discrete spaces $\mathbf{H}_h \approx \mathbf{H}_0^1(\Omega)$ and $L_h \approx L_0^2(\Omega)$. For this we use the reference element $\hat{T} = [-1, 1]^2$ and define for each $T \in \mathbf{T}_h$ the corresponding 1–1 transformation ψ_T: $\hat{T} \to T$. Then we set ('rotated bilinear elements',[5])

$$\hat{Q}_1(T) := \{q \circ \psi_T^{-1} \mid q \in \operatorname{span}\langle x^2 - y^2, x, y, 1\rangle\}. \tag{2}$$

The degrees of freedom are determined by the nodal functionals $\{F_\Gamma^{(a/b)}(\cdot), \Gamma \subset \partial\mathbf{T}_h\}$ with

$$F_\Gamma^{(a)}(v) := |\Gamma|^{-1} \oint_\Gamma v \, d\gamma \quad \text{or} \quad F_\Gamma^{(b)}(v) := v(m_\Gamma). \tag{3}$$

Either choice is unisolvent with $\hat{Q}_1(T)$, but each leads to different finite element spaces, since the applied midpoint rule is only exact for linear functions. The corresponding (*parametric*) finite element spaces $\mathbf{H}_h = \mathbf{H}_h^{(a/b)}$ and L_h are

$$L_h := \{q_h \in L_0^2(\Omega) \mid q_{h|T} = \text{const.}, \forall T \in \mathbf{T}_h\}, \qquad \mathbf{H}_h^{(a/b)} := S_h^{(a/b)} \times S_h^{(a/b)}, \tag{4}$$

with

$$S_h^{(a/b)} := \begin{cases} \{v_h \in L^2(\Omega) \mid v_{h|T} \in \hat{Q}_1(T), \forall T \in \mathbf{T}_h, v_h \text{ continuous w.r.t. all the} \\ \text{nodal functionals } F_{\Gamma_{ij}}^{(a/b)}(\cdot), \forall \Gamma_{ij}, \text{ and } F_{\Gamma_{i0}}^{(a/b)}(v_h) = 0, \forall \Gamma_{i0}\}. \end{cases} \tag{5}$$

Since the spaces $\mathbf{H}_h^{(a/b)}$ are non-conforming, i.e. $\mathbf{H}_h^{(a/b)} \not\subset \mathbf{H}_0^1(\Omega)$, we have to work with elementwise defined bilinear forms and corresponding energy norms:

$$a_h(\mathbf{u}_h, \mathbf{v}_h) := \sum_{T \in \mathbf{T}_h} \int_T \nabla \mathbf{u}_h \cdot \nabla \mathbf{v}_h \, dx, \qquad \|\mathbf{v}_h\|_h := (a_h(\mathbf{v}_h, \mathbf{v}_h))^{1/2}, \tag{6}$$

$$b_h(q_h, \mathbf{v}_h) := -\sum_{T \in \mathbf{T}_h} q_{h|T} \int_T \nabla \cdot \mathbf{v}_h \, dx. \tag{7}$$

Let j_h: $L_0^2(\Omega) \to L_h$ be the operator of piecewise constant interpolation (modified to preserve the zero-mean-value property) which satisfies[4]

$$\|q - j_h q\|_0 \leqslant ch\|q\|_1 \quad \forall q \in L_0^2(\Omega) \cap H^1(\Omega). \tag{8}$$

Further, let $i_h^{(a/b)}$: $\mathbf{H}_0^1(\Omega) \to \mathbf{H}_h^{(a/b)}$ be the global interpolation operator in $\mathbf{H}_h^{(a/b)}$ which is determined by

$$F_\Gamma(i_h^{(a/b)}\mathbf{v}) = F_\Gamma(\mathbf{v}) \quad \forall \Gamma \subset \partial\mathbf{T}_h. \tag{9}$$

Unfortunately, on general non-uniform meshes the optimal order estimates do not hold for $i_h^{(a/b)}$. This is due to the fact that the spaces $\mathbf{H}_h^{(a/b)}$ are not *isoparametric*, i.e. the bilinear transformations ψ_T: $\hat{T} \to T$ are of another polynomial type than the shape functions on $\hat{T}$. In order to guarantee proper approximation properties for $\mathbf{H}_h^{(a/b)}$, we have to impose a certain weak uniformity condition on the meshes $\mathbf{T}_h$. For each element $T \in \mathbf{T}_h$ let $\alpha_T \in (0, \pi]$ denote the maximum angle enclosed between the normal unit vectors corresponding to any opposite edges of T (see Figure 2). Then the quantity

$$\sigma_h := \max\{|\pi - \alpha_T|, \forall T \in \mathbf{T}_h\} \tag{10}$$

is a measure of the degeneration of the mesh $\mathbf{T}_h$.

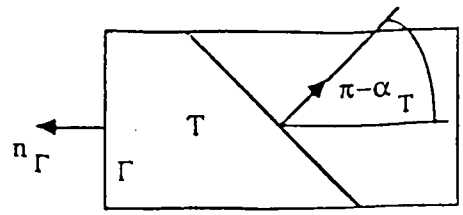

Figure 2. Deterioration of quadrilaterals

Lemma 1

For the interpolation operators $i_h = i_h^{(a/b)}$ the following error estimate holds:[4,5]

$$\|v - i_h v\|_0 + h\|v - i_h v\|_h \leqslant ch(h + \sigma_h)\|v\|_2 \quad \forall v \in \mathrm{H}_0^1(\Omega) \cap H^2(\Omega).$$

Lemma 2

For the corresponding *non-parametric* counterparts without any transformation on to the reference element the following optimal error estimate for $i_h = i_h^{(a/b)}$ holds:[4,5]

$$\|v - i_h v\|_0 + h\|v - i_h v\|_h \leqslant ch^2\|v\|_2 \quad \forall v \in H_0^1(\Omega) \cap H^2(\Omega), \tag{11}$$

without any dependence on σ_h.

Analogously as in the triangular case,[2] it is easy to see that together with the continuous estimate (BB) the uniform stability result holds for the pair $(\mathbf{H}_h^{(a)}, L_h)$:

$$\tilde{\beta}\|p_h\|_0 \leqslant \max_{\mathbf{v}_h \in \mathbf{H}_h^{(a)}} \frac{b_h(p_h, \mathbf{v}_h)}{\|\mathbf{v}_h\|_h}. \tag{12}$$

The 'midpoint-oriented' space $\mathbf{H}_h^{(b)}$ generally does not satisfy the stability property; we have to require the meshes $\mathbf{T}_h$ to be sufficiently uniform.[4,5]

Lemma 3

Suppose that the quantity $\sigma = \sup_{h>0} \sigma_h$ is sufficiently small. Then the stability estimate (12) holds true also for the pairing $(\mathbf{H}_h^{(b)}, L_h)$.

On the basis of our stability estimate (12) and the approximation property of Lemma 1, we can derive the asymptotic error estimates for the *parametric* case.[4,5]

Lemma 4

Suppose that the preceding assumptions hold. Then for $\mathbf{H}_h = \mathbf{H}_h^{(a)}$ and if the quantity $\sigma = \sup_{h>0} \sigma_h$ is sufficiently small also for $\mathbf{H}_h = \mathbf{H}_h^{(b)}$, the discrete Stokes problems have unique solutions $\{\mathbf{u}_h, p_h\} \in \mathbf{H}_h^{(a/b)} \times L_h$ and further there hold

$$\begin{aligned} \|\mathbf{u} - \mathbf{u}_h\|_h + \|p - p_h\|_0 &\leqslant c(h + \sigma_h)\{\|\mathbf{u}\|_2 + \|p\|_1\}, \\ \|\mathbf{u} - \mathbf{u}_h\|_0 + \|p - p_h\|_{-1} &\leqslant c(h + \sigma_h)^2\{\|\mathbf{u}\|_2 + \|p\|_1\}. \end{aligned} \tag{13}$$

These results indicate that the convergence of the *parametric* rotated bilinear Stokes elements for $h \to 0$ requires the underlying meshes $\{\mathbf{T}_h\}_h$ to be asymptotically uniform in the sense

that $\sigma_h = \max\{|\pi - \alpha_T|, \forall T \in \mathbf{T}_h\} \to 0$ as $h \to 0$. This conclusion is supported by our extensive numerical tests.[4,5] In fact, the condition for convergence is not very restrictive, since it is, for instance, automatically satisfied by 'weakly' uniform meshes, which we obtain when using our systematic grid refinement process described before, the usual multigrid approach. Therefore very complex domains are also admitted.

When working with the 'midpoint-oriented' finite element space $\mathbf{H}_h^{(b)}$, it is convenient to replace the bilinear form $b_h(\cdot, \cdot)$ by its numerically integrated version

$$\tilde{b}_h(q_h, \mathbf{v}_h) := -\sum_{T \in \mathbf{T}_h} q_{h|T} \sum_{\Gamma \subset \partial T} |\Gamma| F_\Gamma^{(b)}(\mathbf{v}_h) \cdot \mathbf{n}_\Gamma \approx -\sum_{T \in \mathbf{T}_h} q_{h|T} \sum_{\Gamma \subset \partial T} \oint_\Gamma \mathbf{v}_h \cdot \mathbf{n}_\Gamma \, d\gamma. \tag{14}$$

In this case the uniform stability condition is satisfied without any additional condition on the meshes $\{\mathbf{T}_h\}$. Then, by some standard perturbation arguments,[3] the estimates of Lemma 4 carry over to this case without any condition on the size of σ_h.

The *non-parametric* versions of the spaces $\mathbf{H}_h^{(a/b)}$ have satisfactory approximation properties on general regular meshes. The stability properties are the same as those of their *parametric* counterparts, i.e. the optimal order convergence estimates for $\mathbf{H}_h^{(b)}$ can be guaranteed only when using the modified bilinear form $\tilde{b}_h(\cdot, \cdot)$. Then we get for $\mathbf{H}_h^{(a)}$ and also for $\mathbf{H}_h^{(b)}$ the final results[4]

$$\begin{aligned} \|\mathbf{u} - \mathbf{u}_h\|_h + \|p - p_h\|_0 &\leqslant ch\{\|\mathbf{u}\|_2 + \|p\|_1\}, \\ \|\mathbf{u} - \mathbf{u}_h\|_0 + \|p - p_h\|_{-1} &\leqslant ch^2\{\|\mathbf{u}\|_2 + \|p\|_1\}. \end{aligned} \tag{15}$$

Now we will show how to construct the *discretely divergence-free* subspaces corresponding to the proposed finite element spaces $\mathbf{H}_h^{(a/b)}$. For this we introduce the modified discrete bilinear form $\tilde{b}_h(\cdot, \cdot)$ where

$$\tilde{b}_h(q_h, \mathbf{v}_h) := -\sum_{T \in \mathbf{T}_h} q_{h|T} Q_T(\mathbf{v}_h), \qquad Q_T(\mathbf{v}_h) := \sum_{\Gamma \subset \partial T} |\Gamma| F_\Gamma(\mathbf{v}_h) \cdot \mathbf{n}_\Gamma, \tag{16}$$

which is for $\mathbf{H}_h^{(b)}$ an $O(h^2)$-approximation to the original bilinear form $b_h(\cdot, \cdot)$. Then we call a function $\mathbf{v}_h \in \mathbf{H}_h$ discretely divergence-free if the condition

$$\tilde{b}_h(q_h, \mathbf{v}_h) = 0 \quad \forall q_h \in L_h \tag{17}$$

is satisfied. Since we only use piecewise constant pressure approximations, an equivalent criterion is

$$Q_T(\mathbf{v}_h) = 0 \quad \forall T \in \mathbf{T}_h. \tag{18}$$

With these modifications we can introduce subspaces $\mathbf{H}_h^d \subset \mathbf{H}_h$ and our discrete problem for the velocity is reduced to

find $\mathbf{u}_h^d \in \mathbf{H}_h^d$ such that

$$a_h(\mathbf{u}_h^d, \mathbf{v}_h^d) = (\mathbf{f}, \mathbf{v}_h^d) \quad \forall \mathbf{v}_h^d \in \mathbf{H}_h^d. \tag{$\mathbf{V}_h^d$}$$

Finally, the corresponding pressure $p_h \in L_h$ is determined by the condition

$$\tilde{b}_h(p_h, \mathbf{v}_h^r) = (\mathbf{f}, \mathbf{v}_h^r) - a_h(\mathbf{u}_h^d, \mathbf{v}_h^r) \quad \forall \mathbf{v}_h^r \in \mathbf{H}_h^r, \tag{19}$$

where the functions $\mathbf{v}_h^r$ span the curl-free part of the complete space $\mathbf{H}_h$. In our configuration this is performed[4] by a marching process from element to element without solving any linear

system of equations. Then for the solution $\mathbf{u}_h^d$ of problem (V_h^d) and its corresponding pressure p_h there hold again the error estimates

$$\begin{aligned} \|\mathbf{u} - \mathbf{u}_h^d\|_h + \|p - p_h\|_0 &\leqslant ch\{\|\mathbf{u}\|_2 + \|p\|_1\}, \\ \|\mathbf{u} - \mathbf{u}_h^d\|_0 + \|p - p_h\|_{-1} &\leqslant ch^2\{\|\mathbf{u}\|_2 + \|p\|_1\}. \end{aligned} \tag{20}$$

Now consider a general quadrilateral $T \in \mathbf{T}_h$ (see Figure 3) with vertices a^i, midpoints m^j, edges Γ^j, unit tangential vectors $\mathbf{t}^j$ and unit normal vectors $\mathbf{n}^j$. Let $\varphi_h^j \in S_h^{(a/b)}$ be the usual nodal basis functions of the finite element space $S_h = S_h^{(a/b)}$ (see (9)), restricted to the element T, satisfying $F_{\Gamma^i}(\varphi_h^j) = \delta_{ij}$, $i, j = 1, \ldots, 4$. Then the first group of basis functions $\{\mathbf{v}_h^{i,t}\}$ of $\mathbf{H}_h^d$, corresponding to the edges of $\mathbf{T}_h$, is given by the local definition

$$\mathbf{v}_{h|T}^{i,t} \in \{\varphi_h^j \mathbf{t}^j, j = 1, \ldots, 4\}. \tag{21}$$

The second group $\{\mathbf{v}_h^{i,\psi}\}$, corresponding to the vertices, is locally determined by

$$\mathbf{v}_{h|T}^{i,\psi} \in \left\{ \frac{\varphi_h^k \mathbf{n}^k}{|\Gamma^k|} - \frac{\varphi_h^j \mathbf{n}^j}{|\Gamma^j|}, j = 1, \ldots, 4, k = (j+2) \bmod 4 + 1 \right\}. \tag{22}$$

Thus we get approximations for the tangential velocities on the edges and for the stream-function values at the nodes.[4,6] If we eliminate one of the functions $\{\mathbf{v}_h^{i,\psi}\}$ by prescribing the value at one (boundary) point or by the zero-mean-value condition, we get a basis for the discretely divergence-free subspace $\mathbf{H}_h^d$ assuming that our problem has only one boundary component. For several boundary components (e.g. for flows around obstacles) we do not need any additional basis function, but only have to modify our linear solvers in a simple way (by *projection methods*[4,7]).

