LANGLEY GRANT

IN-34-CR

272845

58P.

Final Report, NASA Contract No. NAG-1-812

Study of Gortler Vortices by Compact Schemes

Monitor: Thomas B. Gatski

NASA-Langley Research Center

Hampton, VA 23665

Part 1:

Compact Scheme for Systems of Equations Applied to Fundamental Problems of Mechanics of Continua

by Jerzy Z. Klimkowski

Department of Mechanical Engieneering

North Carolina A & T State University

Greensboro, NC 27411

tel: (919) 334-7620

Abstract. Compact scheme formulation was used in the treatment of boundary conditions for a system of coupled diffusion and Poisson equations. Models and practical solutions of specific engineering problems arising in solid mechanics, chemical engineering, heat transfer and fluid mechanics are described and analysed for efficiency and accuracy. This report is limited only to two-dimensional cases, and ~~is~~ intended to present a new method of numerical treatment of boundary conditions common in the fundamental problems of mechanics of continua.

(NASA-CR-186494) COMPACT SCHEME FOR SYSTEMS OF EQUATIONS APPLIED TO FUNDAMENTAL PROBLEMS OF MECHANICS OF CONTINUA Final Report (North Carolina Agricultural and Technical State Univ.) 58 p CSCL 20D

N90-21292

Unclas

G3/34 0272845

Chapter 1

Introduction

Among various types of domain decomposition methods, the work of *M.E. Rose et al.* on initial-boundary value problems by finite volume methods is gaining well-deserved attention. This approach offers a fresh look at century-old problems arising in the mechanics of continua. As can be seen from the literature review at the end of this report, the growing body of that effort is not yet available to wider readership. Cited work is primarily theoretical, whereas here some two-dimensional problems arising in solid and fluid dynamics are solved in conjuncture with *Rose's* formulation and the presentation of the method itself is reduced to the necessary minimum.

All the presented applications involve the numerical treatment of harmonic and bi-harmonic operators. Therefore, some simple facts about the Poisson equation have to be reviewed first.

For a plethora of problems, the following equation is created:

$$\oint \overline{grad\, w} * \overline{dS} = F \tag{1.1}$$

It is customary to apply divergence the theorem and rewrite Eq. 1.1 in the form:

$$\int \nabla^2 w \; d\forall = F \tag{1.2}$$

For two-dimensional rectangular grids this procedure leads to a standard five-point stencil representation:

$$\kappa_x(w_{i-1,j} - 2w_{i,j} + w_{i+1,j}) + \kappa_y(w_{i,j-1} - 2w_{i,j} + w_{i,j+1}) = f_{i,j} \tag{1.3}$$

where $\kappa_x = \frac{1}{\Delta x * \Delta x}$ and $\kappa_y = \frac{1}{\Delta y * \Delta y}$.

The right-hand side term $f_{i,j}$ is usually interpretted as a source (per unit volume or area) and is given at the center of each cell.

However, this entire procedure can be easily modified for a new, specific definition of the *normal derivative.*

For reasons to be explained shortly, consider *another* function $\phi = \phi(x, y)$. It is assumed that the computational domain $\mathcal{D}$ is divided into $I * J$ non-overlapping, rectangular cells and the symbol $\phi_{i,j}$ refers to the value of the function ϕ at the center of (ij) -th cell. Whenever this argument of the function is displaced by half-width of cell 'left' ('right') the same function will be referred to as $u_{i-,j}$ $(u_{i+,j})$. Similar notation is used when 'upward' or 'downward' displacement is considered. A simple diagram for the above notation is presented in $Fig.1.$.

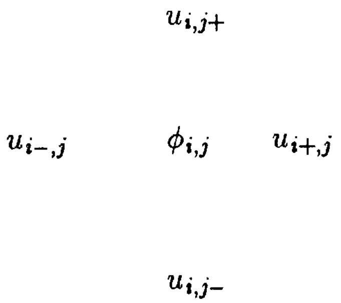


Fig.1. $\mathcal{D}_{ij}$ cell.

Half-step normal derivatives are given by:

$$v^+_{x\,ij} = \frac{2}{\Delta x}(u_{i+,j} - \phi_{ij}) \tag{1.4}$$

$$v^-_{x\,ij} = \frac{2}{\Delta x}(\phi_{ij} - u_{i-,j}) \tag{1.5}$$

$$v^{+}_{y\,ij} = \frac{2}{\Delta y}(u_{i,j+} - \phi_{ij}) \tag{1.6}$$

$$v^{-}_{y\,ij} = \frac{2}{\Delta y}(\phi_{ij} - u_{i,j-}) \tag{1.7}$$

It is assumed that cells $\mathcal{D}_{ij}$ form a non-overlapping cover of the entire computational domain $\mathcal{D}$ and it will be hereon required that functions u and derivatives v be continuous across centers of faces of neighboring cells. It was shown in [1] that values ϕ_{ij} play only auxiliary role, whereas the requirement of the continuity of normal derivatives furnishes additional equations necessary to complete the formulation. This requirement can be written in the form:

$$v^{+}_{x\,ij} = v^{-}_{x\,i+1,j} \tag{1.8}$$

$$v^{+}_{y\,ij} = v^{-}_{y\,i,j+1} \tag{1.9}$$

Substituting u for w , and yet unspecified, function -W for F, the entire analysis is repeated for half-step derivatives 1.4-1.7, for each cell $\mathcal{D}_{ij}$. This time however, direct summation replaces the use of the divergence theorem. As a result the following system of equations is obtained:

$$\gamma_x(u_{i-,j} - 2\phi_{ij} + u_{i+,j}) + \gamma_y(u_{i,j-} - 2\phi_{ij} + u_{i,j+}) + w_{ij} = 0, \tag{1.10}$$

where: $\gamma_x = 2\kappa_x$ and $\gamma_y = 2\kappa_y$.

This should be complemented by equations stemming from the continuity of normal derivatives.

The scheme introduced above is an example of a *new, weak compact scheme,* in terminology introduced by *Rose* , [1]. 'New' refers to an improved modeling of normal derivatives; the half-step formulas that were absent from [4], whereas 'weak' refers to the fact that only derivatives normal to faces of cells are involved. No comprehensive study of the practical applications was presented to date. Some pilot results are given in [6]. Numerous problems arise when applying compact scheme in higher dimensions and/or systems of equations. Among them, the difficulty of matching this 'stingy' formulation with an equally efficient numerical procedure. In the context of standard engineering problems of continuum mechanics, three different types of solutions were attempted. The first one, based on Paceman-Rachford alternative direction implicit (ADI) scheme [2], revealed serious stability problem. Second attempt involved the application of Kaczmarz algorithm [3] for the entire system of equations (including the ADI) and was hampered by slow covergence and low accuracy, although several potential advantages became apparent. Finally, a compromized approach based on the Gauss algorithm was tested and proved to be a viable alternative.

Chapter 2

Coupled Systems of Equations

For the square domain $\mathcal{D}$, we will consider a coupled system of time-dependent equations given by:

$$\frac{\partial}{\partial t}\Psi = \nabla^2\Psi - f \tag{2.1}$$

$$\nabla^2\Phi = -\Psi. \tag{2.2}$$

We will impose the boundary conditions in the form:

$$\Phi(x,0) = \Phi(x,1) = \Phi(0,y) = \Phi(1,y) = 0 \tag{2.3}$$

and

$$\frac{\partial}{\partial y}\Phi(x,0) = \frac{\partial}{\partial y}\Phi(x,1) = \frac{\partial}{\partial x}\Phi(0,y) = \frac{\partial}{\partial x}\Phi(1,y) = 0 \tag{2.4}$$

Initial conditions are given only for the Φ-function by:

$$\Phi(0,x,y) = \Phi_0(x,y) \tag{2.5}$$

It is important here, that the boundary conditions are imposed only on the function Φ, therefore the method of solution considered further in this report, has to take into the account the transfer of these boundary conditions. Considerations based on Green's theorem lead to a simple resolution of this problem.

Combining Eq's 2.1 and 2.2 with

$$\int(\Phi\nabla^2\Psi - \Psi\nabla^2\Phi)d\mathcal{V} \;=\; \oint(\Phi\nabla\Psi - \Psi\nabla\Phi) * \mathbf{n}dS \tag{2.6}$$

the following relation is obtained:

$$\begin{aligned}\int\Phi\frac{\partial}{\partial t}\Psi d\mathcal{V} + \int(\Psi^2 + \Phi f)d\mathcal{V} = \\ \oint(\Phi\nabla\Psi - \Psi\nabla\Phi) * \mathbf{n}dS\end{aligned} \tag{2.7}$$

Furthermore, introducing the notation:

$\mathbf{v} = \nabla\Psi$

$\mathbf{u} = \nabla\Phi$

$v_n = \nabla\Psi * \mathbf{n}$ and

$u_n = \nabla\Phi * \mathbf{n}$, respectively, and applying another form of the Green's theorem:

$$\int\nabla\Phi * \nabla\Gamma d\mathcal{V} + \int\Phi\nabla^2\Gamma d\mathcal{V} \;=\; \oint\Phi\nabla\Gamma * \mathbf{n}dS \tag{2.8}$$

one arrives at:

$$\begin{aligned}\frac{1}{2}\frac{d}{dt}\int u^2 d\mathcal{V} + \int(\Psi^2 + \Phi f)d\mathcal{V} = \\ \oint[\Phi(v_n + \frac{\partial}{\partial t}u_n) - u_n\Psi]dS.\end{aligned} \tag{2.9}$$

This relation, Eq. 2.9 above, constitutes the energy principle for the entire system. For the type of boundary conditions expressed by Eq. 2.3-2.4, the uniqueness of a solution for a steady case can be easily proven.