After introducing these new basis functions, the size of our linear system is reduced from about 5 *nel* unknowns (*nel* is the number of elements) for the usual formulation to about 3 *nel* in the divergence-free case. A disadvantage of the new formulation is that the corresponding stiffness matrix S_d, with

$$S_d^{(i,j)} = a_h(\mathbf{v}_h^{i,d}, \mathbf{v}_h^{j,d}), \tag{23}$$

has a condition number like $O(h^{-4})$,[4] while the corresponding mass matrix has a condition number like $O(h^{-2})$. However, since we were able to derive a multigrid algorithm with convergence rates independent of h, this fact is unimportant for our work.

Our corresponding multigrid algorithm is developed for linear (unsymmetric) problems, with a given coefficient function $\mathbf{U}$, of the type

$$\alpha\mathbf{u} - \nu\Delta\mathbf{u} + (\mathbf{U} \cdot \nabla)\mathbf{u} + \nabla p = \mathbf{f}, \qquad \nabla \cdot \mathbf{u} = 0, \tag{24}$$

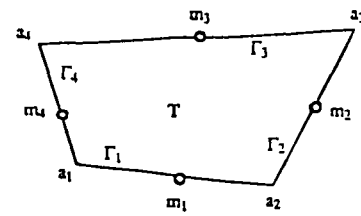

Figure 3. General quadrilateral T

which arise from the linearization of the non-stationary Navier–Stokes equations. The components of the algorithm are the usual ones: some smoothing steps with an appropriate *smoothing operator*, a *coarse grid correction* with appropriate grid transfer operators, and a *step length control*, which is usual for non-conforming finite elements.[4,7] Then for the positive definite case we can show that the convergence rates are independent of the mesh size h.[4] In all our computations (with an F-cycle) we use as *smoothing operator* two pre- and post-smoothing steps with the Gauss–Seidel method, and for the grid transfer a *macro-elementwise divergence-free interpolation* which interpolates a given discretely divergence-free frunction on level $2h$ on to level h, again discretely divergence-free.[4,7] To explain this essentially new idea, we have to go into a little more detail. Figure 4 shows one coarse element on level $2h$ and the corresponding refined elements on level h. The *macro-elementwise interpolation*, which also works analogously for the scalar non-conforming and the Morley space,[7] goes something like the following.

1. Transfer the divergence-free coefficient vector (Ψ_{2h}, Ut_{2h}) into the primitive coefficient vector (U_{2h}, V_{2h}).
2. Interpolate 'fully' on the macro-element to get (U_h, V_h).
3. Compute Ut_h and Un_h on all fine grid edges.
4. Set $\Psi_h = \Psi_{2h}$ at the macro-nodes and compute at the new vertices the values for Ψ_h by integrating Un_h.
5. Take the average of Ψ_h and Ut_h, which lie on macro-edges.

These operations, using *local transfer matrices*, can be performed very quickly and efficiently. In fact, solving a Stokes equation takes about the same amount of time as solving a Lapalce equation. After all our extensive numerical tests[4,7] we can state that this proposed combination is a very efficient and robust solver for linear problems of the above type, indpendent of all given data, domain and triangulation. Thus we can really claim to have a 'black box' solver, at least for the linear problems arising in the discretization process for the fully non-stationary Navier–Stokes equations. In the next section we will describe the discretization scheme for the non-linear and non-stationary Navier–Stokes equations and discuss some of its properties.

3. THE NAVIER–STOKES EQUATIONS AND THEIR DISCRETIZATION

Our formulation of the stationary Navier–Stokes equations reads as

$$-\nu\Delta\mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p = \mathbf{f}, \quad \nabla\cdot\mathbf{u} = 0 \quad \text{in } \Omega, \qquad \mathbf{u} = \mathbf{g} \quad \text{on } \partial\Omega, \tag{25}$$

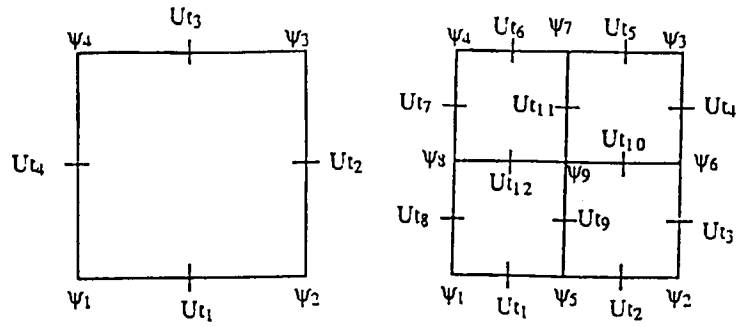

Figure 4. Macro-element and refined elements

and respectively the variational form[2]

find $\mathbf{u} \in \mathbf{V}(\Omega)$ such that

$$va(\mathbf{u}, \mathbf{v}) + n(\mathbf{u}, \mathbf{u}, \mathbf{v}) = (\mathbf{f}, \mathbf{v}) \quad \forall \mathbf{v} \in \mathbf{V}(\Omega), \tag{NS^d}$$

with the trilinear form

$$n(\mathbf{u}, \mathbf{v}, \mathbf{w}) := \int_\Omega u_i \frac{\partial v_j}{\partial x_i} w_j \, dx. \tag{26}$$

Introducing the discrete form $n_h(\cdot, \cdot, \cdot)$, where

$$n_h(\mathbf{u}_h, \mathbf{v}_h, \mathbf{w}_h) := \sum_{T \in \mathbf{T}_h} \int_T u_{h,i} \frac{\partial v_{h,j}}{\partial x_i} w_{h,j} \, dx, \tag{27}$$

and using the bilinear forms $a_h(\cdot, \cdot)$ and $b_h(\cdot, \cdot)$ from Section 2, our discrete problem is

find $\mathbf{u}_h^d \in \mathbf{H}_h^d$ such that

$$va_h(\mathbf{u}_h^d, \mathbf{v}_h^d) + n_h(\mathbf{u}_h^d, \mathbf{u}_h^d, \mathbf{v}_h^d) = (\mathbf{f}, \mathbf{v}_h^d) \quad \forall \mathbf{v}_h^d \in \mathbf{H}_h^d. \tag{NS$_h^d$}$$

The corresponding pressure p_h can again be computed by simple postprocessing.

Since this central discretization of the convection part leads to stiffness matrices not of positive type (M-matrices) and to numerical oscillations of the solution, we use an upwind discretization based on the works of Ohmori and Ushijima[8] and Tobiska and Schieweck.[9] The main idea is to introduce some edge-oriented lumping regions and lumping operators. For this we divide each quadrilateral $T \in \mathbf{T}_h$ into eight barycentric fragments S_{ij} and define for each edge Γ_l and corresponding midpoint m_l the lumping region R_l (see Figure 5) given by

$$R_l := \bigcup_{k \in \Delta_l} S_{lk}, \tag{28}$$

where Λ_l is the set of indices k such that m_l and m_k are neighbouring midpoints. Defining the edge $\Gamma_{lk} := \partial S_{lk} \cap \partial S_{kl}$, the boundary ∂R_l of the region R_l can be written as

$$\partial R_l = \bigcup_{k \in \Lambda_l} \Gamma_{lk} \tag{29}$$

and we achieve a new (edge-oriented) partition of $\bar{\Omega} = \cup_{T \in \mathbf{T}_h} \bar{T}$ as

$$\bar{\Omega} = \bigcup_l \bar{R}_l. \tag{30}$$

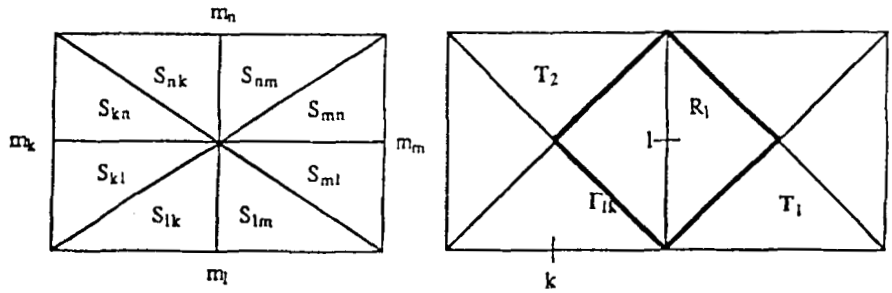

Figure 5. Barycentric fragments S_{ij} of T and lumping regions R_l around midpoint l

Now, defining the piecewise constant lumping operator L_h, $(L_h \mathbf{v}_h)(x) = \mathbf{v}_h(m_l)$, $\forall x \in R_l$, this results in

$$\|\mathbf{v}_h\|_0 \sim \|L_h \mathbf{v}_h\|_0, \quad \|L_h \mathbf{v}_h - \mathbf{v}_h\|_0 \leqslant ch\|\mathbf{v}_h\|_h \quad \forall \mathbf{v}_h \in \mathbf{H}_h. \tag{31}$$

Writing $n_h(\mathbf{u}_h, \mathbf{v}_h, \mathbf{w}_h)$ as the sum $n_h = n_h^1 + n_h^2$, with

$$n_h^1(\mathbf{u}_h, \mathbf{v}_h, \mathbf{w}_h) := \sum_{T \in \mathcal{T}_h} \int_T \frac{\partial u_{h,i} v_{h,j}}{\partial x_i} w_{h,j} \, \mathrm{d}x, \tag{32}$$

$$n_h^2(\mathbf{u}_h, \mathbf{v}_h, \mathbf{w}_h) := \sum_{T \in \mathcal{T}_h} \int_T \frac{\partial u_{h,i}}{\partial x_i} v_{h,j} w_{h,j} \, \mathrm{d}x, \tag{33}$$

we replace $n_h^1(\mathbf{u}_h, \mathbf{v}_h, \mathbf{w}_h)$ by $n_h^1(\mathbf{u}_h, \mathbf{v}_h, L_h \mathbf{w}_h)$ and $n_h^2(\mathbf{u}_h, \mathbf{v}_h, \mathbf{w}_h)$ by $n_n^2(\mathbf{u}_h, L_h \mathbf{v}_h, L_h \mathbf{w}_h)$. Then we modify[8] n_h^1 by an upwinded form (replace $\mathbf{v}_h$ by $\mathbf{v}_h^{lk}$), apply the Gauss theorem and the result is the terms

$$\tilde{n}_h^1(\mathbf{u}_h, \mathbf{v}_h, \mathbf{w}_h) := \sum_l \sum_{k \in \Delta_l} \oint_{\Gamma_{lk}} (\mathbf{u}_h \cdot \mathbf{n}_{lk}) \mathbf{v}_h^{lk} \, \mathrm{d}\gamma \mathbf{w}_h(m_l), \tag{34}$$

$$\tilde{n}_h^2(\mathbf{u}_h, \mathbf{v}_h, \mathbf{w}_h) := \sum_l \sum_{k \in \Lambda_l} \oint_{\Gamma_{lk}} \mathbf{u}_h \cdot \mathbf{n}_{lk} \, \mathrm{d}\gamma \mathbf{v}_h(m_l) \mathbf{w}_h(m_l), \tag{35}$$

with

$$\mathbf{v}_h^{lk} := \lambda_{lk} \mathbf{v}_h(m_l) + (1 - \lambda_{lk}) \mathbf{v}_h(m_k), \tag{36}$$

where the functions λ_{lk} are

$$\lambda_{lk} = 1 - \lambda_{kl}, \qquad |\lambda_{lk}| \leqslant c. \tag{37}$$

Finally, the new form $\tilde{n}_h(\mathbf{u}_h, \mathbf{v}_h, \mathbf{w}_h)$ is defined as

$$\tilde{n}_h(\mathbf{u}_h, \mathbf{v}_h, \mathbf{w}_h) := \sum_l \sum_{k \in \Lambda_l} \oint_{\Gamma_{lk}} \mathbf{u}_h \cdot \mathbf{n}_{lk} \, \mathrm{d}\gamma (1 - \lambda_{lk}(\mathbf{u}_h))(\mathbf{v}_h(m_k) - \mathbf{v}_h(m_l)) \mathbf{w}_h(m_l). \tag{38}$$

We used the following possibilities for λ_{lk} (with $x := (1/\nu) \oint_{\Gamma_{lk}} \mathbf{u}_h \cdot \mathbf{n}_{lk} \, \mathrm{d}\gamma$ as measure for the local Reynolds number).

(i) Simple upwind:

$$\lambda_{lk}(\mathbf{u}_h) := \begin{Bmatrix} 1 & \text{if } x \geqslant 0 \\ 0 & \text{otherwise} \end{Bmatrix}.$$

(ii) Samarskij upwind:[10]

$$\lambda_{lk}(\mathbf{u}_h) := \begin{Bmatrix} (\frac{1}{2} + x)/(1 + x) & \text{if } x \geqslant 0 \\ 1/2(1 - x) & \text{otherwise} \end{Bmatrix}.$$

For the following analysis we restrict ourselves to the case of the simple upwind scheme. Then our discrete problem reads

find $\mathbf{u}_h^d \in \mathbf{H}_h^d$ such that

$$\nu a_h(\mathbf{u}_h^d, \mathbf{v}_h^d) + \tilde{n}_h(\mathbf{u}_h^d, \mathbf{u}_h^d, \mathbf{v}_h^d) = (\mathbf{f}, \mathbf{v}_h^d) \quad \forall \mathbf{v}_h^d \in \mathbf{H}_h^d. \tag{$\widetilde{\mathrm{NS}}_h^d$}$$

Lemma 5

The following hold[4] for functions in $\mathbf{H}_h$.

(i) $\tilde{n}_h(\mathbf{u}_h^d, \mathbf{v}_h, \mathbf{v}_h) \geqslant 0$, $\forall \mathbf{u}_h^d \in \mathbf{H}_h^d$, $\forall \mathbf{v}_h \in \mathbf{H}_h$ (positivity).
(ii) $|\tilde{n}_h(\mathbf{u}_h^0, \mathbf{v}_h, \mathbf{w}_h) - \tilde{n}_h(\mathbf{u}_h^1, \mathbf{v}_h, \mathbf{w}_h)| \leqslant c\|\mathbf{u}_h^0 - \mathbf{u}_h^1\|_h \|\mathbf{v}_h\|_h \|\mathbf{w}_h\|_h$ (continuity).
(iii) $|n_h(\mathbf{u}_h, \mathbf{v}_h, \mathbf{w}_h) - \tilde{n}_h(\mathbf{u}_h, \mathbf{v}_h, \mathbf{w}_h)| \leqslant ch|\log h|\, \|\mathbf{u}_h\|_h \|\mathbf{v}_h\|_h \|\mathbf{w}_h\|_h$ (approximation).
(iv) For $\mathbf{f} \in L^2(\Omega)$ there exists at least one solution $\{\mathbf{u}_h^d, p_h\} \in \mathbf{H}_h^d \times L_h$ (existence).

For the weighted upwind scheme (Samarskij upwind) it can be shown in one dimension[10] that the resulting discretization is of second-order accuracy, but the corresponding result in two dimensions is still an open problem. For the simple upwind it can be shown that using the primitive function spaces $\mathbf{H}_h$ and L_h, the corresponding stiffness matrix S_h, with

$$S_h^{(ij)} := va_h(\mathbf{v}_h^{i,d}, \mathbf{v}_h^{j,d}) + \tilde{n}_h(\mathbf{u}_h^n, \mathbf{v}_h^{i,d}, \mathbf{v}_h^{j,d}), \tag{39}$$

is an M-matrix,[8,9] which results in very nice linear algebraic properties concerning convergence results for the jacobi, SOR or ILU method. This is essential to our solution method.

For solving the discrete problems ($\widetilde{\mathrm{NS}}_h^d$), we first introduce matrices A_h and $N_h(U_h)$, corresponding to the bilinear and trilinear forms respectively, and reformulate our problem as

$$\begin{aligned} &\text{find } \mathbf{u}_h^d = \sum U_h^i \mathbf{v}_h^{i,d} \in \mathbf{H}_h^d \text{ such that} \\ &\qquad vA_hU_h + N_h(U_h)U_h = F_h. \end{aligned} \tag{NS_h^d}$$

Then we use a *fixed point defect correction method*, given a starting vector U_h^0 defined by

$$U_h^{n+1} = U_h^n + \omega_h[vA_h + \tilde{N}_h(U_h^n)]^{-1}(F_h - vA_hU_h^n - N_h(U_h^n)U_h^n). \tag{40}$$

For the term $N_h(U_h^n)$ in the defect computation we can use one of the proposed upwind discretizations or the central one in (27), while for solving linear problems of the Oseen type,

$$-v\Delta\mathbf{u} + (\mathbf{u}^n \cdot \nabla)\mathbf{u} + \nabla p = \mathrm{RHS}, \qquad \nabla \cdot \mathbf{u} = 0, \tag{41}$$

we always take one of the upwind discretizations because of their nice linear algebraic properties. The damping parameter ω_h can be chosen as a fixed value or can be computed adaptively. Letting X_h be the solution of the linear problem corresponding to $[vA_h + \tilde{N}_h(U_h^n)]^{-1}$ and letting U_h be the exact discrete solution, we try to minimize the error $U_h^n + \omega_h X_h - U_h$ in a suitable norm. If we take the defect norm, we have to minimize the non-linear functional in the Euclidean vector norm

$$\|[vA_h + N_h(U_h^n + \omega_h X_h)](U_h^n + \omega_h X_h) - F_h\|_E. \tag{42}$$

We linearize this non-linear problem by replacing $N_h(U_h^n + \omega_h X_h)$ by $N_h(U_h^n + \omega_0 X_h)$, where ω_0 is now fixed. Then the solution of the corresponding minimization problem is

$$\omega_h = \frac{(A_h^0 X_h, F_h - A_h^0 U_h^n)}{(A_h^0 X_h, A_h^0 X_h)}, \tag{43}$$

with $A_h^0 = vA_h + N_h(U_h^n + \omega_0 X_h)$.

Remark

For simulating non-Newtonian incompressible fluids,[11] we simply have to introduce instead of the linear diffusion term $v\Delta\mathbf{u}$ the non-linear term $\nabla \cdot (v(\varepsilon + |\nabla\mathbf{u}|)^{-\alpha}\nabla)\mathbf{u}$, with $\alpha \in [0, 1]$.

Therefore in (40) we only have to replace A_h by $A_h(U_h^n)$ and our *fixed point defect correction method* reads

$$U_h^{n+1} = U_h^n + \omega_h[\nu A_h(U_h^n) + \tilde{N}_h(U_h^n)]^{-1}(F_h - \nu A_h(U_h^n)U_h^n - N_h(U_h^n)U_h^n). \tag{44}$$

First numerical results for this problem can be found in Reference 11.

With this solver for the stationary Navier–Stokes solver, we will develop a corresponding fast and robust solution method for the non-stationary equations, written as

$$\mathbf{u}_t - \nu\Delta\mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p = \mathbf{f}, \quad \nabla\cdot\mathbf{u} = 0 \quad \text{in } \Omega \times (0, T),$$

$$\mathbf{u} = \mathbf{g} \quad \text{on } \partial\Omega, \qquad \mathbf{u}_{|t=0} = \mathbf{u}_0.$$

Because of the stiffness of this problem, A-stable methods are desired for the time discretization. For the analysis of the schemes used it is most instructive to look first at the scalar linear test equation

$$\dot{x}(t) + \lambda x(t) = 0, \quad t \geq 0, \tag{45}$$

where $\lambda \in \mathbf{C}$, $\mathrm{Re}\ \lambda \geq 0$. A time-stepping scheme applied to this equation, with a constant time step size Δt, generates a sequence of values $x_n \sim x(t_n)$, with $t_n := n\Delta t$. The behaviour of the scheme as $t \to \infty$, depending on the parameter λ, is usually characterized by the *amplification factor* $\omega = \omega(\lambda\Delta t)$. In particular, for the one-step schemes which we use there holds $x_n = \omega^n x_0$. In terms of ω we can formulate the following desirable properties of time-stepping schemes.[12]

(i) $|\omega(\lambda\Delta t)| \geqslant 1$ (local stability).
(ii) $\lim_{\mathrm{Re}\,\lambda\to\infty}|\omega(\lambda\Delta t)| \geqslant 1 - O(\Delta t)$ (global regularity).
(iii) $\lim_{\mathrm{Re}\,\lambda\to\infty}|\omega(\lambda\Delta t)| \geqslant 1 - \delta < 1$ (smoothing property: strong A-stability).
(iv) $|\omega(\lambda\Delta t)| \sim 1$ for $\mathrm{Re}\ \lambda = 0$ (non-dissipative).

The following one-step schemes are given in a form which applies to the general linear evolution equation

$$u_t + A(t)u = f(t), \qquad A_n = A(t_n), \qquad f_n = f(t_n). \tag{46}$$

The approximation properties can be expressed in terms of the amplification factor $\omega(\lambda\Delta t)$, where λ is an eigenvalue of $A(t)$. Now our hope is to carry these conclusions also to the non-linear Navier–Stokes equations. The following results can be found in more detail in References 4 and 12.

The one-step ϑ-schemes

$$[I + \vartheta\Delta t A_{n+1}]u_{n+1} = [I - (1-\vartheta)\Delta t A_n]u_n + \vartheta\Delta t f_{n+1} + (1-\vartheta)\Delta t f_n, \tag{47}$$

with amplification factors

$$\omega(z) = \frac{1 - (1-\vartheta)z}{1 + \vartheta z} \tag{48}$$

We exclude the explicit Euler scheme ($\vartheta = 0$) since it is only conditionally stable ($\Delta t \leqslant 1/\lambda$), of first-order accuracy, and also since the mass matrix is not spectrally equivalent to a diagonal matrix (*lumping*). Its implicit counterpart, the implicit Euler scheme ($\vartheta = 1$), is strongly A-stable ($|\omega(z)| \to 0$ for $\mathrm{Re}\ \lambda \to \infty$) but tends to damp out free oscillations ($|\omega(i\Delta t)| < 1$, $|\omega| = 0.995$ for

$\Delta t = 0.1$) and is also of first-order accuracy. The Crank–Nicolson scheme ($\vartheta = \frac{1}{2}$) is of second order but has only a low-damping property ($|\omega(z)| \to 1$ for Re $\lambda \to \infty$) and is only A-stable; however, free oscillations are well preserved ($|\omega(i\Delta t)| = 1$). Summing up, both schemes have some advantages and some disadvantages: The implicit Euler is very robust but inaccurate and strongly damping, while the Crank–Nicolson is more accurate but tends to become unstable. A scheme possessing the advantages of both of these is the following.

The fractional step ϑ-scheme

Choosing $\vartheta \in (0, 1)$, $\vartheta' = 1 - 2\vartheta$ and $\alpha \in [0, 1]$, $\beta = 1 - \alpha$, the macro-time step $t_n \to t_{n+1}$ is split into the following three substeps (see also Reference 13):

$$\begin{aligned}
[I + \alpha\vartheta\Delta t A_{n+\vartheta}]u_{n+\vartheta} &= [I - \beta\vartheta\Delta t A_n]u_n + \vartheta\Delta t f_n,\\
[I + \beta\vartheta'\Delta t A_{n+1-\vartheta}]u_{n+1-\vartheta} &= [I - \alpha\vartheta'\Delta t A_{n+\vartheta}]u_{n+\vartheta} + \vartheta'\Delta t f_{n+1-\vartheta},\\
[I + \alpha\vartheta\Delta t A_{n+1}]u_{n+1} &= [I - \beta\vartheta\Delta t A_{n+1-\vartheta}]u_{n+1-\vartheta} + \vartheta\Delta t f_{n+1},
\end{aligned} \tag{49}$$

with

$$\omega(z) = \frac{(1 - \beta\vartheta z)^2(1 - \alpha\vartheta' z)}{(1 + \alpha\vartheta z)^2(1 + \beta\vartheta' z)}. \tag{50}$$

We always perform the macro-time step $t_n \to t_{n+3\Delta t}$. Then three steps of the one-step schemes and the three substeps of the fractional step scheme lead to the same time and the step sizes, costs and discretization errors are comparable.

For the special choice $\vartheta = 1 - \sqrt{2}/2$ this scheme is of second order. Taking $\alpha = (1 - 2\vartheta)/(1 - \vartheta)$, the coefficient matrices are the same in all substeps. Further, the scheme is strongly A-stable ($\lim_{\mathrm{Re}\,\lambda \to \infty} |\omega(z)| = \beta/\alpha \sim 0.7$) and free oscillations are well preserved ($|\omega(i\Delta t)| \sim 0.9998$ for $\Delta t = 0.8$). These theoretical results will be (numerically) verified by the following extensive test calculations in the next section.

Applying these ϑ-schemes to our fixed point defect correction method for the solution of the Navier–Stokes equations, we have to solve non-linear systems of equations at each time step of the following type:

$$\begin{aligned}
[M_h + \alpha\vartheta\Delta t(\nu A_h + N_h(U_{h,n+1}))]U_{h,n+1} = {} & [M_h - \beta\vartheta'\Delta t(\nu A_h + N_h(U_{h,n}))]U_{h,n}\\
& + \vartheta\Delta t F_{h,n+1} + \vartheta'\Delta t F_{h,n},
\end{aligned} \tag{51}$$

for some constants α, β, ϑ and ϑ', depending on the actual method and the discretization $N_h(\cdot)$ of the convection part. Consequently, in each substep we have to solve a non-linear problem

$$[M_h + c\nu\Delta t A_h + c\Delta t N_h(X_h)]X_h = \mathrm{RHS}, \tag{52}$$

which is easily done using the proposed method for the stationary Navier–Stokes equations.

Thus we think that we have developed a very robust method for the fully non-stationary equations, consisting of the fractional step ϑ-scheme for the time discretization, the fixed point defect correction method for the non-linear problems, an upwind discretization for the convection terms, and a multigrid algorithm, adapted to the discretely divergence-free finite elements, for the linear non-symmetric subproblems. This is the candidate for our 'FRA black box' solver, but we have to prove all conditions in the following test calculations.

4. NUMERICAL RESULTS

In this section we will discuss some questions concerning our proposed FRA solver:

(i) a comparison of the influence of different discretization schemes in space and time w.r.t. the accuracy, robustness and efficiency
(ii) the influence of the 'semi-adaptively' chosen grids
(iii) the question of the 'right' (artificial) boundary conditions (for many more details concerning this question see Reference 14.