Consider the forcing function, f, to be equal to zero everywhere in the computational domain, $\mathcal{D}$. Then the function Φ becomes a solution to the Laplace equation. Moreover, the boundary conditions are uniformly equal zero and, as a result $\Phi = 0$, everywhere in $\mathcal{D}$. For the time-dependent problem, the analysis of Eq. 2.9 leads to the conclusion that the system has a contracting property, that is, solution converges asymptoticaly to zero, everywhere in $\mathcal{D}$.

It is worth mentioning that the boundary conditions used here are related to the dissipative boundary conditions in the sense of Kreiss [8]. In fact, the left-hand side of Eq 2.9 needs to be non-positive for the assertion about the uniqueness of the solution to hold, just like in Kreiss conditions, except that his statements were applied to a single function relations. This observation will be very useful for some practical applications considered further in this work.

Discrete Energy Principle

In the context of compact schemes, the discrete equivalent of the energy principle was first proven for the diffusion equation by *Rose* [9] and [10]. Derivation, below, constitutes its extensions necessary for cases including coupled systems of diffusion and Poisson equations.

The notation introduced for gradients will be used in rewriting the system of Eq's 2.1-2.2. in the form:

$$\begin{aligned} \nabla\Phi &= \mathbf{u} && (2.10)\\ \nabla\Psi &= \mathbf{v} && (2.11)\\ \nabla * \mathbf{u} + \Psi &= 0 && (2.12)\\ \frac{\partial}{\partial t}\Psi &= \nabla * \mathbf{v} - f. && (2.13) \end{aligned}$$

It is easier to follow the derivation when $\mathbf{u}$ and $\mathbf{v}$ are interpreted through their respective components, namely, $\mathbf{u} = (p, q)$ and $\mathbf{v} = (r, s)$.
It will be assumed that the Cartesian grid is uniform with steps Δx, and Δy in x-

and y-directions respectively. In the same way, Δt is a step in time.

Following [1], half-step formulas for derivatives are introduced, capitalized Greek letters refer to values of functions at centers of elements:

$$
\begin{aligned}
p_{ij}^{+} &= \frac{2}{\Delta x}(\phi_{i+\frac{1}{2},j} - \Phi_{i,j}) && (2.14)\\
p_{ij}^{-} &= \frac{2}{\Delta x}(\Phi_{i,j} - \phi_{i-\frac{1}{2},j}) && (2.15)\\
q_{ij}^{+} &= \frac{2}{\Delta y}(\phi_{i,j+\frac{1}{2}} - \Phi_{i,j}) && (2.16)\\
q_{ij}^{-} &= \frac{2}{\Delta x}(\Phi_{i,j} - \phi_{i,j-\frac{1}{2}}) && (2.17)\\
r_{ij}^{+} &= \frac{2}{\Delta x}(\psi_{i+\frac{1}{2},j} - \Psi_{i,j}) && (2.18)\\
r_{ij}^{-} &= \frac{2}{\Delta x}(\Psi_{i,j} - \psi_{i-\frac{1}{2},j}) && (2.19)\\
s_{ij}^{+} &= \frac{2}{\Delta y}(\psi_{i,j+\frac{1}{2}} - \Psi_{i,j}) && (2.20)\\
s_{ij}^{-} &= \frac{2}{\Delta x}(\Psi_{i,j} - \psi_{i,j-\frac{1}{2}}). && (2.21)
\end{aligned}
$$

Standard differencing and averaging operators (in time and in space) will be used extensively:

$$
\begin{aligned}
\mu_x\gamma &= \frac{1}{2}(\gamma_{i+\frac{1}{2}} + \gamma_{i-\frac{1}{2}}) && (2.22)\\
\delta_x\gamma &= (\gamma_{i+\frac{1}{2}} - \gamma_{i-\frac{1}{2}}). && (2.23)
\end{aligned}
$$

Similarily, for y-direction and time operations. In order to make the notation more transparent, "tilde" (like in $\tilde{\gamma}$) and "t" subscript (like in γ_t) will be used for time averaging and time differencing, alternatively.

Applying the μ and δ operators to the half-step formulas, Eq 2.14-2.21, as it was done in [1], important relations for finite difference operators are obtained:

$$\Delta x\mu_x p = \delta_x\phi \qquad (2.24)$$

$$\begin{aligned}
\Delta y \mu_y q &= \delta_y \phi & (2.25)\\
\Delta x \mu_x r &= \delta_x \psi & (2.26)\\
\Delta y \mu_y s &= \delta_y \psi & (2.27)\\
\Phi &= \mu_x \phi - \frac{\Delta x}{4} \delta_x p & (2.28)\\
\Phi &= \mu_y \phi - \frac{\Delta y}{4} \delta_y q & (2.29)\\
\Psi &= \mu_x \psi - \frac{\Delta x}{4} \delta_x r & (2.30)\\
\Psi &= \mu_y \psi - \frac{\Delta y}{4} \delta_y s. & (2.31)
\end{aligned}$$

Indices were ignored in the above formulas. The very nature of differencing and averaging, presented here, produces additional identities, regardless which of the variables, x, y or t is involved:

$$\begin{aligned}
\delta(\gamma\rho) &= \mu(\gamma)\delta(\rho) + \delta(\gamma)\mu(\rho) & (2.32)\\
\mu(\gamma\rho) &= \mu(\gamma)\mu(\rho) + \frac{1}{4}\delta(\gamma)\delta(\rho). & (2.33)
\end{aligned}$$

The first of the identities, Eq. 2.32, constitutes the fulcrum of the proofs contained in [10], because of the simple fact that $\frac{1}{2}\delta_t(\gamma^2) = \mu_t(\gamma)\delta_t(\gamma)$. This was followed by a suitable defintions for both terms on the right-hand side.
As it was already mentioned before, the continuity of normal derivatives will be required for any two adjacent cells, e.g. $p^{+}_{i,j} = p^{-}_{i+1,j}$. This fact is essential in the formulation of boundary conditions corresponding to Eq. 2.4, but it also manifests itself in rather non-intuitive spatial relations, specific to the compact scheme. We will re-interpret the model system, Eqs. 2.1 and 2.2, using the discretization already introduced.

Equations:

$$\begin{aligned}
\frac{1}{\Delta x}\delta_x p_{ij} + \frac{1}{\Delta y}\delta_y q_{ij} &= -\Psi_{ij} & (2.34)\\
\frac{1}{\Delta t}\delta_t \Psi_{ij} &= \mu_t(\frac{1}{\Delta x}\delta_x r_{ij} + \frac{1}{\Delta y}\delta_y s_{ij}) - \mu_t f_{ij} & (2.35)
\end{aligned}$$

are consistent with the system 2.10-2.13. In fact, the introduction of Eqs. 2.34 and 2.35 defines the way in which spatial and time relations will be handled. Moreover, these equations, Eq's 2.24-2.31, assumptions about the continuity of normal derivatives and relations 2.32-2.33 are sufficient to prove the discrete energy principle. Only a sketch of this proof will be presented here, because necessary algebraic manipulations are tedious but fairly elementary.

Consider the expression " $\frac{1}{\Delta t}\mu_t\Phi\delta_t\Psi$ ". Using the identities 2.24-2.31 in Eq. 2.35 it can be proven that:

$$\frac{1}{\Delta t}\mu_t\Phi\delta_t\Psi = \frac{1}{\Delta x}\delta_x(\tilde{\phi}\tilde{r}) + \frac{1}{\Delta y}\delta_y(\tilde{\phi}\tilde{s}) - \mu_x(\tilde{p}\tilde{r}) - \mu_y(\tilde{q}\tilde{s}) - \tilde{\Phi}\tilde{f}. \tag{2.36}$$

There is, however, an additional relation obtained through the time-differencing of Eq. 2.34. Applying similar algebraic manipulations to this relation, one obtains:

$$\frac{1}{\Delta t}\mu_t\Phi\delta_t\Psi = -\frac{1}{\Delta x}\delta_x(\tilde{\phi}p_t) - \frac{1}{\Delta y}\delta_y(\tilde{\phi}q_t) - \mu_x[\frac{1}{2}(p^2)_t] - \mu_y[\frac{1}{2}(q^2)_t] \tag{2.37}$$

Both relations, Eq 2.36 and Eq. 2.37, are valid for each cell in $\mathcal{D}$. In order to finish the proof, terms $\mu_x(\tilde{p}\tilde{r})$ and $\mu_y(\tilde{q}\tilde{s})$ have to be eliminated. This can be accomplished by a formal manipulation of Eq. 2.34. First the relation:

$$\frac{1}{\Delta x}\delta_x(\psi p)\frac{1}{\Delta y}\delta_y(\psi q) = -\Psi^2 + \mu_x(pr) + \mu_y(qs) \tag{2.38}$$

is obtained, and a simple observation can be made. In spite of Eq. 3.33, Eq. 2.38 can be rewritten for time-averaged quantities, because it was originally based on linear, spacial relations, that could have been time-averaged beforehand.