Before we start with the *presentation* of the results, we have to say something about the visualization techniques used, since this is a very important factor in fluid dynamics simulations (for more details see Reference 15). For *stationary* pictures we use the well-known (and inexpensive) graphics package MOVIE.BYU. This package works well for tasks such as plotting of streamlines, plotting of vectors, presentation of the pressure in 3D, coloured norms of velocity and pressure, and finding *reattachment points* or vortex separation zones. Additionally, new routines are needed for the visualization of *non-stationary* data. One reason is that for simulations of more complex time-dependent flows, even thinking of quasi-turbulent processes, only movies can show 'all' relevent information. Therefore we developed a fully interactive particle-tracing tool[15] and a correspondingly suited data structure, which creates movies for the computer screen or video tapes, comparable with 'real life' experiments. Figure 6 demonstrates the importance of these *non-stationary* tools by visualizing a vortex street behind an obstacle in several different ways. The five pictures represent the same flow after 150 time steps: with absolute and relative streamlines (after subtracting the Stokes flow, first row), with absolute and relative velocities (second row), and with the corresponding particle-tracing tool. These figures clearly demonstrate the importance of the 'right' visualization. For our test calculations we chose the following types: *driven cavity* and *backward-facing step* type (because of their simple geometry, both are well known in mathematical works), flow around obstacles, flow in a *venturi pipe*, flow through a hole in an 'infinite' space, and 'quasi-turbulent' jet flow in a box. For 'mathematically historical' reasons we start with the usual (lid-) *driven cavity* problem on the unit square:

$$-\nu\Delta\mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p = 0, \quad \nabla\cdot\mathbf{u} = 0 \quad \text{in } \Omega,$$

$$\mathbf{u} = \mathbf{0} \quad \text{on } \partial\Omega\backslash\{y = 1\}, \qquad \mathbf{u} = (1, 0)^T \quad \text{on } \partial\Omega \cap \{y = 1\}.$$

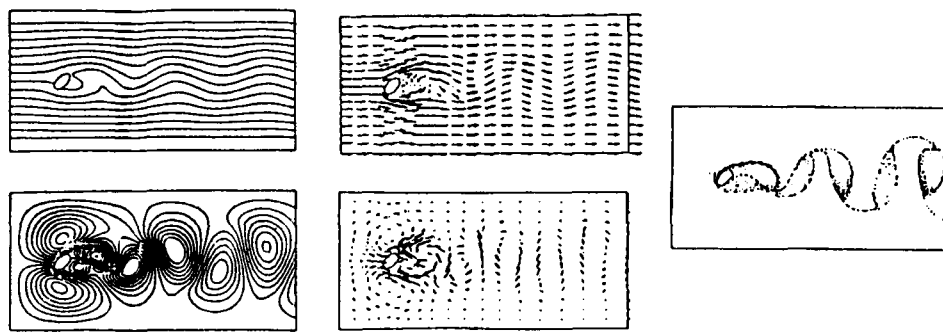

Figure 6. Different visualizations for the *Karman* vortex street ($Re = 500$)

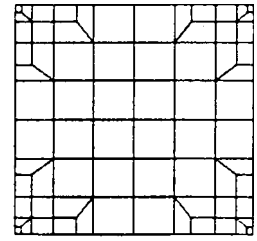

Figure 7. *Semi-adaptive* coarse grid for *driven cavity*

Table I. Values for *driven cavity* on *semi-adaptive* grids

	Centre vortex		Right vortex	
	Ψ	(x, y)	Ψ	(x, y)
		Four refinements		
		Five refinements		
C	−0·090	(0·521, 0·542)	0·213 − 2	(0·856, 0·092)
S	−0·137	(0·531, 0·553)	0·225 − 2	(0·851, 0·098)
SC	−0·123	(0·510, 0·542)	0·224 − 2	(0·851, 0·098)
		Comparisons		

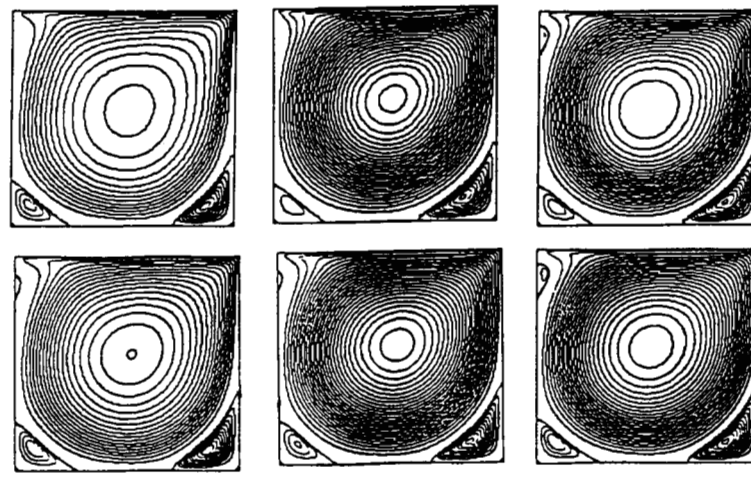

Figure 8. Streamlines (*semiadaptive*) for C, S and SC schemes: first row on level 4, second row on level 5

Figure 7 shows the 'semi-adaptive' coarse grid used, which is refined four times (4801 vertices, 14,209 unknowns, $h_{min} = 1/250$, $h_{max} = 1/48$) or five times (18,817 vertices, 56,065 unknowns, $h_{min} = 1/500$, $h_{max} = 1/96$.

We choose $Re = 2000$, since for this Reynolds number range (see e.g. References 16 and 17) a 'new' secondary eddy appears at the upper left edge. In Table I the values and co-ordinates are given for the main vortex and the secondary vortex in the lower right corner and are compared with the values of Vanka (V)[18] and Zhang (Z)[19] (both on rectangular grids). We show the solutions of the central discretization (C) (by our defect correction method; see (40)), the Samarskij upwind discretization (S) and additionally the upwind solution with (only) one subsequent central defect correction step (SC). The corresponding pictures can be found in Figure 8. For comparison, in Table II the corresponding results on an equidistant grid are given for $1/h = 96$, 128 and 192 and in Figure 9 the resulting plots.

The first observations resulting from these and many other tests[4] are the following.

1. Only on equidistant grids and exclusively for low Reynolds numbers do the central schemes seem to be of higher accuracy than the upwind schemes.
2. Using *semi-adaptive* grids adapted to the expected separation zones, much better results can be achieved by the upwind schemes. For the central schemes additional refinement is needed for numerical stabilization in regions of higher velocity. However, these schemes are so unstable that even one subsequent defect correction step can cause bad results.
3. Even for a mesh size of $1/h = 192$ we are still far away from the continuous solution (because of large changes in the streamline values), so it seems to be impossible for this code to compute reference solutions for this range of Reynolds numbers using mid-range work-stations.

Some of these observations are confirmed by the results for $Re = 10{,}000$ given in Table III

Table II. Values for *driven cavity* on *equidistant* grids

	Centre vortex		Right vortex	
	Ψ	(x, y)	Ψ	(x, y)
		Mesh size 1/h = 96		
C	−0·086	(0·520, 0·552)	0·205 − 2	(0·854, 0·093)
S	−0·146	(0·520, 0·562)	0·233 − 2	(0·843, 0·104)
SC	−0·129	(0·510, 0·552)	0·245 − 2	(0·843, 0·104)
		Mesh size 1/h = 128		
C	−0·096	(0·523, 0·546)	0·219 − 2	(0·851, 0·101)
S	−0·145	(0·523, 0·554)	0·246 − 2	(0·843, 0·101)
SC	−0·130	(0·515, 0·554)	0·251 − 2	(0·843, 0·101)
		Mesh size 1/h = 192		
C	−0·107	(0·520, 0·546)	0·233 − 2	(0·848, 0·098)
S	−0·142	(0·517, 0·546)	0·251 − 1	(0·843, 0·098)
SC	−0·127	(0·520, 0·546)	0·256 − 2	(0·838, 0·093)

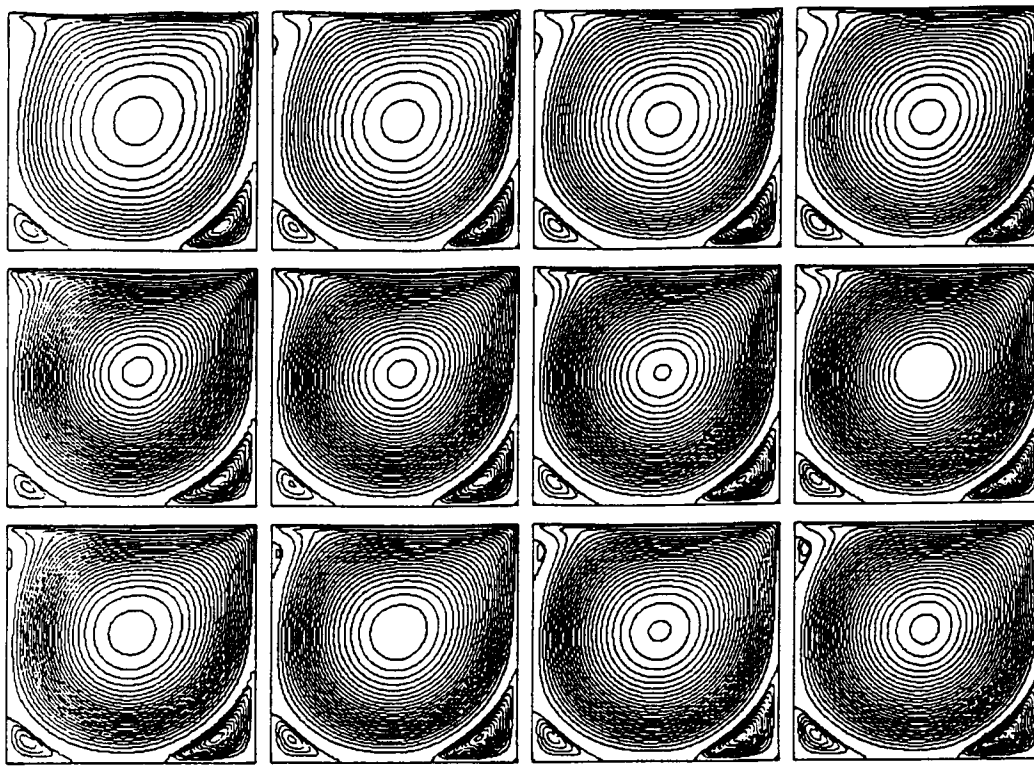

Figure 9. Streamlines (*equidistant*) for C, S and SC schemes: row-wise $1/h = 64, 96, 128, 192$

Table III. Values on *semi-adaptive* grid for $Re = 10{,}000$ (level 5)

	Centre vortex		Right vortex	
	Ψ	(x, y)	Ψ	(x, y)
C	−0·038	(0·531, 0·510)	0·139 − 2	(0·839, 0·041)
S	−0·126	(0·521, 0·542)	0·368 − 2	(0·808, 0·070)
		Comparisons		
G	−0·119	(0·511, 0·533)	0·342 − 2	(0·765, 0·058)
Z	−0·126	(0·511, 0·531)	0·308 − 2	(0·757, 0·058)

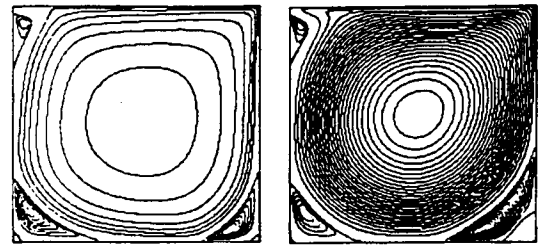

Figure 10. Streamlines for C and S schemes ($Re = 10{,}000$)

and plotted in Figure 10, which show very unphysical vortices for the central scheme (compared with Z[19] and Ghia *et al.* (G)[16]).

We also made some non-stationary computations which highlight some very interesting (in)stability properties of the Crank–Nicolson scheme. We computed the flow for $Re = 15{,}000$ starting with a low-Reynolds-number solution and using the time step size $\Delta t = 0.1$. After some time small oscillations appeared near the right upper corner. At first they did not influence the overall solution, but after many more steps the solution exploded (see Figure 11). For the same input data the implicit Euler scheme and the fractional step scheme performed well. The reason for this was our fault, since the non-linear system at each time step was not solved with enough accuracy. We always performed 10 steps (maximum) with our fixed point iteration method (40) without damping ($\omega_h = 1$) to gain one digit per iteration. This procedure always worked well (and still works fine) for the non-steady equations, only in case the solution blew up. Thus we think that this may be an interesting pointer to the (n) instabilities of the Crank–Nicolson scheme which may appear in a long-time calculation, especially if started with rough initial values.

Some other very interesting computations concerning *driven cavity* flows (perhaps more appropriate to physical experiments) are the following. Figure 12 shows the results of computing the flow in a long pipe with a cavity inside for a high Reynolds number. The length of the cavity is $L = 1$, the maximum pipe inflow velocity is $U = 1$ and the viscosity is $1/\nu = 100{,}000$. Since we are only interested in the effects in the cavity, we restricted ourselves to the computational domain shown and used the 'do nothing' outflow boundary condition.[14] We reached a 'periodically' oscillating flow (presented here in eight characteristic states) with a very remarkable velocity profile for $y = 1$. The maximum velocity in the cavity is about 0·04, which leads to a characteristic (cavity) Reynolds number of 4000. It is obvious that these more physical boundary conditions lead to a velocity profile at the line $y = 1$ not comparable with the (usual) *lid-driven cavity*.

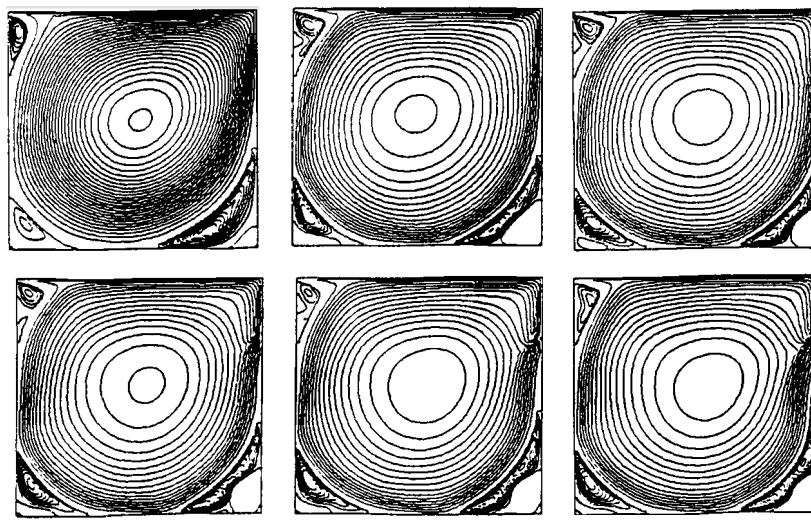

Figure 11. Crank–Nicolson scheme for $Re = 15{,}000$

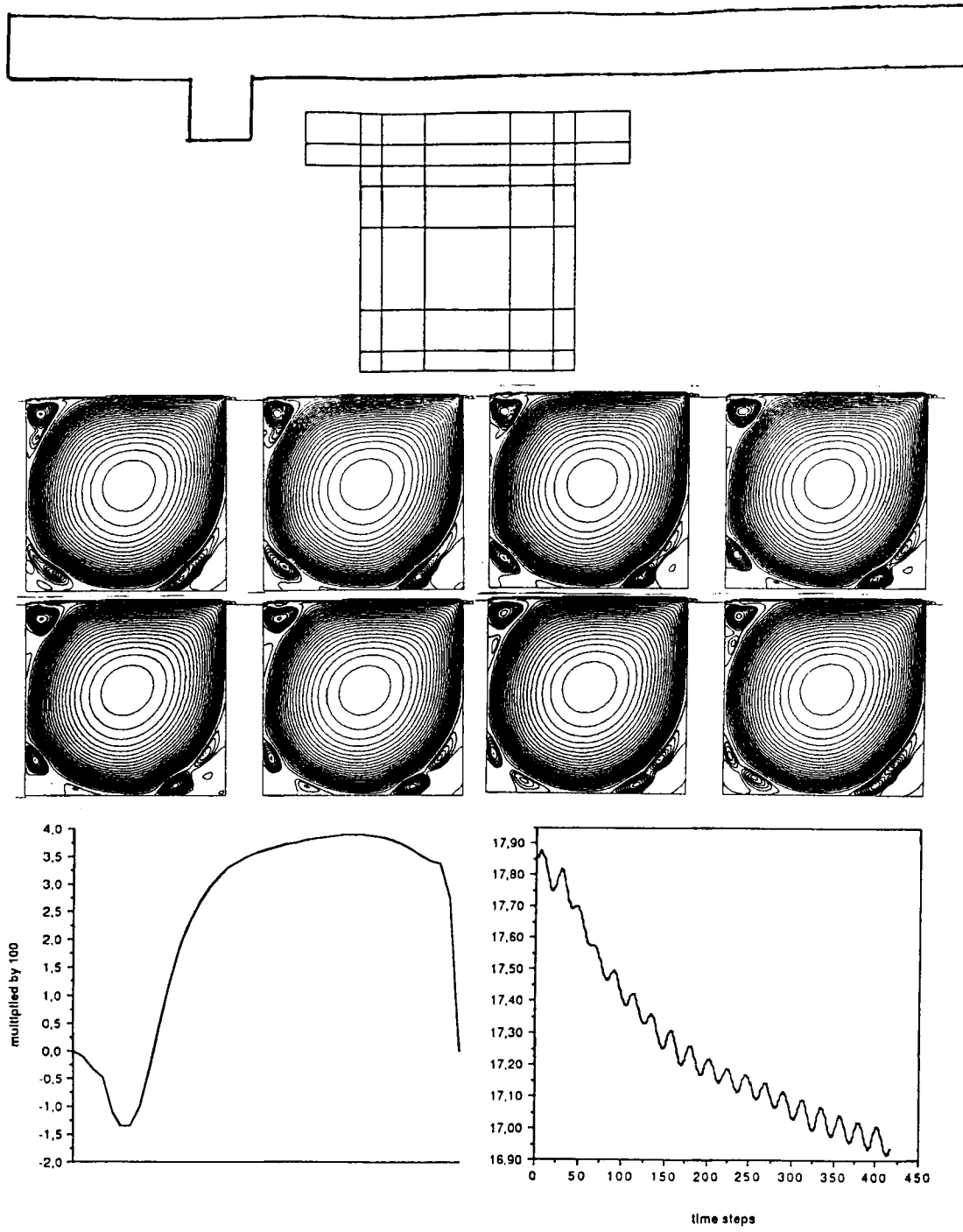

Figure 12. Computations for a 'natural' cavity problem: full domain, restricted computational domain with coarse grid, streamlines for different times, typical velocity profile at $y = 1$, norm of the velocity for subsequent time steps

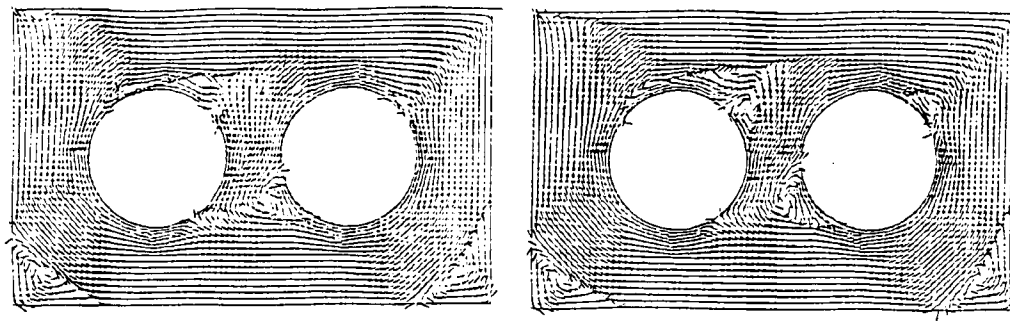

Figure 13. *Driven cavity* with two circular obstacles for $1/\nu = 500$

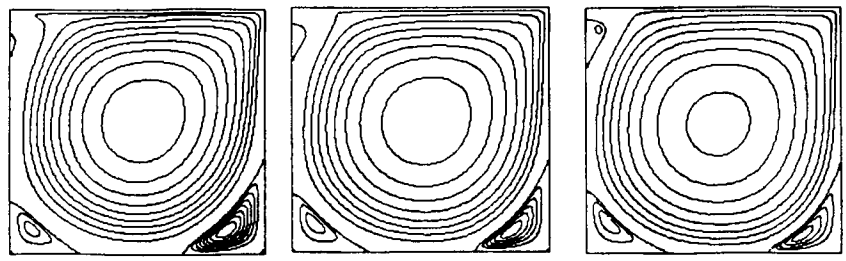

Figure 14. Non-Newtonian flow for $\alpha = 0, 0{\cdot}3, 0{\cdot}5$

Another example shows the applicability of our method to several boundary components. This test is a *driven cavity* computation with two circular obstacles inside (see Figure 13). It is remarkable that the centrally discretized solution (left) has smaller separation zones than the upwind solution (right).

As a final result concerning this type of cavity problem we tested our non-Newtonian fluid model (44) (see also Reference 11 for many more details). In Figure 14 some results with $Re = 2000$ for $\alpha = 0$ (i.e. Newtonian flow), 0·3 and 0·5 are presented.

The next problem is the well-known *backward-facing step* flow, for which many comparisons and results can be found in Reference 20. Our computational domain has height 1, length before the step 3, height of the step $\frac{1}{2}$ and length after the step L. We made many calculations concerning the *reattachment point*[20] for various discretizations of the convective part and various lengths L; all our results are very similar to those proposed in Reference 20 (for many more details see Reference 4). The mesh used (for $L = 20$) is represented by the coarse grid in Figure 15; the actual grid was refined four times, resulting in 24,000 vertices and 70,000 unknowns. The following calculations for $Re = 5000$ and $\Delta t = 0{\cdot}33$ are most interesting: they show a 'quasi-periodically' oscillating flow represented by streamlines and particle-tracing plots at every third time step (Figure 16) and at every ninth time step (Figure 17). For a better visualization we scaled the y-co-ordinate by a factor of two.

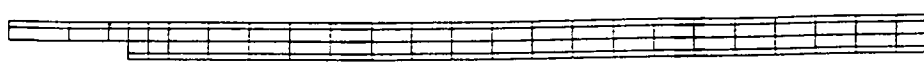

Figure 15. Coarse grid used

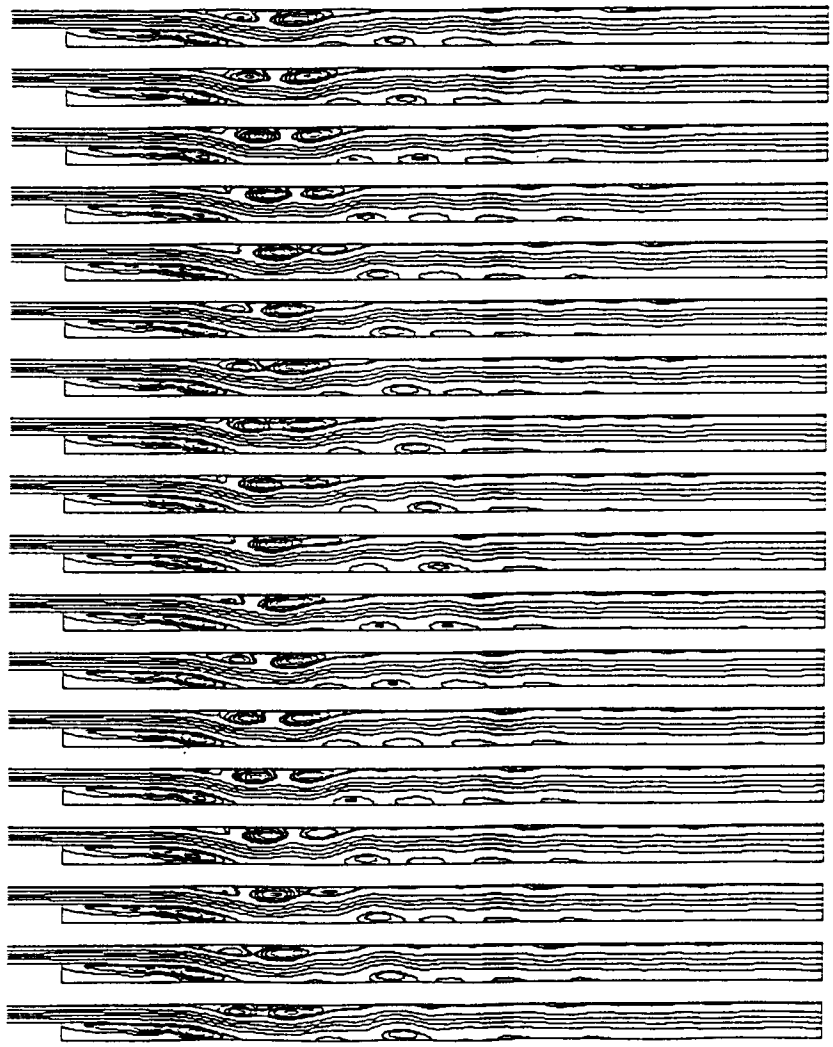

Figure 16. Streamlines for $Re = 5000$

The following calculations concern the flow around an obstacle, here in the form of an inclined ellipse in a channel. A typical computational domain and the corresponding *semi-adaptive* coarse grid are shown in Figure 18. This mesh is refined four times, leading to 27,000 vertices and 80,000 unknowns. The height of the channel is 7, the major axis of the ellipse is 1 and the viscosity is $1/\nu = 500$. First we show some results concerning the difficulties in visualizing non-stationary flows (see also Reference 15). Looking again at Figure 6, the differences between the 'stationary' tools (streamlines, vector plots) and the 'non-stationary' tools (particle tracing) are obvious and justify the costly development of such methods. Also, the question of the 'right' boundary conditions was treated (for more details see Reference 14). Figure 19 shows the same computation with our 'do nothing' outflow condition and constant inflow for two different lengths of the channel. In the next figures we demonstrate some results for the same configuration

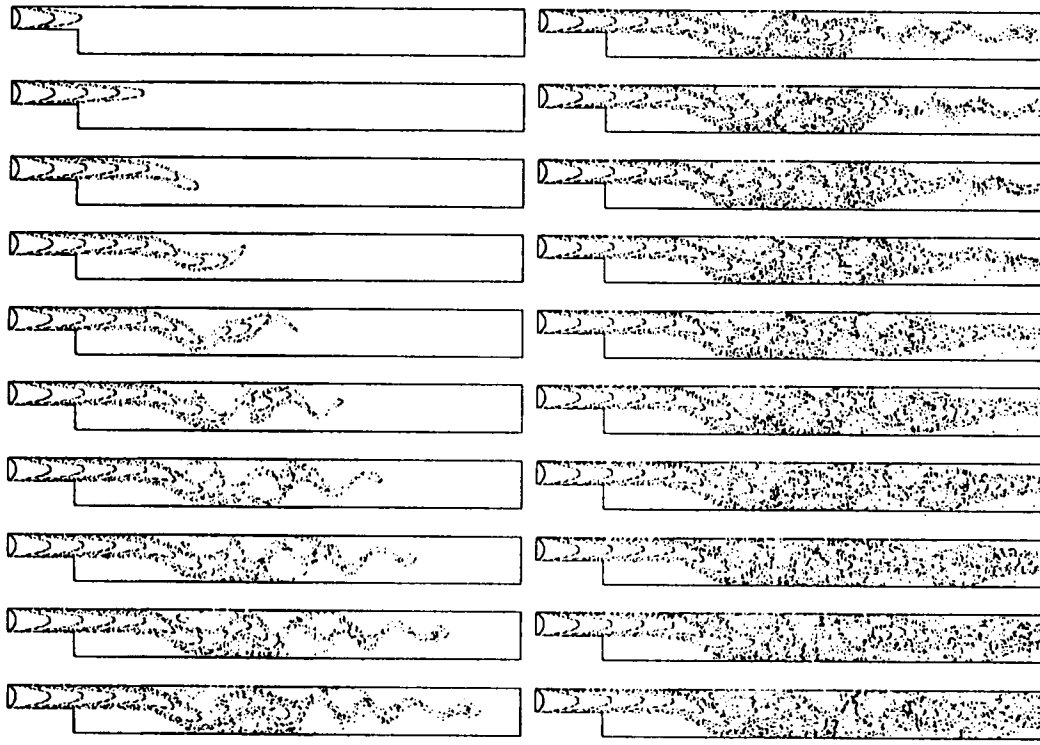

Figure 17. Particle-tracing plots for $Re = 5000$

of all input parameters (Samarskij upwinding, $1/\nu = 500$, $T = 50$, start from rest) but with varying time step sizes ($\Delta t = 0{\cdot}11, 0{\cdot}33, 1{\cdot}0$) and various time-stepping schemes (implicit Euler (IE), Crank–Nicolson (CN), fractional step (FR)). In Figure 20 the *relative* streamlines (after subtracting the corresponding Stokes flow) for $T = 7$, 17 and 50 and $\Delta t = 0{\cdot}33$ are depicted. The implicit Euler scheme results in less accurate solutions, corresponding to its first-order accuracy, while the Crank–Nicolson and fractional step schemes, both of second order, lead to better results. Figure 21 demonstrates that our time step size is fine enough (comparing the FR results for $\Delta t = 1{\cdot}0$ and $0{\cdot}11$) and that for the implicit Euler scheme the time step must be chosen (at

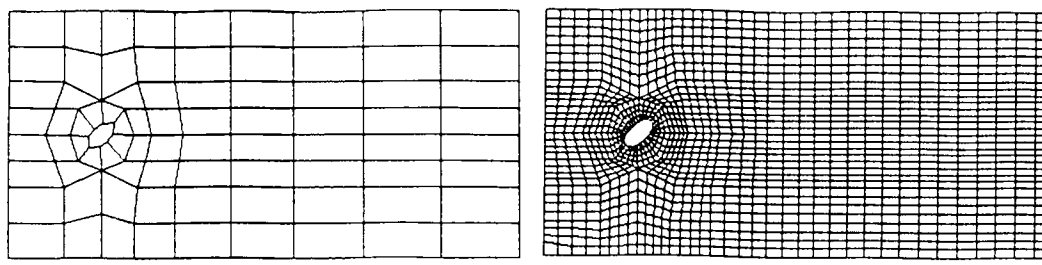

Figure 18. Coarse and refined grid

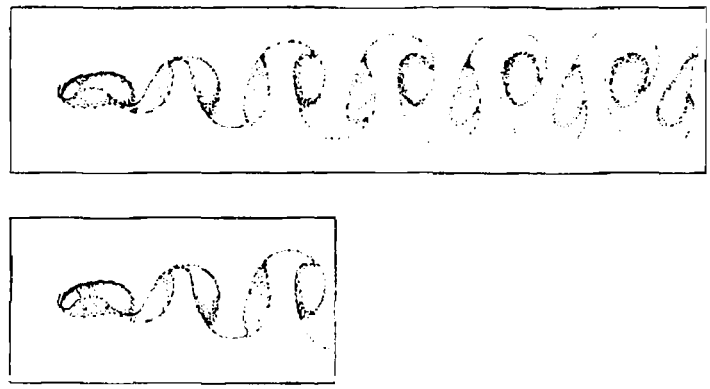

Figure 19. Different lengths of the channel

least) 10 times smaller than that of the fractional step scheme. Additionally, we show in Figure 22 that (in many cases) it is impossible to differentiate between the space error (left column for fractional step with $\Delta t = 0{\cdot}33$ on the coarser mesh) and the time error (right column for implicit Euler on the fine mesh). Both calculations seem to produce (unphysically) stationary results. Finally, in Figure 23 we refined only around the ellipse, with a coarse mesh after the ellipse (left column) and for comparison a very fine mesh only around the ellipse (right column). The result, as expected, is much worse in the second case.