Recombining Eq. 2.36 and Eq. 2.37 with Eq. 2.38 (for time-averaged quantities), the formal equivalent of the energy principle per cell is obtained:

$$\begin{aligned}\mu_x[(\frac{1}{2}p^2)_t] + \mu_y[(\frac{1}{2}q^2)_t] + \tilde{\Psi}^2 + \tilde{\Phi}\tilde{f} = \frac{1}{\Delta x}\delta_x(\tilde{\phi}\tilde{r}) + \frac{1}{\Delta y}\delta_y(\tilde{\phi}\tilde{s}) \\ -\frac{1}{\Delta x}\delta_x(\tilde{\psi}\tilde{p}) - \frac{1}{\Delta y}\delta_y(\tilde{\psi}\tilde{q}) + \frac{1}{\Delta x}\delta_x(\tilde{\phi}p_t) + \frac{1}{\Delta y}\delta_y(\tilde{\phi}q_t)\end{aligned} \tag{2.39}$$

In order to produce a more transparent form, Eq. 2.39 has to be written for each cell of the domain and then the summation can be be performed. Some of the terms on the right-hand side cancel due to the requirement of continuity of normal derivatives, as well as the continuities of functions themselves, which is enforced implicitly through the notation already introduced. In the most consise form and neglecting the forcing function, the final result is given by:

$$\begin{aligned}\frac{1}{2\Delta t}\delta_t\|u\|^2 + \|\tilde{\Psi}\|^2 = \\ \Sigma_{i=1,\ldots,I,j=J}[\tilde{\phi}(\tilde{r} + p_t) - \tilde{\psi}\tilde{p}]\Delta y - \\ \Sigma_{i=1,\ldots,I,j=1}[\tilde{\phi}(\tilde{r} + p_t) - \tilde{\psi}\tilde{p}]\Delta y + \\ \Sigma_{i=I,j=1,\ldots,J}[\tilde{\phi}(\tilde{s} + q_t) - \tilde{\psi}\tilde{q}]\Delta x - \\ \Sigma_{i=1,j=1,\ldots,J}[\tilde{\phi}(\tilde{r} + q_t) - \tilde{\psi}\tilde{q}]\Delta x.\end{aligned} \tag{2.40}$$

The analogy with the Eq. 2.9 is complete in the sense that integral norms are replaced by the discrete-summation ones. Moreover, the homogenous boundary conditions for the function Φ and its normal derivatives are enforceable in a straightforward way. Therefore, the discrete solution is bound by the boundary and initial conditions and should converge well. In fact, second order accuracy is expected for the Φ function, its derivatives and its Laplacian. The *generalized dissipative boundary conditions* expressed in Eq. 2.9 and Eq. 2.40, need not to be exactly homogenous. The proper statement should be limited to the requirement of non-positivness of the right-hand sides of these equations. This fact is particularly tempting, because the Eq. 2.1 is a diffusion equation by itself, whereas the result of Eq. 2.40 holds regardless of what kind of boundary conditions are imposed on function Ψ. This is true, regardless whether the specific implementation of a numerical scheme, at some solution step does or does not impose some additional numerical boundary conditions on this function. Furthermore, there should be a plethora of cases where the diffusion equation, by itself, should have a unique and well-behaved numerical solution for some regions in $\mathcal{D}$, thus forcing the uniqueness of the other, coupled function Φ.

In relaxation of the requirements stated in Eq. 2.4, two points of view need to be taken into the account:

- First, as it was proven earlier by *Rose* [9],[10], for the diffusion equation, similar discrete energy principle can be derived. Therefore, the additional boundary conditions on Ψ in some region, say $\mathcal{D}_\psi$, is correct, provided that on the circumference of this domain $\partial\mathcal{D}_\psi$, the dissipative conditions for the diffusion equation are met. Moreover, for a coupled system of equations, the dissipative nature of the boundary conditions for a system may be a result of a coupling and a particular shape of the boundary.
- Second, for a number of important engineering problems, dissipative nature of the boundary conditions may sometimes be proven using fairly elementary physical arguments.

Example 1. Static load on a rectangular plate.

In Chapter 4, Eqs. 2.1 and 2.4 are used to describe deflection of a thin, square, homogenous plate under a static load f . The function Φ is interpretted as a displacement. Homogenous boundary conditions represent the solid support, and the zeroing of normal derivatives is usually referred to as "clamping".

If no "clamp" is required on one of plate's edges, say $y = 0$ or the "bottom" edge, then in the vicinity,

$$-\Psi = \frac{\partial^2\Phi}{\partial y^2} \tag{2.41}$$

and

$$u_n = \frac{\partial \Phi}{\partial y} \tag{2.42}$$

have the same sign, except for some very complicated distributions of the load. This is due to the fact that the u_n term represents the slope while the other term, $-\Psi$ describes the attenuation of this slope away from the edge. Since the displacement of all edges is still assumed to be zero, boundary conditions for the entire system are dissipative. In this case, the imposition of "unclamped" boundary condition for a specific edge is necessary, because the term $-\Psi u_n$ is senstitve to the orientation of the normal vector.

Example 2. Driven cavity flow.

In Chapter 5, modified Eq 2.1 and 2.3 are used to describe a flow in a square cavity. Functions Ψ and Φ represent vorticity and stream function, respectively. Homogenous boundary conditions imposed on the stream function correspond to no-penetration requirement appropriate for solid boundaries and/or streamlines, whereas normal derivatives of the same function, represent components of velocity, tangent to the boundaries, which are zero, due to the no-slip requirement. It is assumed that three of the walls are stationary and the motion of the fluid is imposed by a steady movement of the third "wall" in its own plane. In agreement with the formulas:

$$v_x = \frac{\partial \Phi}{\partial y} \tag{2.43}$$

and

$$v_y = -\frac{\partial \Phi}{\partial x}, \tag{2.44}$$

value of u_n represents the velocity of the "plate" driving the motion. At the same time, Ψ is the vorticity at a center of a cell adjacent to the moving boundary. It is a matter of a simple verification to realize that u_n and Ψ have always the same sign. Since homogenous boundary conditions for Ψ are still required, boundary conditions are dissipative.

Example 4. Mixing motion driven by an agitator.
Another example of fluid motion is of concern here, because it represents yet another way of imposing of boundary conditions on the system discussed in this report. It is assumed that on a circumference of a square domain $\mathcal{D}$, no-slip and no-penetration boundary conditions are imposed. At the center of the domain, at centers of some centrally-located cells, a sinusoidal variation of vorticity is required. This emulates a motion of a "square" agitator/impeller of a mixer.

The dissipative nature of boundary conditions is probably a result of the separation of energy principles for the vorticity and stream functions along some unspecified flow line. This can also be discussed using straightforward physical arguments. Consider the tem:

$$\psi u_n = \psi(-v_y n_x + v_x n_y). \tag{2.45}$$

In agreement with already introduced formulas, we will sum this term along the circumference of a square region $\mathcal{D}_\varepsilon$ surrounding all the cells representing the agitator. When vorticity Ψ has a positive value, then there will always be a region in which vorticity has a postitive value, due to imposed continuity of all functions. This holds, for grids which are fine enough and provided that it is the agitator that drives the fluid motion. Summation corresponding to the term :

$$\int \Psi(-v_y n_x + v_x n_y) dS \tag{2.46}$$

reveals that it is always *negative* . The other part of the integral depends on the proximity of the solid walls and the strength of the mixer. That is, in some specific cases, its absolute value may be smaller than the other part. This itself suggests that under some circumstances, the solution to the flow "inside" the agitator may be unique. As a result, the function Φ which is the stream function has a unique solution in the region $\mathcal{D}_\varepsilon$, where no additional conditions were ever imposed on this function.

In order to prove the existence of the solution in the domain $\mathcal{D}_\Sigma = \mathcal{D} - \mathcal{D}_\varepsilon$, we will consider a streamline, surrounding the agitating cells, but still internal to the region $\mathcal{D}_\varepsilon$. Since no mass addition is allowed, such a streamline exists, therefore boundary conditions are dissipative along this line. Moreover, they are such along the outer circumference of the domain $\mathcal{D}_\Sigma$.

Chapter 3

Note on Crank-Nicholson and ADI schemes

As it was pointed in [1] and in [9], the compact scheme for the diffusion equation in Cartesian elements can be reduced to a Crank-Nicholson or ADI method involving only the center-cell values.
Using the earlier-required continuities of normal derivatives, which apply only at the inner boundaries between elements, the *potential forms* are obtained.

Consider the following condition:

$$r^{-}_{i+1,j} = r^{+}_{i,j} \tag{3.1}$$

which in conjunction with Eq. 2.18 and Eq. 2.19 implies:

$$\psi_{i+\frac{1}{2},j} = \frac{1}{2}(\Psi_{i+1,j} + \Psi_{i,j}). \tag{3.2}$$

This procedure may be applied to any of the variables involved in the system, as well as to the derivatives, c.f. Eq's 2.14-2.21. For example:

$$r^{+}_{ij} = \frac{1}{\Delta x}(\Psi_{i+1,j} - \Psi_{i,j}) \tag{3.3}$$

$$r^{-}_{ij} = \frac{1}{\Delta x}(\Psi_{i,j} - \Psi_{i-1,j}) \tag{3.4}$$

For a coupled system of equations, Eqs. 2.1-2.2, boundary conditions are imposed on the function Φ, therefore for the Poisson subsystem the original formulation is kept. However, the diffusion subsystem may be cast in a lot simpler form. Referring specificaly to the Eq. 2.35, it will be noted that the time-stepping operation involves only center-cell values of function Ψ, provided that the right-hand side could be rewritten in a similar fashion. This is accomplished using identities 3.3 and 3.4. Further, we will prove that the entire analysis following the Eq 2.40 still holds.

- First, we will state that:

$$\frac{1}{\Delta x}\delta_x r_{ij} + \frac{1}{\Delta y}\delta_y s_{ij} = \tag{3.5}$$

$$\frac{1}{\Delta x^2}(\Psi_{i+1,j} - 2\Psi_{i,j,} + \Psi_{i-1,j}) + \frac{1}{\Delta y^2}(\Psi_{i,j+1} - 2\Psi_{i,j,} + \Psi_{i,j-1})$$

  holds for any "inner" ij-th Cartesian element. This relation has the form of the Crank-Nicholson scheme, except that it requires the imposition of boundary conditions at centers of cells, adjacent to the boundary of the computational domain. For the coupled system of equations, however. this issue may be resolved by an inspection of Eq. 2.34. It will be noted, that for the Laplace subsystem, all normal derivatives were retained, therefore for all boundary cells, conditions 2.3 and 2.4 are enforceable. That way, conditions needed for the values of Ψ at the centers of these cells are obtained.

- Second, except for the boundary cells, relations 2.39 hold, together with the continuity of functions and normal derivatives. Therefore, Eq. 2.40 is valid, provided that the original formulation is retained for all cells adjacent to the boundary.