After all our extensive calculations for these 'classical mathematical' test problems (*driven cavity*, *backward-facing step*, flow around an obstacle) we would like to make some observations.

We prefer the upwind discretizations for the convective parts because they lead, as opposed to the theoretical more accurate central schemes, to stabilized solutions where the accuracy can be improved by our 'semi-adaptive' meshes. For the time discretization we favour the fractional

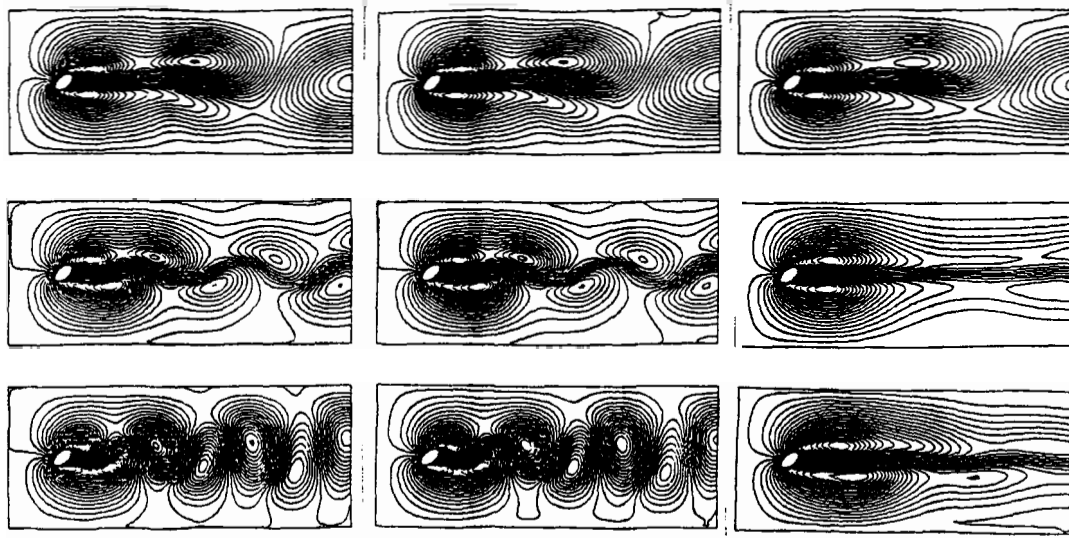

Figure 20. Flow for $\Delta t = 0{\cdot}33$ for FR (left), CN (middle) and IE (right) schemes

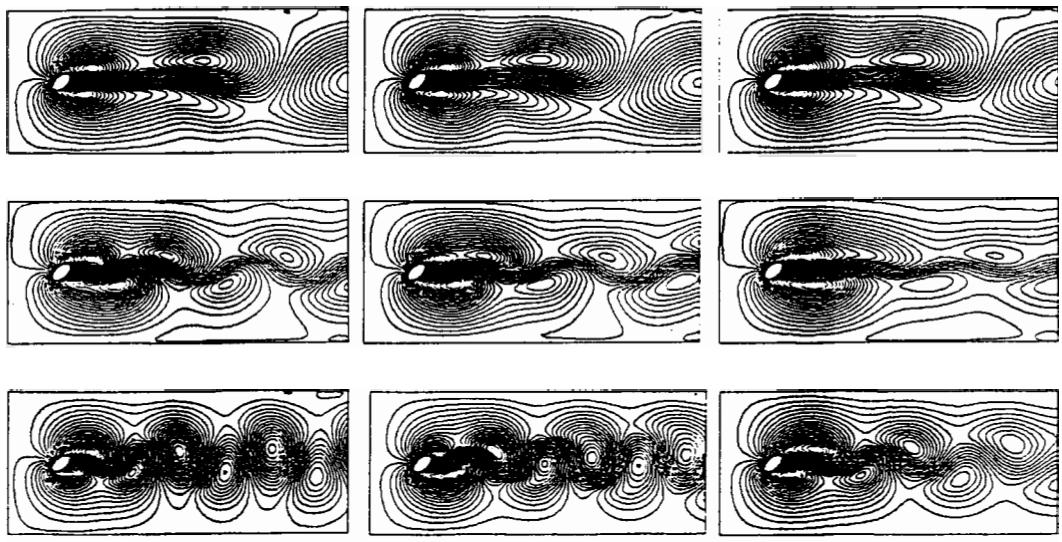

Figure 21. FR–0·11, FR–1·0 and IE–0·11

step ϑ-scheme since it is very robust (like implicit Euler) and accurate (like Crank–Nicolson), with about same amount of arithmetic operations for building up and solving the corresponding stiffness matrices (one has to compare three steps of Crank–Nicolson with the three substeps of the fractional step scheme to get the same error and costs). Using the 'do nothing' boundary condition, we can restrict the computation to smaller computational domains. The proposed fixed point defect correction method is very robust and efficient for the non-linear equations, and together with the outer iteration for the time discretization we also have a good non-

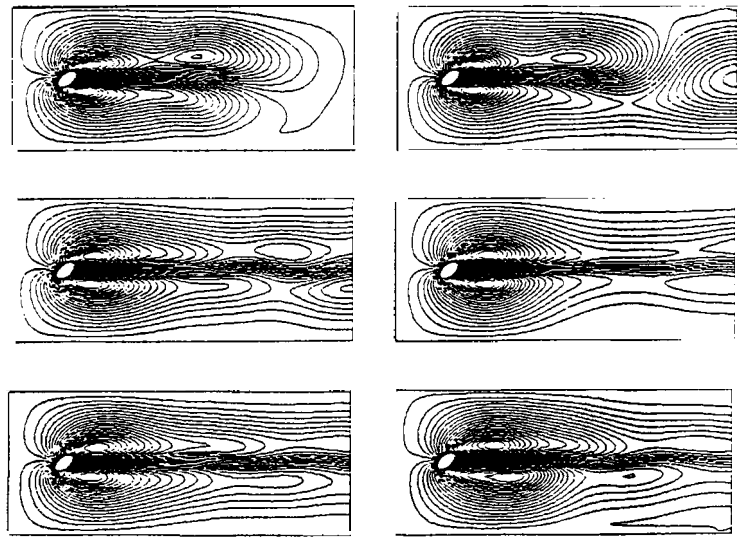

Figure 22. FR 0·33 coarse and IE–0·33–fine

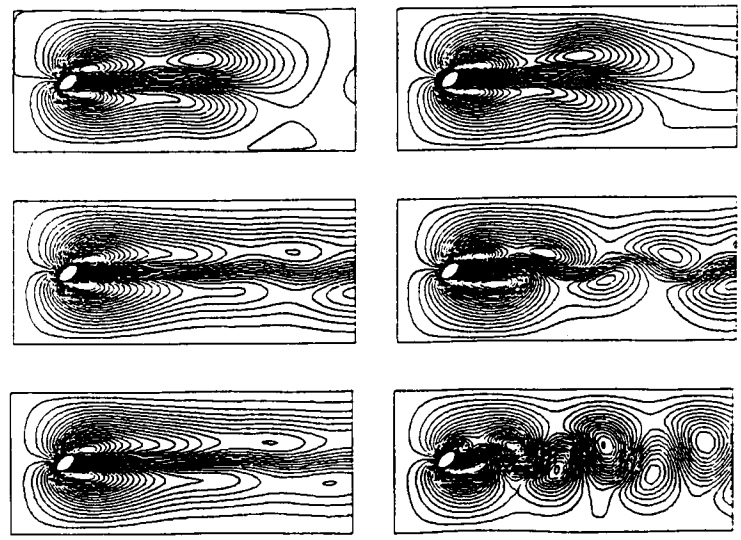

Figure 23. Refinement after the ellipse and refinement around the ellipse

stationary solver. Together with the proposed graphical tools our method seems to fulfil the conditions of an FRA solver. Thus our next step is to apply this implemented code to some more complicated and more interesting problems such as the following.

The first problem (as a demonstration of the Bernoulli principle) is the flow through a special pipe ('venturi pipe'[14]) modelling a device used in sailboats to drain water from the hull. The model used is drawn in Figure 24. We expect some inflow (upper small device) into the boat for low velocity and outflow for higher velocity. As boundary conditions we prescribe a flux at the inlet (left aperture) and the 'do nothing' condition (equivalent to zero mean pressure[14]) at both of the holes. We performed two stationary calculations for viscosity parameters of 10 and 50 and a non-stationary calculation for $1/\nu = 1000$ in which we reached a 'turbulent' flow. In Figures 25 and 26 the corresponding streamlines (snapshots for $1/\nu = 1000$), the velocity profile in the upper pipe and the non-stationary flow (by particle tracing and streamlines) are displayed for subsequent times.

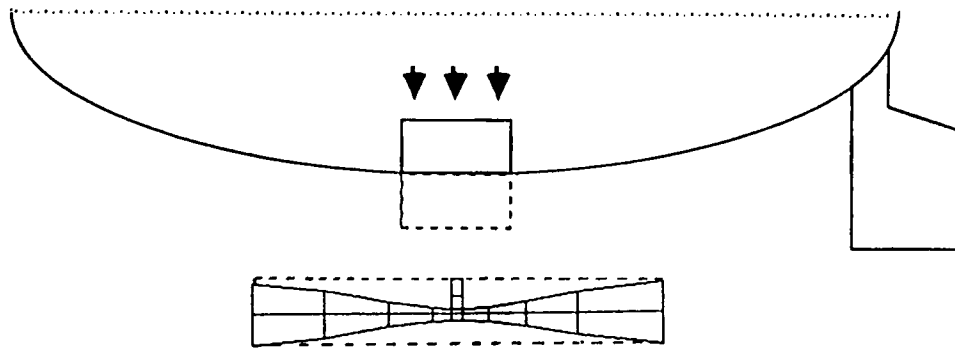

Figure 24. Computational model and coarse grid for 'venturi pipe'

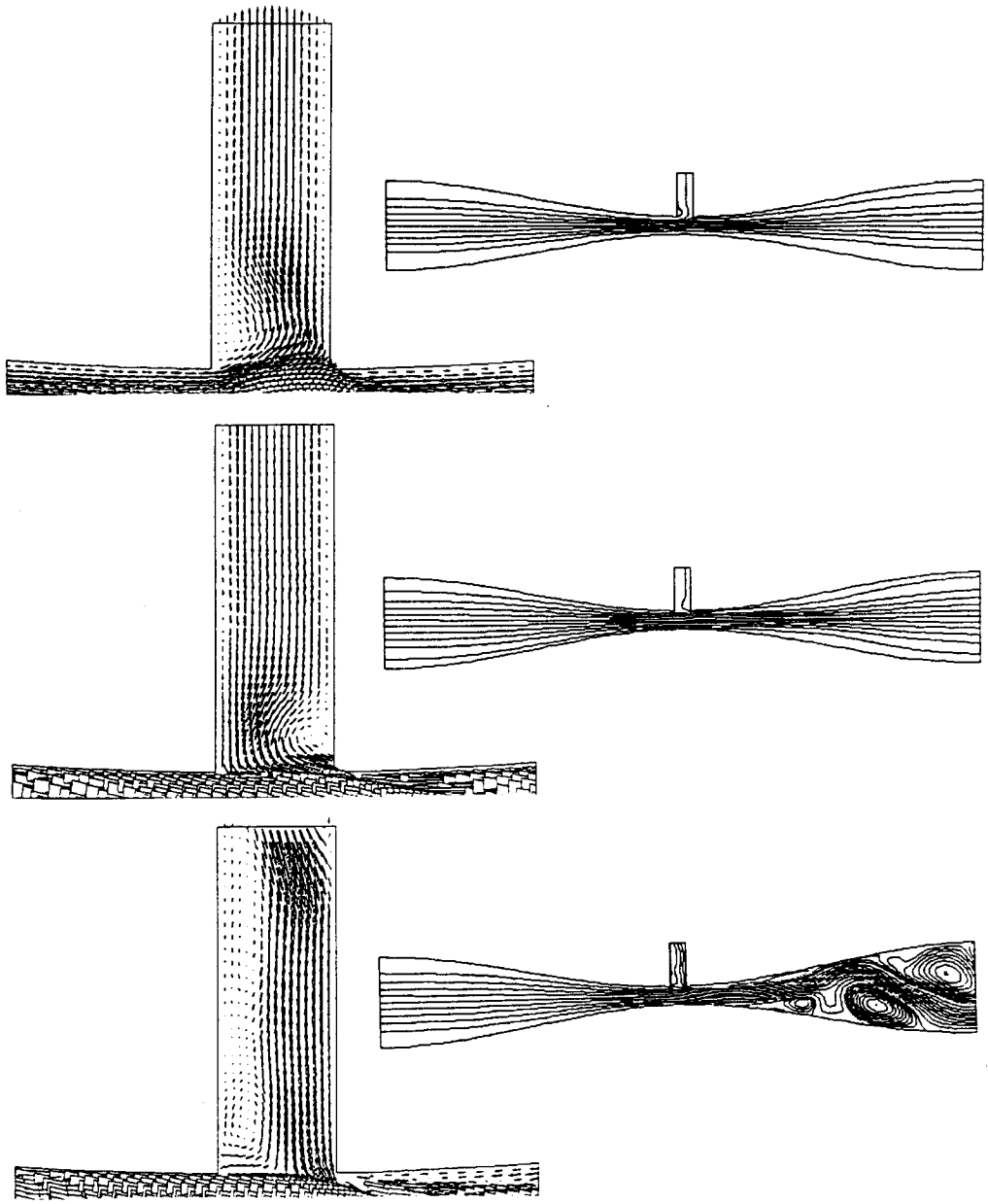

Figure 25. Streamlines and velocity profiles ($1/\nu = 10, 50, 1000$) for the 'venturi pipe'

The next example is the flow through a hole in an 'infinite space'. We restricted the computational domain to the 'circle' pictured in Figure 27, with homogeneous boundary values at the wall (the hole is in the centre) and the 'do nothing' condition on the circle boundary (for more details see Reference 14). The hole width is 0·2 and the radius of the domain is 4·5. In Figure 28 some results (streamlines) for a 'large' viscosity ($1/\nu = 1, 10, 50$) and prescribed flux

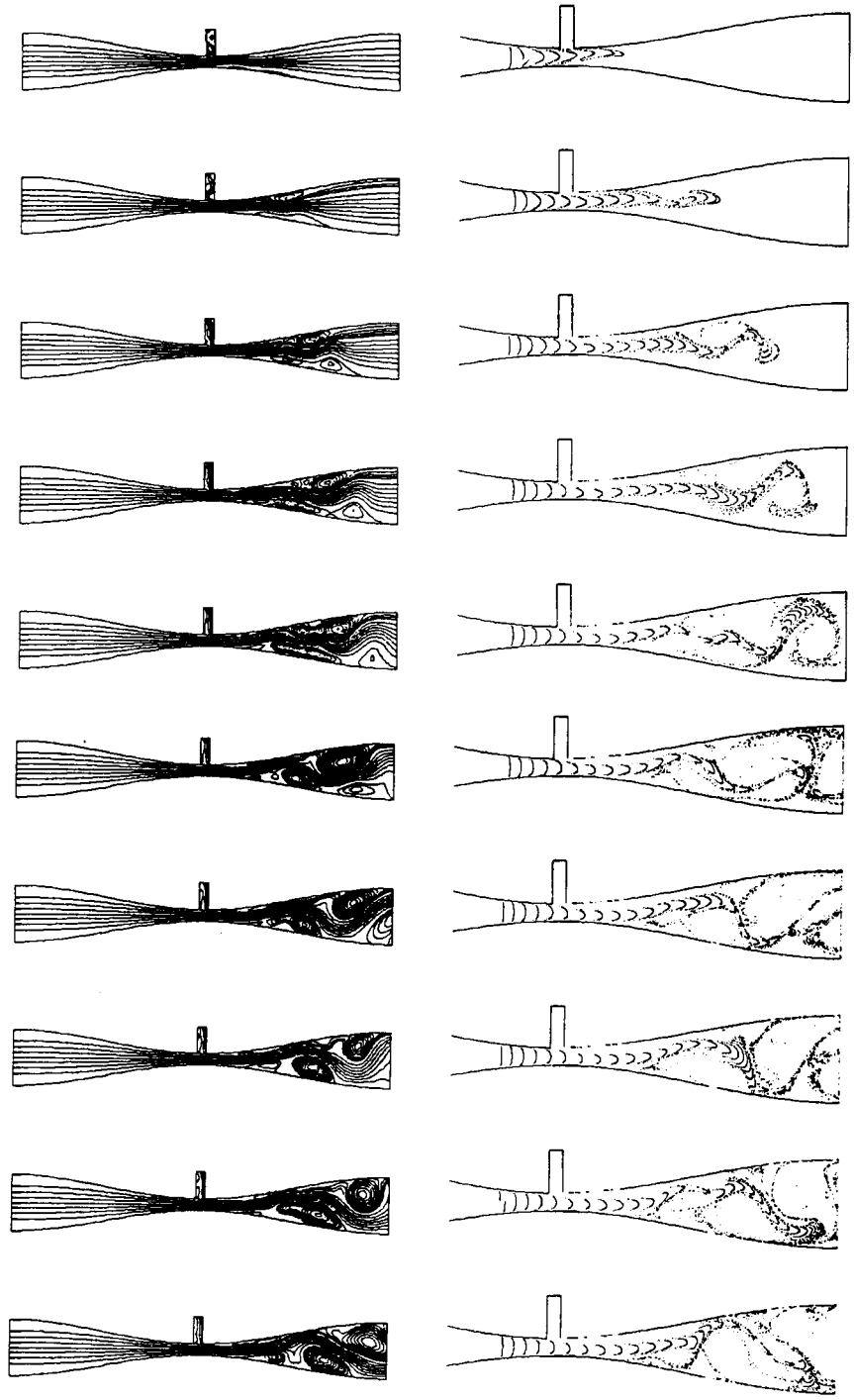

Figure 26. Time-dependent flow (particle tracing and streamlines) for $1/\nu = 1000$

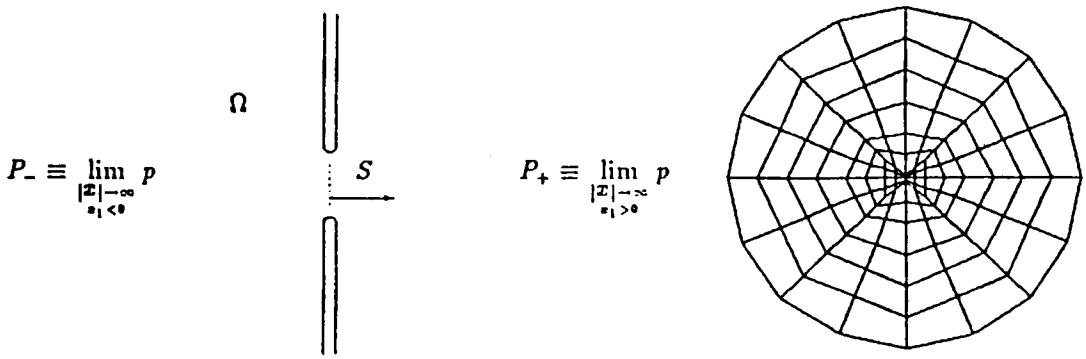

Figure 27. Domain and coarse grid for flow through a hole

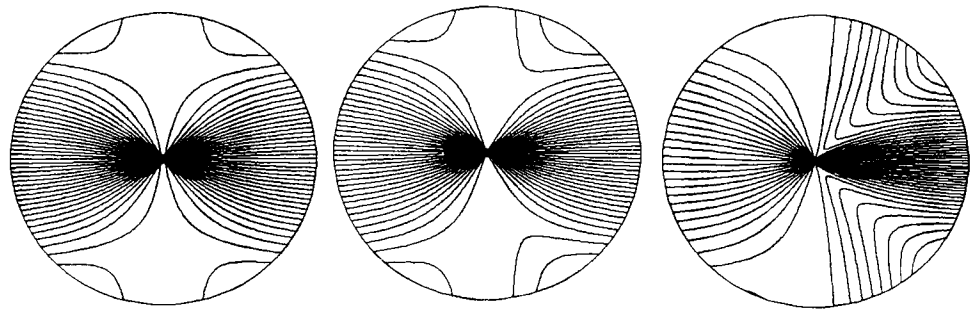

Figure 28. Streamlines for $1/\nu = 1, 10, 50$

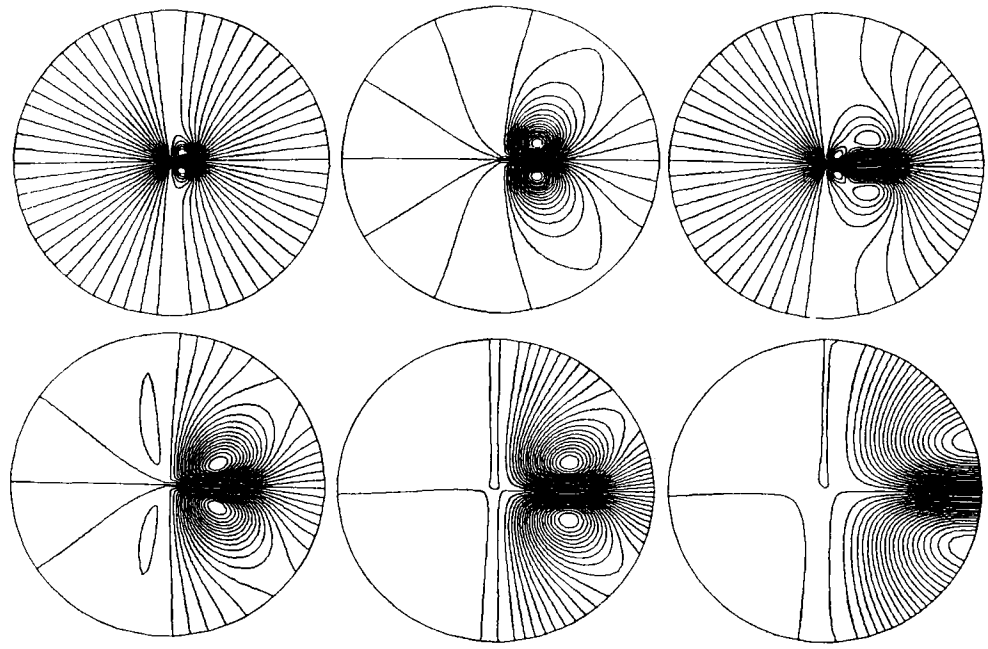

Figure 29. Streamlines for subsequent time steps

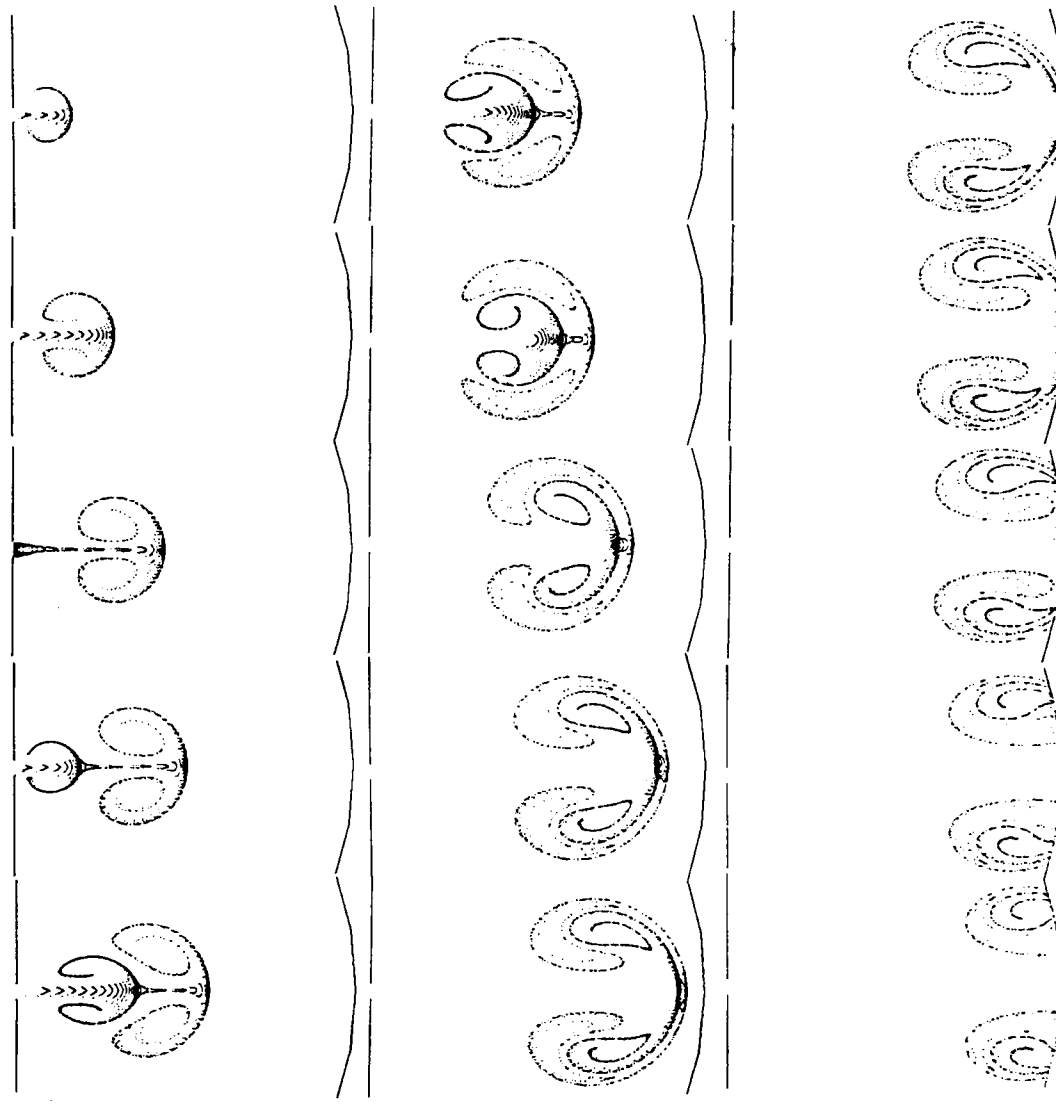

Figure 30. Particle-tracing representation of the flow

$S = 0{\cdot}1$ are plotted. Additionally, we performed some non-stationary calculations ('*puff-puff*') for $1/\nu = 100$ using the pressure drop condition.[14] We prescribed the pressure difference $\Delta p = 1$ for the first 40 time steps, then $\Delta p = 0$ for the next 40, again $\Delta p = 1$ for 40 steps and finally $\Delta p = 0$ until the end (700 time steps with $\Delta t = 0{\cdot}1$). In Figures 29 and 30 sequences of particle-tracing and streamline pictures are depicted. The second 'smoke ring' catches the first one and passes through it nearly, while the first one is accelerated at the same time ('leapfrogging'; compare with the book of Van Dyke[21]). Both rings pass through our artifical boundary, demonstrating the quality of the 'do nothing' boundary condition in this case.