  Let's, accept such a scenario; that is, assume that Eqs. 2.33 and 2.34 are valid there. Now, the off-boundary values of normal derivatives can be replaced using the formulas similar to Eqs. 3.3 and 3.4, while the off-boundary values of Ψ can be replaced using Eqs. 3.1 and 3.2. Furthermore, as it was already noted before, for each of the boundary cells an equation describing the center-cell value was already obtained, therefore by subtraction, additional equations for values of normal derivatives at all boundaries would be produced. However, using the half-step formulas Eqs. 2.18-2.21 and obtained center-cell values of Ψ, "edge" values of this function would be immediately produced. Finally, except for the corner cells, where some caution has to be exerted, relations for the boundary values of function Ψ and its normal derivatives could be solved *a posteriori* , permitting one to handle the diffusion subsystem as a Crank-Nicholson scheme.

For corner cells, the conditions stated by Eq's 2.3 and 2.4 will be strictly enforced. This allows for a straightforward analysis of edge values of function Ψ.

As it was indicated in [2] the alternative direction implicit method [ADI] offers a very interesting alternative to the Crank-Nicholson scheme. It has been applied to Stokes-type flows before, in [2] and [7]. In [6] the scheme was proven to be adaptable to the compact scheme formulation. For a two-dimensional formulation, ADI may be best described as a half-time-step method with x- and y- differentiations frozen alternatively, resulting in equations being tridiagonal.

Unfortunately, for coupled systems of equations, this scheme reveals high sensitivity to boundary conditions. In fact, a very complicated backtracking is usually necessary in order to assure the numerical stability. It would be very hard to review how this numerical instability is offset by different numerical gadgets. We will limit ourselves to only one general statement. Refering to Eq. 2.39, it is safe to say that in handling x- and y- differentiation at different time levels, there is simply no method to guarantee that the boundary conditions remain dissipative at the same time level.

Chapter 4

Applications in Solid Mechanics

Numerical solutions of two systems of equations were analyzed for efficiency and accuracy. The first system, Eq's 4.1-4.2 below, describes a steady deflection of a thin, square, clamped plate, under a distributed load. The second system, Eq's 4.6-4.7, is an extension of the former one, where additional time dependence is included. In a rigorous sense, the second system does not describe the time evolution of the solution to a thin plate problem, it however serves as a model for other problems in continuum mechanics, like fluid dynamics, chemical flows and heat transfer. It is particularily suited for two-dimensional Stokes flows. Some examples of practical interest in fluid dynamics will be considered in the next Chapter.

Steady Problem

The steady problem, may be solved directly using the transfer of boundary conditions described already. All the existence and uniqueness conditions hold just like in the unsteady case. However, the system itself reduces to a pair of coupled Poisson equations and is not very interesting. Therefore as far as numerics is concerned, all the steady cases will be treated as asymptotic (in time) cases of the unsteady formulation.

Consider three functions $u = u(x, y), w(x, y)$ and $f_1(x, y)$ fullfilling the following system of equations:

$$\begin{aligned} \nabla^2 w &= f_1 \qquad (4.1) \\ \nabla^2 u &= -w \qquad (4.2) \end{aligned}$$

for $0 \leq x \leq\ 1$ and $0 \leq y \leq 1$

This system is usually rewritten in terms of a single fourth-order equation:

$$\nabla^4 u = -f_1 \qquad (4.3)$$

where typical boundary conditions are:

$$u(x, 0) = u(x, 1) = u(0, y) = u(1, y) = 0 \qquad (4.4)$$

and

$$\frac{\partial}{\partial y} u(x, 0) = \frac{\partial}{\partial y} u(x, 1) = \frac{\partial}{\partial x} u(0, y) = \frac{\partial}{\partial x} u(1, y) = 0 \qquad (4.5)$$

In general, system of Eq 4.1-4.2 with boundary conditions of Eq 4.4-4.5 may be cast in terms of two eliptic equations with Dirichlet-type boundary conditions. Then, the coupling is accomplished through the forcing term in the second, and through the boundary conditions for the first equation. Specific formulation of the boundary conditions depends on the numerical strategy to be used.

Solution to the steady problem, may be viewed as an asymptotic solution to the unsteady problem, presented next.

Unsteady Problem

Time-dependence is introduced *ad hoc* in the following way:

$$\begin{aligned} \frac{\partial}{\partial t} w &= \nabla^2 w - f_2 && (4.6) \\ \nabla^2 u &= -w && (4.7) \end{aligned}$$

Together with boundary conditions Eq 2.4 and initial condition:

$$u(0,x,y) = u_0(x,y) \tag{4.8}$$

Boundary conditions for Eq. 4.6 are specific to the strategy and are discussed as a part of the discretization process. Herein we will only point out to the discussion following the Eq. 3.5. It was indicated there that the conditions of the type:

$$p^{-}_{1j} = 0 \tag{4.9}$$

are sufficient to produce the boundary relations for Eq. 4.8.

Discretization Process

The square $[0,1] \otimes [0,1]$ is divided into I^2 square cells, where indices i,j refer to cell's centers. Time, t, is indexed with the superscript, n, and the following notation is used:

$$\begin{aligned} t^1 &= 0, && (4.10) \\ t^{n+\frac{1}{2}} &= t^n + 0.5 * \Delta t, && \\ t^N &= T_{max} && \\ w^n_{i,j} &= w(t^n, x_i, y_j), && (4.11) \\ \phi^n_{i,j} &= u(t^n, x_i, y_j), && (4.12) \\ u^n_{i+,j-} &= u(t^n, x_i + 0.5\Delta x, y_j - 0.5\Delta y). && (4.13) \end{aligned}$$

Paceman-Rachford-type alternative direction implicit (ADI) scheme [2] for Eq. 3.6. takes the form:

$$\begin{aligned} w_{i,j}^{n+\frac{1}{2}} - w_{i,j}^{n} = {} & \lambda_x \quad (w_{i-1,j}^{n+\frac{1}{2}} - 2w_{i,j}^{n+\frac{1}{2}} + w_{i+1,j}^{n+\frac{1}{2}}) \\ +\lambda_y(w_{i,j-1}^{n} - 2w_{i,j}^{n} + w_{i,j+1}^{n}) - 0.5\Delta t f_{2i,j}^{n+\frac{1}{2}} & \qquad (4.14) \\ w_{i,j}^{n+1} - w_{i,j}^{n+\frac{1}{2}} = {} & \lambda_x \quad (w_{i-1,j}^{n+\frac{1}{2}} - 2w_{i,j}^{n+\frac{1}{2}} + w_{i+1,j}^{n+\frac{1}{2}}) \\ +\lambda_y(w_{i,j-1}^{n+1} - 2w_{i,j}^{n+1} + w_{i,j+1}^{n+1}) - 0.5\Delta t f_{2i,j}^{n+\frac{1}{2}}, & \qquad (4.15) \end{aligned}$$

where coefficients λ_x and λ_y are given by:

$$\lambda_x = \frac{0.5\Delta t}{\Delta x * \Delta x} \qquad (4.16)$$

$$\lambda_y = \frac{0.5\Delta t}{\Delta y * \Delta y} \qquad (4.17)$$

Boundary conditions for Eq's 4.14 and 4.15 are not known apriori therefore they have to be calculated at each step from the solution to the scheme for Eq. 4.7. As already mentioned, the Poisson problem Eq. 4.7 is cast in terms of a Compact Finite Volume Method [1]. In the notation used previously, this scheme can be rewritten in the following way:

$$\begin{aligned} w_{i,j}^{m} + \gamma_x(u_{i-,j}^{m} - 2\phi_{i,j}^{m} + u_{i+,j}^{m}) + \gamma_y(u_{i,j-}^{m} - 2\phi_{i,j}^{m} + u_{i,j+}^{m}) &= 0, \quad (4.18) \\ \phi_{i-1,j}^{m} - 2\, u_{i-,j}^{m} + \phi_{i,j}^{m} &= 0, \quad (4.19) \\ \phi_{i,j-1}^{m} - 2\, u_{i,j-}^{m} + \phi_{i,j}^{m} &= 0. \quad (4.20) \end{aligned}$$

For square cells, $\Delta x = \Delta y$, it is prudent to take $\gamma_x = \gamma_y = 1$ rather than $\gamma_x = \frac{2}{\Delta x * \Delta x}$ and $\gamma_y = \frac{2}{\Delta y * \Delta y}$. This procedure is equivalent to the scaling of the w function and it improves the conditionning of the entire system. Now on, this rescaling will be implicitly assumed.

Boundary conditions for the Eq. 4.18, 4.19 and 4.20 are generated using Eq.4.4 and this subsystem may be viewed as a Dirichlet problem for the Poisson equation. In order to account for the boundary conditions expressed by Eq. 4.5, for cells adjacent to boundary of the computational domain one would welcome relations of the type:

$$\begin{aligned} u^m_{1-,j} = \phi^m_{1,j} &= 0 && (4.21)\\ u^m_{I+,j} = \phi^m_{I,j} &= 0 && (4.22)\\ u^m_{i,1-} = \phi^m_{i,1} &= 0 && (4.23)\\ u^m_{i,I+} = \phi_{i,I} &= 0 && (4.24) \end{aligned}$$

New boundary conditions for the Eq. 4.14 and Eq.4.15. follow directly from the Eq. 4.21-4.24. and the definition of the compact scheme. Refering to the Eq. 4.2. specifically, for the lower ($y = 0$) boundary, the following relations are obtained:

$$w^m_{i,1} = -\gamma_x(u^m_{i-,1} + u^m_{i+,1}) - \gamma_y(u^m_{i,1+}). \tag{4.25}$$

Similar relations hold for all the cells adjacent to the boundaries of the computational domain and need not to be repeated here. Eqs. 4.5, although similar in form, are not substituted for any of Eqs. 4.18. They, together with the Eq. 4.14. and Eq. 4.15. constitute a Dirichlet problem for the diffusion equation. The entire system is now specified for a set of $2I*(2I+1)$ variables, per each half-time step $m = 1, 1+\frac{1}{2}, 2, 2+\frac{1}{2} \ldots$

ADI Routine

The very first attempt to solve the problem was essentialy based on methods presented by Peyret and Taylor [2]. Tridiagonal solver for Eq's 4.14-4.15 was implemented. Similarily, Eqs. 4.18-4.20 were cast in time-dependent form and a specialized tridiagonal solver, accounting for the continuity relations, Eqs. 4.19-4.20, was developed. Boundary conditions for the system were produced at each time half-step. Moreover, since Eq. 3.25 does not contain relaxation parameter, various methods of such inclusion were tested. The system was, nevertheless persistently unstable.

Compact scheme does not benefit from explicit Taylor expansions, therefore higher-order boundary conditions become rather convoluted. Except for methods of Forouk and Fusegi [7], one is still faced with the necessity of backtracking in time in order to correct the mid-step boundary values. Rather than using these complicated methods of obtaining proper boundary conditions, a global approach based on successive overrelaxation routine (SOR) was tested.

Kaczmarz Routine

The system of algebraic equation, Eqs. 4.14-4.15 and Eqs. 4.18-4.20 together with the suitable boundary and initial conditions was solved using Kaczmarz Successive Overrelaxation Scheme [3]. A similar approach was used for an earlier version of a Compact Scheme by Gatski, Grosch and Rose (GGR code, [4]). As noted in [4], Kaczmarz algorithm is very easy to implement, because equations may be programmed in groups and, therefore amenable to the out-of-core computations one might add. For a particular system considered herein, the $\mathcal{A}$ matrix is not only sparse but it has almost cellular form. It is quite easy to pivot just few rows in order to obtain a system which can be vectorized and synchronized (parallelism) to a very large extent. The most simple, although somewhat redundant example of such a procedure can be given by the following. By subtracting Eq. 4.25 from a suitably indexed Eq. 4.18, and using the fact that function u has zero values on the boundary, and carrying similar procedures for all cells adjacent to the boundary of the computational domain, complicated boundary conditions expressed by equations similar to Eq. 4.25 may be replaced by $\phi_{ij} = 0$. A dearth of similar strategies is possible, provided that available computer has at least four processors and a typically slow convergence of SOR algorithm does not hamper the computational process too much. Results presented in this subsection and parts of the next one were obtained on a 64-bit Convex C120 computer working in a sequential mode.

For each half-time step, all dependent variables can be renumerated and absorbed into vector X, with $J = 2I * (2I + 1)$ entries. System can be symbolically written as:

$$\mathcal{A}_i * X = b_i \tag{4.26}$$

where $\mathcal{A}_i$ is a row-vector corresponding to an i-th equation of the algebraic system. We will assume that the L^2 norm of $\mathcal{A}_i$ is always 1.

Define the residue to be: $res_i = \mathcal{A}_i * X - b_i$

and introduce X^0 as the initial guess for the vector of unknowns, then the Kaczmarz procedure consists of the following:

Step 1. $X^n_{i+1} = X^n_i - \alpha \mathcal{A}_i res^n_i; for : i = 1, \ldots J - 1, n = 0, 1, 2, \ldots$

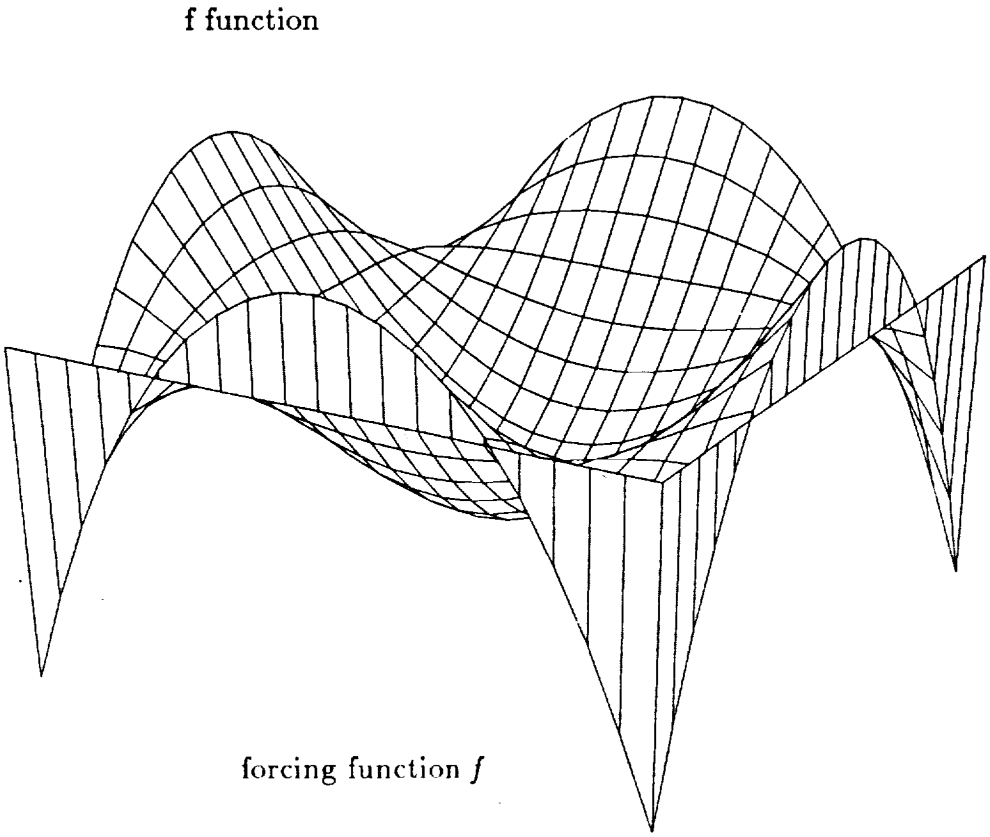
Step 2. $X^{n+1}_1 = X^n_J$

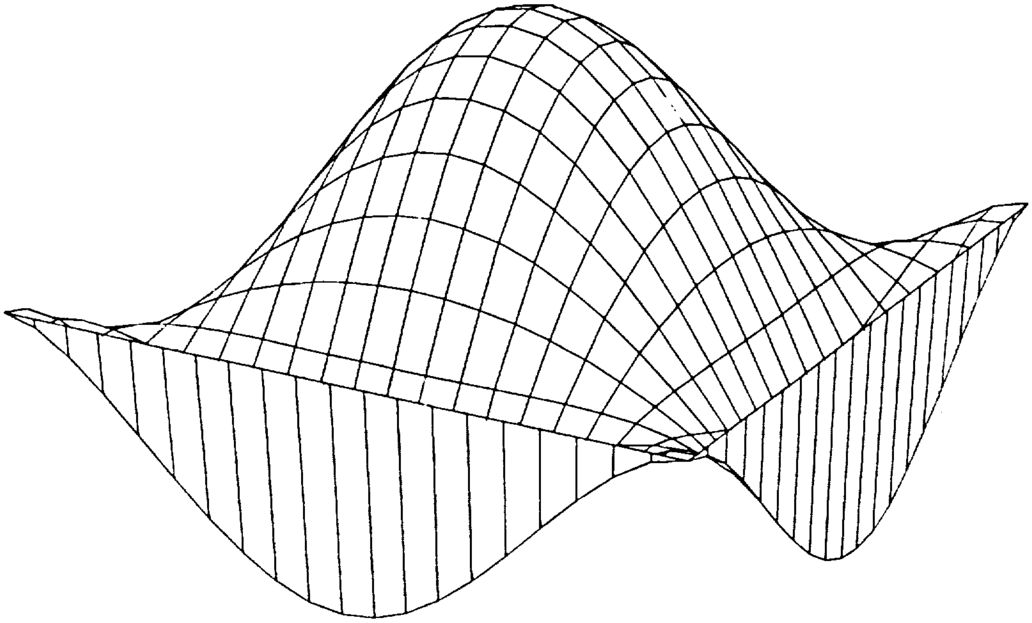
The procedure is repeated until the absolute value of residue res achieves required minimum. The value of the relaxation parameter α should vary between 1 and 2.