As a final example we calculate the 'airflow' of a jet in a box. The computational domain is

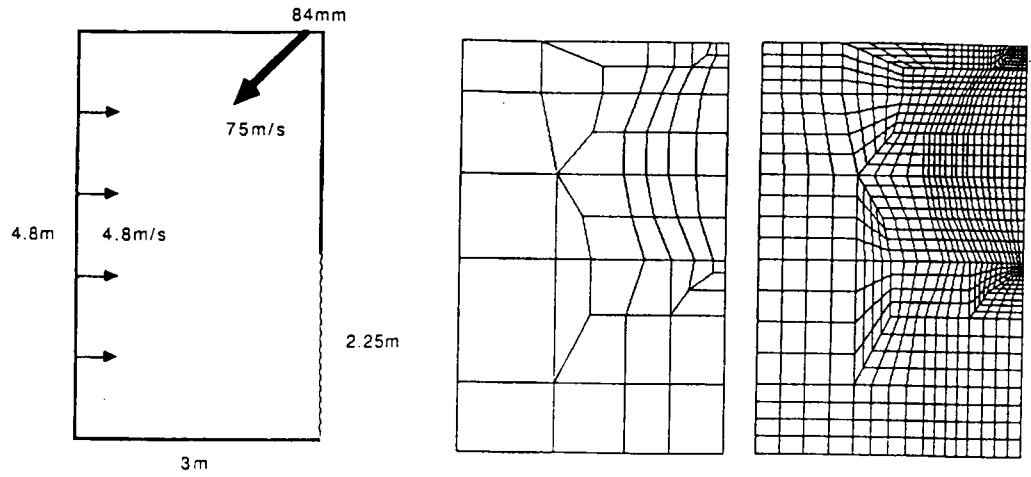

Figure 31. Computational domain, coarse grid and finer grid for the jet flow

presented in Figure 31. The inflow is 4·8 m/s^{-1} at the left edge and 75 m/s^{-1} near the right upper corner (with a width of 84 mm), while the 'do nothing' (outflow) condition is used in the lower part of the right edge. The viscosity is $\nu = 10^{-3}\ m^2\ s^{-1}$. The resultant flow is 'highly' non-stationary and 'turbulent'-looking, with a very complex structure. It is very remarkable that for this high Reynolds number the chosen mesh (the depicted coarse grid was refined five times, resulting in 85,000 grid points, with 32 grid points for the inflow part near the upper right corner) is fine enough to show how the symmetry of the incoming sharp jet is broken far away from influences coming from the boundary. The corresponding streamline pictures are shown in Figure 32 and the particle-tracing plots in Figure 33. Again we see the differences in the visualization techniques.

5. CONCLUSIONS

We have developed a solution method for the non-stationary Navier–Stokes equations in two dimensions belonging to the class of FRA solvers proposed in the introduction. Our main components are discretely divergence-free finite elements of second order, a corresponding multigrid algorithm for the linearized Navier–Stokes equations, an upwind discretization for the convective parts, a fixed point defect correction method for the non-linearity, an accurate and robust time-stepping scheme and appropriate visualization tools on workstations for both stationary and non-stationary flows. We have tested this code for several well-known problems such as driven cavity flow and flow over a step or around an obstacle and for some new, more complicated problems. The next step will be its implementation on new computer architectures such as transputer systems, where first successful attempts have been made by Schieweck.[22] We will also develop a corresponding numerical method with graphical support in three space dimensions. Besides a clever visualization, the most difficult point will be an efficient treatment of the divergence-free finite elements. First attempts have been made by Hecht,[23] but his approach must be improved. Then the corresponding multigrid routines must be implemented efficiently as in 2D, because they consume the most computer time in this code. We hope to have the possibility of testing this code in 2D and 3D on more practical and physically interesting problems.

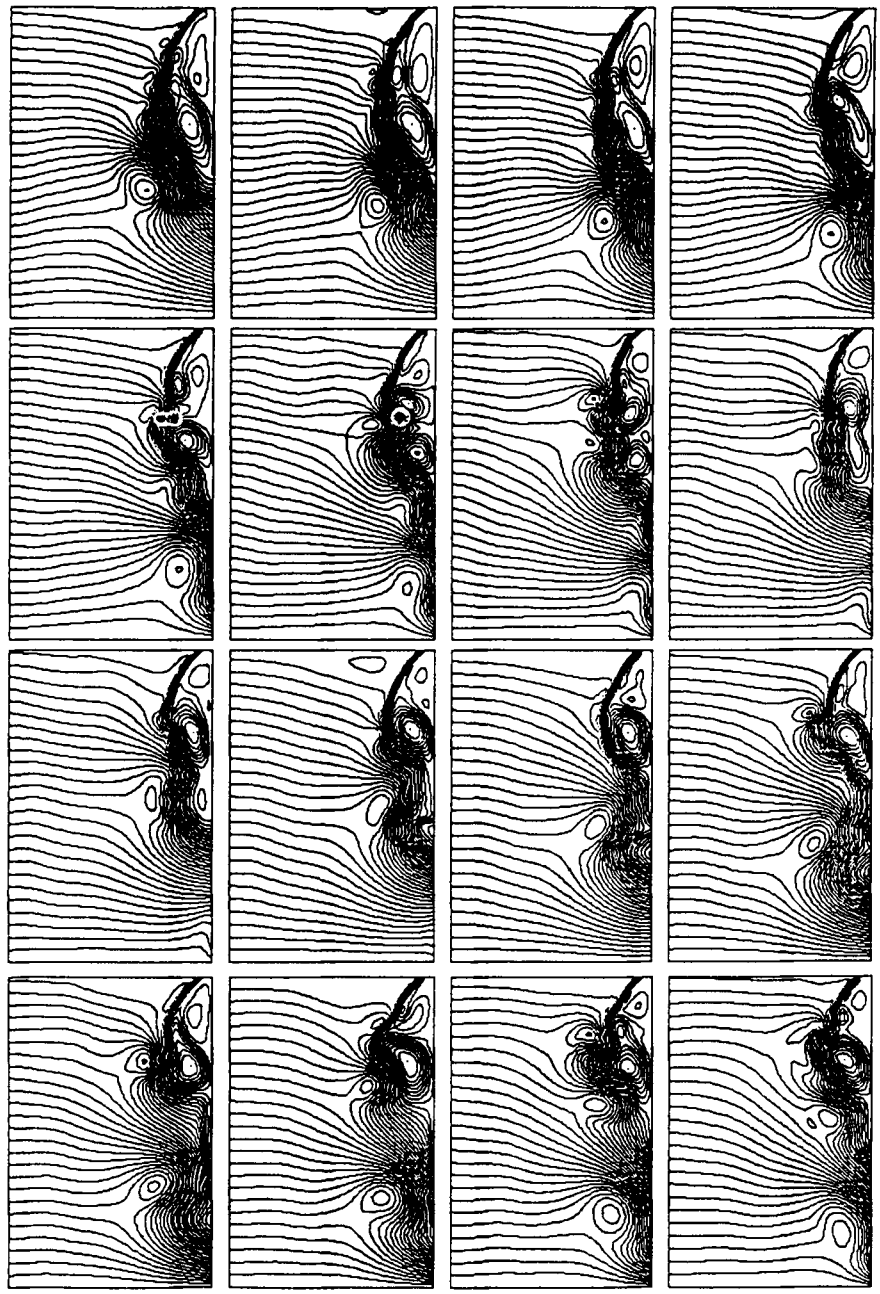

Figure 32. Streamlines for jet flow in a box

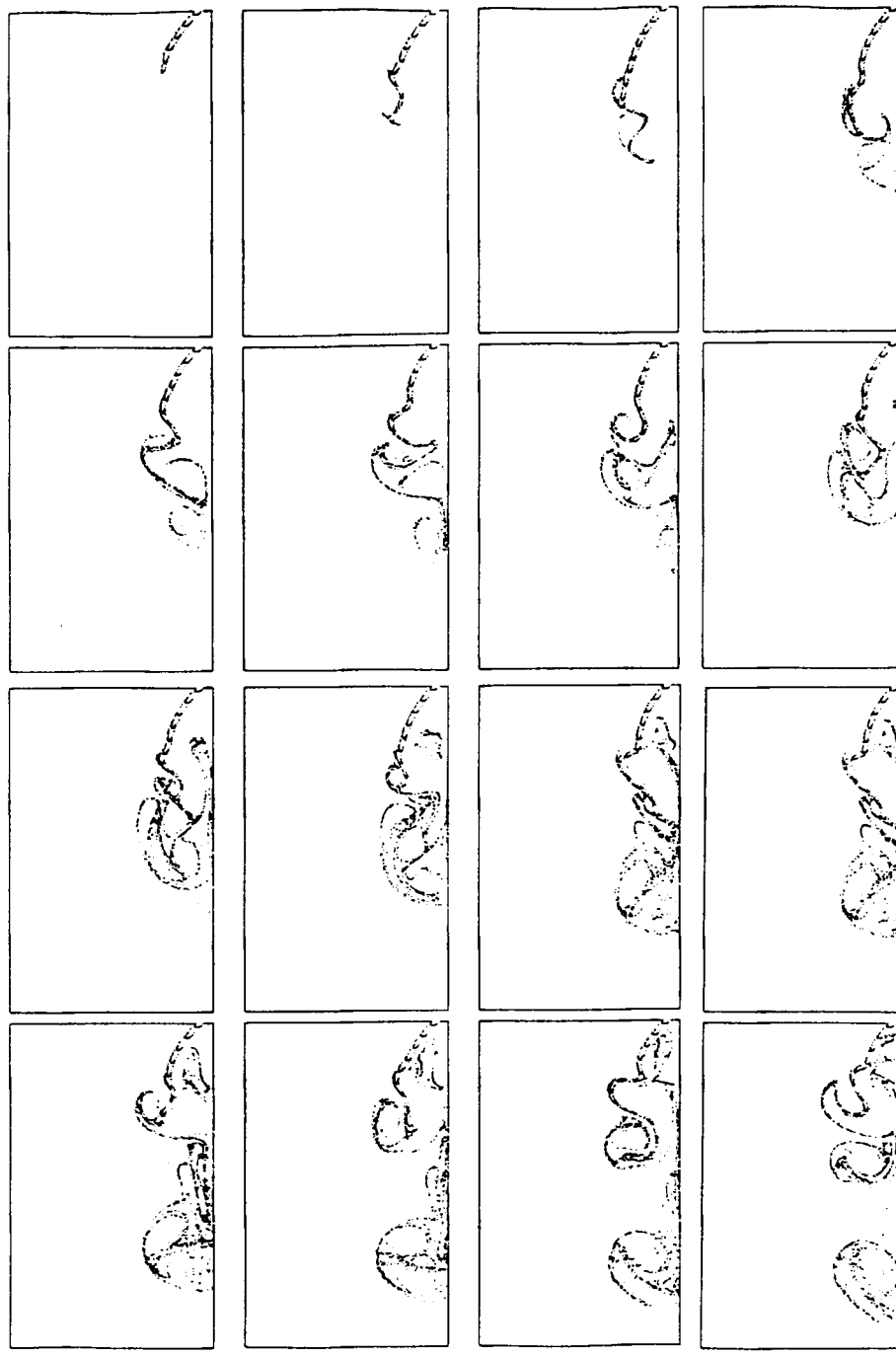

Figure 33. Particle-tracing plots for jet flow in a box

REFERENCES

1. H. Blum, J. Harig, S. Müller and S. Turek, '**FEAT2D**. Finite element analysis tools. User manual. Release 1·3', *Tech. Rep.*, University of Heidelberg, 1992.
2. V. Girault and P. A. Raviart, *Finite Element Methods for Navier–Stokes Equations*, Springer, Berlin, 1986.
3. Ph. G. Ciarlet, *The Finite Element Method for Elliptic Problems*, North-Holland, Amsterdam, 1976.
4. S. Turek, 'Ein robustes und effizientes Mehrgitterverfahren zur Lösung der instationären, inkompressiblen 2-D Navier–Stokes-Gleichungen mit diskret divergenzfreien finiten Elementen', *Thesis*, University of Heidelberg, 1991.
5. R. Rannacher and S. Turek, 'A simple nonconforming quadrilateral Stokes element', *Numer. Methods Partial Differ. Eqns*, **8**, 97–111 (1992).
6. D. Griffiths, 'An approximately divergence-free 9-node velocity element (with variations) for incompressible flows', *Int. j. numer. methods fluids*, **1**, 323–346 (1981).
7. S. Turek, 'Multigrid techniques for a class of discretely divergence-free finite element spaces', in preparation.
8. K. Ohmori and T. Ushijima, 'A technique of upstream type applied to a linear nonconforming finite element approximation of convective diffusion equations', *RAIRO Numer. Anal.*, **18**, 309–332 (1984).
9. L. Tobiska and F. Schieweck, 'A nonconforming finite element method of upstream type applied to the stationary Navier–Stokes equation', *MMAN*, **23**, 627–647 (1989).
10. L. Tobiska, 'Full and weighted upwind finite element methods', in J. W. Schmidt and H. Späth (eds), *Splines in Numerical Analysis, Int. Seminar ISAM 89*, Weissig, 1989, Akademie-Verlag, Berlin, 1989.
11. J. Malek and S. Turek, 'Non-newtonian flow prediction by divergence-free finite elements', *Tech. Rep. 687, SFB 123*, University of Heidelberg, 1992.
12. R. Rannacher, 'Numerical analysis of nonstationary fluid flow (a survey)', *Tech. Rep. 492, SFB 123*, University of Heidelberg, 1988.
13. R. Glowinski and J. Periaux, 'Numerical methods for nonlinear problems in fluid dynamics', *Proc. Int. Seminar on Scientific Supercomputers*, Paris, February 1987, North-Holland, Amsterdam, 198x.
14. J. Heywood, R. Rannacher and S. Turek, 'Artifical boundaries and flux and pressure conditions for the incompressible Navier–Stokes equations', *Tech. Rep. 681, SFB 123*, University of Heidelberg, 1992.
15. S. Turek, 'Visualization tools for the nonstationary Navier–Stokes equations', *Tech. Rep. 680, SFB 123*, University of Heidelberg, 1992.
16. U. Ghia, K. N. Ghia and T. C. Shin, 'High-*Re* solutions for incompressible flows using the Navier–Stokes equations and a multigrid method', *J. Comput. Phys.*, **48**, 387–411 (1982).
17. R. Schreiber and H. B. Keller, 'Driven cavity flows by efficient numerical techniques', *J. Comput. Phys.*, **49**, 310–333 (1983).
18. S. P. Vanka, 'Implicit multigrid solutions of Navier–Stokes equations in primitive variables', *J. Comput. Phys.*, **65**, 138–158 (1985).
19. L. B. Zhang, 'A second-order upwinding finite difference scheme for the steady Navier–Stokes equations in primitive variables in a driven cavity with a multigrid solver', *MMAN*, **24**, 133–150 (1990).
20. K. Morgan, J. Periaux and F. Thomasset, *Notes on Numerical Fluid Mechanics*, Vol. 9, *Analysis of Laminar Flow over a Backward Facing Step*, 1984.
21. M. Van Dyke, *An Album of Fluid Motion*, Parabolic, Stanford, CA, 1982.
22. F. Schieweck, 'A parallel multigrid algorithm for solving the Navier–Stokes equations on a transputer system', *Tech. Rep. 699, SFB 123*, University of Heidelberg, 1992.
23. F. Hecht, 'Construction d'une base de fonctions P_1 non conformes á divergence nulle dans R^3', *RAIRO Numer. Anal.*, **15**, 119–150 (1981).
24. L. Tobiska, 'Numerical methods in singularly perturbed problems', in H. G. Roos, A. Felgenhauer and L. Angermann (eds), *Int. Seminar in Applied Mathematics, Int. Seminar ISAM 91*, Dresden, 1991.