Results I

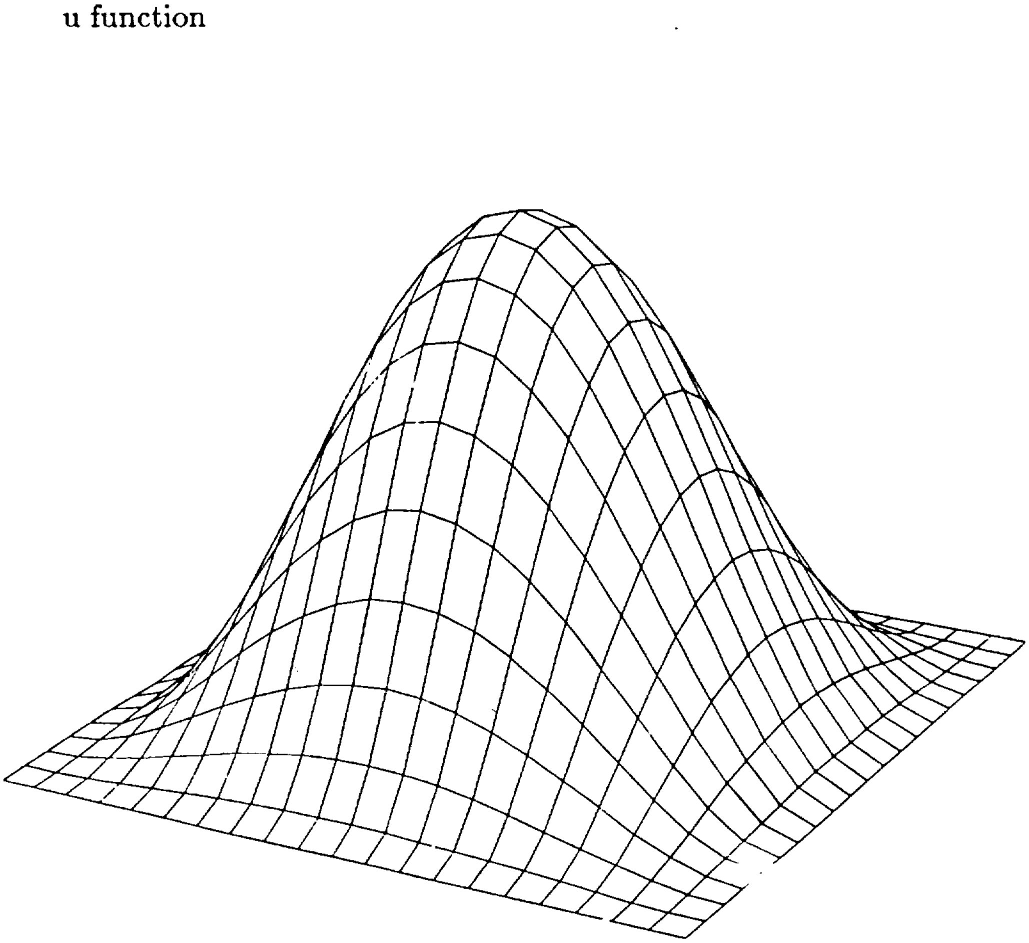
For all the testing some analytic functions had to be selected. They were constructed in the following way: The x- and y-dependence as well as time-dependence were introduced through $X = x^2 * (1 - x)^2, Y = y^2 * (1 - y)^2$ and

$T = 0.5 * (1 - t) * (2 - t)$ and the function u was introduced by $u = T * X * Y$. Functions w and f_1 (f_2), needed for calculations and error norm evaluations, were obtained analytically by substitutions into Eq. 4.2 and Eq. 4.1 (or Eq.4.6), respectively. Shapes of functions u, w and f_1 are drawn in Fig.2.





function w



displacement u

Fig.2 Functions u, w and f_1 .

Steady Problem

The solution process of the *steady* problem involves the ADI scheme, therefore the ' t ' variable had to be interpreted as 'ficticious' time and $T = 1$. In this case, the actual solution becomes an asymptotic (in time) solution to the *unsteady* problem and the initial guess becomes merely the *initial condition.* For all runs with 'ficticious' time, $u_0 = 0$ was selected, however in order to speed up the process a simple tridiagonal solver for Eqs. 4.14. and 4.15. was introduced ahead of the Kaczmarz routine. This solver was used only once for the very first half-time step.

In order to establish the second-order accuracy of the method, maximum error and l^2 error norms for u and (unscaled) w functions were computed for various grid coarsness. Using the values of functions u and w obtained analitically following the norms were computed:

$$\| u \|_{max} = max \mid u_{anal} - u_{comp} \mid$$

$$\| w \|_{max} = max \mid w_{anal} - w_{comp} \mid$$

$$\| u \|_2 = \frac{1}{2I*(2I+1)}\left(\sum \left(u_{i-+,j-+anal} - u_{i-+,j-+comp}\right)^2\right)^{\frac{1}{2}}$$

$$\| w \|_2 = \frac{1}{I}\left(\sum \left(w_{i,janal} - w_{i,jcomp}\right)^2\right)^{\frac{1}{2}}$$

Table 1.a Error norms for u and w functions

maximum residue = 10^{-5}

no.ofcells	$\| u \|_{max}$	$\| u \|_2$	$\| w \|_{max}$	$\| w \|_2$
3^2	0.00160326	0.00171	0.02515275	0.01977
6^2	0.00036485	0.00060	0.01268210	0.00610
12^2	0.00010021	0.00021	0.00489993	0.00152

Table 1.b Error norms for u and w functions

maximum residue $= 10^{-7}$

$no.of cells$	$\| u \|_{max}$	$\| u \|_2$	$\| w \|_{max}$	$\| w \|_2$
3^2	0.00160131	0.00171	0.02518925	0.01979
6^2	0.00036470	0.00060	0.01264418	0.00608
12^2	0.00009965	0.00021	0.00485032	0.00155

As seen from the Table 1, above, the system gives a second-order accurate solution to the biharmonic equation. The accuracy of the auxiliary function w is lagging because of rescaling used to improve the conditioning of the entire system.

The relaxation parameter α should be optimal, in the sense that it should reduce to the minimum the number of iterations required to satisfy the residue criterion Experiments showed that the optimal value of the relaxation parameter α varies strongly with the number of cells in the computational domain.

Table 2. No of Cells versus optimal α

3^2	1.54
5^2	1.75
6^2	1.84
10^2	1.90
12^2	1.91
20^2	1.82

Even for an optimal α, the number of iterations needed to achieve the required minimum varied strongly with the number of cells and the value of the threshhold.

Table 3. No of Cells versus the number of iterations

$res_1 = 10^{-5}, res_2 = 10^{-6}, res_3 = 10^{-8}$

$no.of cells$	res_1	res_2	res_3
5^2	62	72	123
10^2	1016	1201	...
20^2	35628	...	...

Results presented in Table 3. show that the number of necessary iterations grows rapidly with the number of cells. Moreover, it is inversely proportional to the degree of tolerance imposed on the residue of the system.

Unsteady Problem

Because of the rescaling of the w -function, solutions to the unsteady problem become very expensive; extremly low tolerance on residue has to be required in order to prevent errors from becoming much larger than the actual of values of the solution. Nevertheless, a steady problem can still be handled using a supercomputer with an advantage of significantly lower memory rquirements.

Gauss Routine

The idea of pivoting of the $\mathcal{A}$ matrix, introduced previously was brought to a natural conclusion by replacing the SOR algorithm with a simple Gauss transformation with pivoting. The simple **IBM** *matinv* subroutine was adopted. The ADI scheme was also abandoned in favor of Crank-Nicholson scheme, for Eq. 3.6. For completeness it is given below:

$$\begin{aligned} w_{i,j}^{n+1} - w_{i,j}^{n} = \; & \lambda_x \; (w_{i-1,j}^{n+1} - 2w_{i,j}^{n+1} + w_{i+1,j}^{n+1}) \\ + \; & \lambda_y \; (w_{i,j-1}^{n+1} - 2w_{i,j}^{n+1} + w_{i,j+1}^{n+1}) \\ + \; & \lambda_x \; (w_{i-1,j}^{n} - 2w_{i,j}^{n} + w_{i+1,j}^{n}) \\ + \; & \lambda_y \; (w_{i,j-1}^{n} - 2w_{i,j}^{n} + w_{i,j+1}^{n}) - \Delta t f_{2i,j}^{n+\frac{1}{2}} \end{aligned} \qquad (4.27)$$

Introduction of the Gauss transformation allowed for a more efficient study of properties of solutions. Results presented in Table 1. were recalculated using *matinv* routine and are presented below:

Table 4. Error norms for u and w functions

steady case

$no.of cells$	$\| u \|_{max}$	$\| u \|_2$	$\| w \|_{max}$	$\| w \|_2$
3^2	0.00154321	0.00167	0.02623457	0.02002
6^2	0.00036470	0.00060	0.01264307	0.00608
12^2	0.00009965	0.00006	0.00484978	0.00155

The corresponding *unsteady* results are presented in Table 5. They were obtained using the same Gauss subroutine but on Cray Y-MP supercomputer. The upper entries correspond to the error values for time, t=1.0 and the lower entries correspond to the error values at t=2.0, cf. the definition of function T, in this subparagraph.

Table 5. Error norms for u and w functions

unsteady case

$no.of cells$	$\| u \|_{max}$	$\| w \|_{max}$	$\| w \|_2$
3^2	0.00000233	0.00016804	0.00006
...	0.00000195	0.00014060	0.00005
6^2	0.00000138	0.00003912	0.00002
...	0.00000125	0.00003486	0.00002
12^2	0.00000040	0.00001284	0.000007
...	0.00000036	0.00001206	0.000006

Results presented in Table 5 show the "quality" of the approximation, since in the considered example values of functions u and w for t=1.0,2.0 are zero.

Chapter 5

Applications in Fluid Dynamics

In [2], the numerical treatment of a stream function - vorticity formulation for two-dimmensional Stokes flow was described in detail. Equations are usually written in the form:

$$
\begin{aligned}
\frac{\partial}{\partial t}\omega + (\bar{a} * \nabla)\omega &= \frac{1}{Re}\nabla^2\omega && (5.1)\\
0 &= \nabla^2\chi + \omega && (5.2)\\
a_x &= \frac{\partial\chi}{\partial y} && (5.3)\\
a_y &= -\frac{\partial\chi}{\partial x} && (5.4)
\end{aligned}
$$

In the equations above, ω is vorticity, χ - stream function, and a_x and a_y are components of velocity vector. Re, is the Reynolds number.

Herein, an application of a *weak* compact scheme for handling of boundary conditions is considered. In general, whenever vorticity is involved in the problem, the *strong* version has to be implemented, because derivatives normal, as well as tangent to the faces of the cells are necessary. However, for a test case of wall-bound, vortical flows and a specific use of the scheme discussed here, it is still possible to employ only the less complicated, *weak* scheme and gain considerable savings in CPU time and memory.

The form of Eq. 5.1-5.2 is essentially the same as Eq. 4.6-4.7. Values of vorticity ω are evaluated at center-cell locations in the customary way. Second-order formulas for convection terms are given by:

$$\left(\frac{\partial\chi}{\partial y}\right)_{ij} = \frac{\chi_{i,j+} - \chi_{i,j-}}{\Delta y} \tag{5.5}$$

$$\left(\frac{\partial\chi}{\partial x}\right)_{ij} = \frac{\chi_{i+,j} - \chi_{i-,j}}{\Delta y} \tag{5.6}$$

$$\left(\frac{\partial\omega}{\partial x}\right)_{ij} = \frac{\omega_{i+1,j} - \omega_{i-1,j}}{2\Delta x} \tag{5.7}$$

$$\left(\frac{\partial\omega}{\partial y}\right)_{ij} = \frac{\omega_{i,j+1} - \omega_{i,j-1}}{2\Delta y} \tag{5.8}$$

A standard Newton-Raphson procedure was used to handle the nonlinear terms. In the final run the *matinv* subroutine was abandoned in favor of its fully vectorized counterpart from Linpack [5]. Computations were carried out on Cray Y-MP supercomputer at North Carolina Supercomputing Center, Research Triangle Park, NC.

Special care was taken to vectorize the code in order to make it efficient. Except for I/O operations and some complicated indexing routines, the code is thoroughly vectorized. Statistics showed that 60-87 percent of the time was spent in a subroutine rearanging Newton-Raphson iterations.

Note on Upwind Differentiation

In Chapter 3 the method of the transfer of boundary conditions was described in detail. The important facet of this methods was the elimination of "edge" values of the function Ψ. The entire procedure may be interpreted as a solution in the sense of staggered grids. Moreover, as it was shown in [6], such a formulation improves the accuracy. For fluid dynamics what is even more important, is the fact that there was no need to even consider values of vorticity at solid wall boundaries. In a sense, the center-cell value of vorticity is a resultant of mid-face values of velocity, c.f. Eq 2.34. This statement can, in principle be carried in full for the potential

form of the *compact scheme* . There are, however some conceptual obstacles that need to be addressed before such a programme can be implemented.
We will rewrite the system 5.1-5.4 in a form:

$$\mathbf{u} = \nabla\Phi \tag{5.9}$$

$$\nabla * \mathbf{u} + \Psi = 0 \tag{5.10}$$

$$\mathbf{v} = \epsilon\nabla\Psi - \overline{a}\Psi \tag{5.11}$$

$$\frac{\partial}{\partial t}\Psi = \nabla * \mathbf{v} - f \tag{5.12}$$

$$a_x = u_y \tag{5.13}$$

$$a_y = -u_x \tag{5.14}$$

System 5.10-5.15 is equivalent to Eqs. 5.1-5.4, because Eqs. 5.9, 5.13 and 5.14 guarantee that the divergence of the velocity is zero. Moreover, the energy principle is readily available:

$$\begin{aligned} &\frac{1}{2}\frac{d}{dt}\int u^2 d\forall + \epsilon\int \Psi^2 d\forall + \\ &\int \Phi f d\forall - \int \Psi(\overline{a} * \nabla\Phi) d\forall = \\ &\oint [\Phi(\frac{\partial}{\partial t}u_n + v_n) - \epsilon\Psi u_n] dS \end{aligned} \tag{5.15}$$

Comparing the above equation with Eq. 2.9, we note that although there is a slight difference in the definition of the **v** vector, both equations have essentially the same form. There is an additional term in Eq. 5.15. Using Eq. 5.10 and 5.15 it can be proven that it is equal zero. Now, the entire analysis following the Eq. 2.9 may be easily repeated for the system 5.9-5.14.
Surprisingly, this additional term which is handled so easily for the differential equation, causes rather severe problems in finite difference counterparts. This is due to the fact that for $\epsilon = \frac{1}{Re} >> 1$ the system 5.1-5.4 behaves quite like a hyperbolic one. In terms of numerical analysis, such a problem is resolved using

the "upwinding" or "upwind differentiation". Methods for such a treatment were already developed in [10], but they were limited to a single diffusion equation. The essence of that modification consists of introduction of cell-Reynolds-number-dependent weights into formulas of the type 2.14-2.21.
For a coupled system of equations, such as the one described in this report, "upwinding" has to be performed consistently for the stream function and the vorticity. Otherwise, due to the nature of the extra term in Eq. 5.16, no convergence could be guaranteed. In itself such a procedure is not very complicated. True difficulties arise when the upwind differentiation is employed in the study of wall-bound vortices. This is a topic for a separate study and only the approximate formulas 5.5-5.9 will be used. Resulting numerical instabilities are not severe, they are easily pinpointed by the inspection of motion graphics, developed for the project.

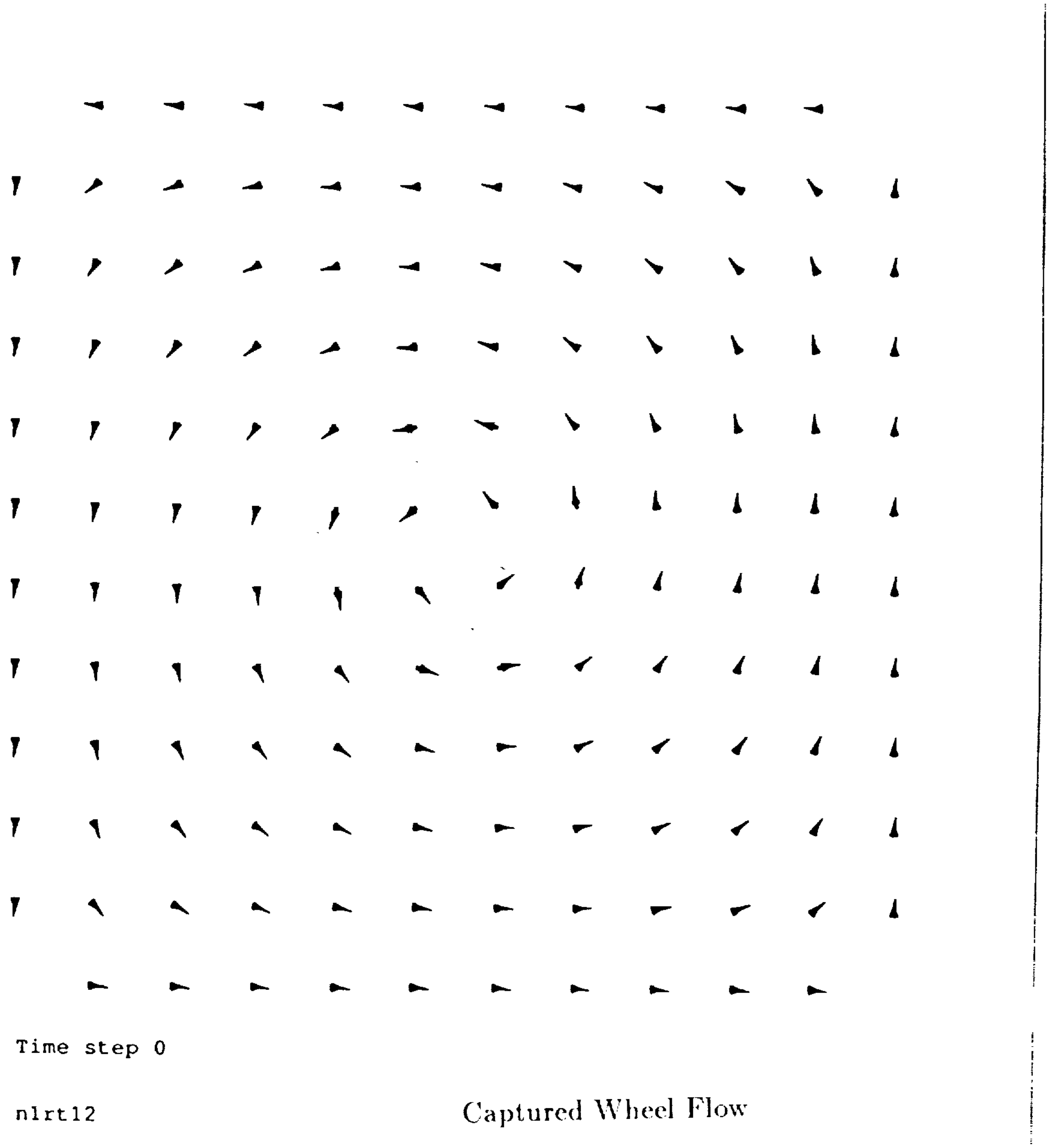
Results II

Three practical problems presented below, have no known analytic solutions, therefore the analysis has to rely on a careful study of pictures, including the animations recorded on a video. Only results for 12x12 cell runs are presented. The choice of flows was not accidental. In order to keep things simple the advantage was taken from the fact that for the wall-bound vortices the value of stream function χ on the circumference of the computational domain can be taken equal to zero without the loss of generality.

Captured Wheel Flow.

A two dimensional vortex is initially at a steady state defined by $V_\theta = c * r$. Vorticity has a constant, positive value. Instantaneously, impermeable walls are impressed upon the flow. Walls form a square concentric with the center of the vortex. In figures presented below, Reynolds number is defined as $\mathbf{Re} = \frac{2*c*L^2}{\nu}$, where c is the proportionality constant of the vortex, L the length of the side of the bounding wall, ν is the coefficient of kinematic viscosity. Boundary conditions used for this flow are standard for viscous flows. In the numerical context, they become the same as the ones used for the clamped plate, in the previous section.

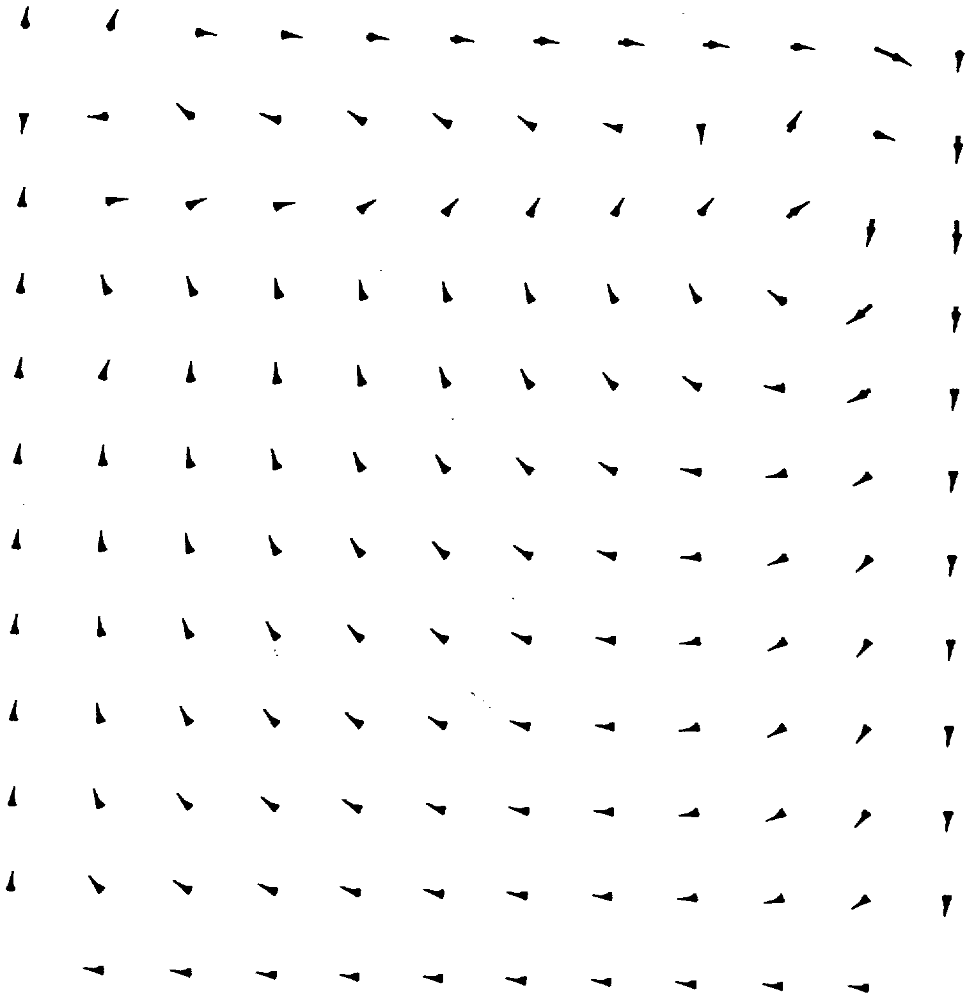
Fig.3 Captured Wheel Flow **Re=10000** .



Because of the continuous decay of the velocity field, no instabilities were detected.

Driven Cavity Flow.

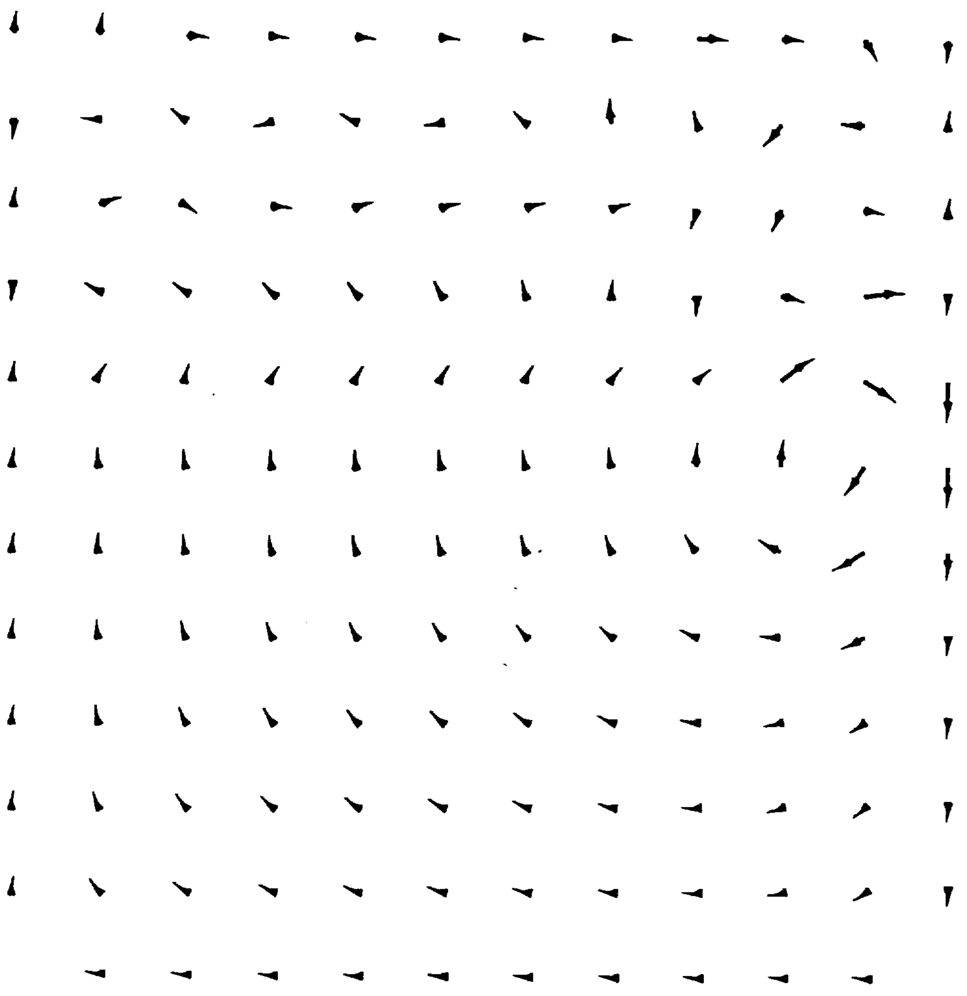
A two-dimensional flow in a rectangular cavity, Fig. 4, is assumed to be viscous and initially at rest. The walls are impermeable. The velocity **U** is impressed upon the topmost layers of the fluid, instantenously. Reynolds number is based on the length of the gap and the magnitude of the impressed velocity.



Time step 50

ngap12

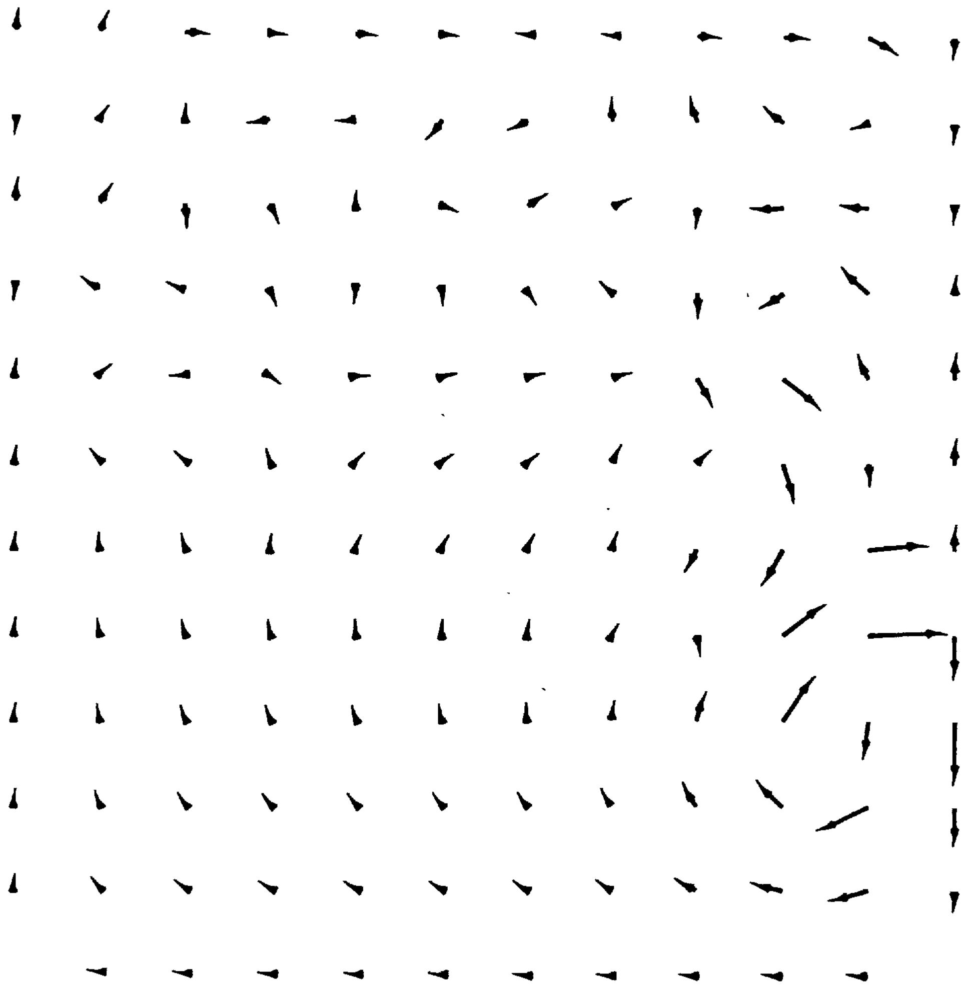
Driven Cavity Flow



Time step 100

ngap12

Driven Cavity Flow



Time step 150

ngap12

Driven Cavity Flow

Fig.4 Driven Cavity Flow, velocity field at different times, **Re=10000** .

As in the previous case, setting the value of stream function χ equal to zero at the circumference of the computational domain automatically enforces components of velocity normal to the wall to be zero. Components of velocity tangent to the boundary are defined as derivatives of χ normal to the wall. Except the side open to the outside flow, all tangent components of velocity are zero. As discussed in the paragraph on implementations of Kaczmarz algorithm, it is legitimate to impose conditions on normal derivatives of stream function, by manipulating the value of this function at centers of cells adjacent to the boundary. The result is written explicitly:

$$\chi_{iI} = -0.5\Delta y * U_i \qquad (5.16)$$

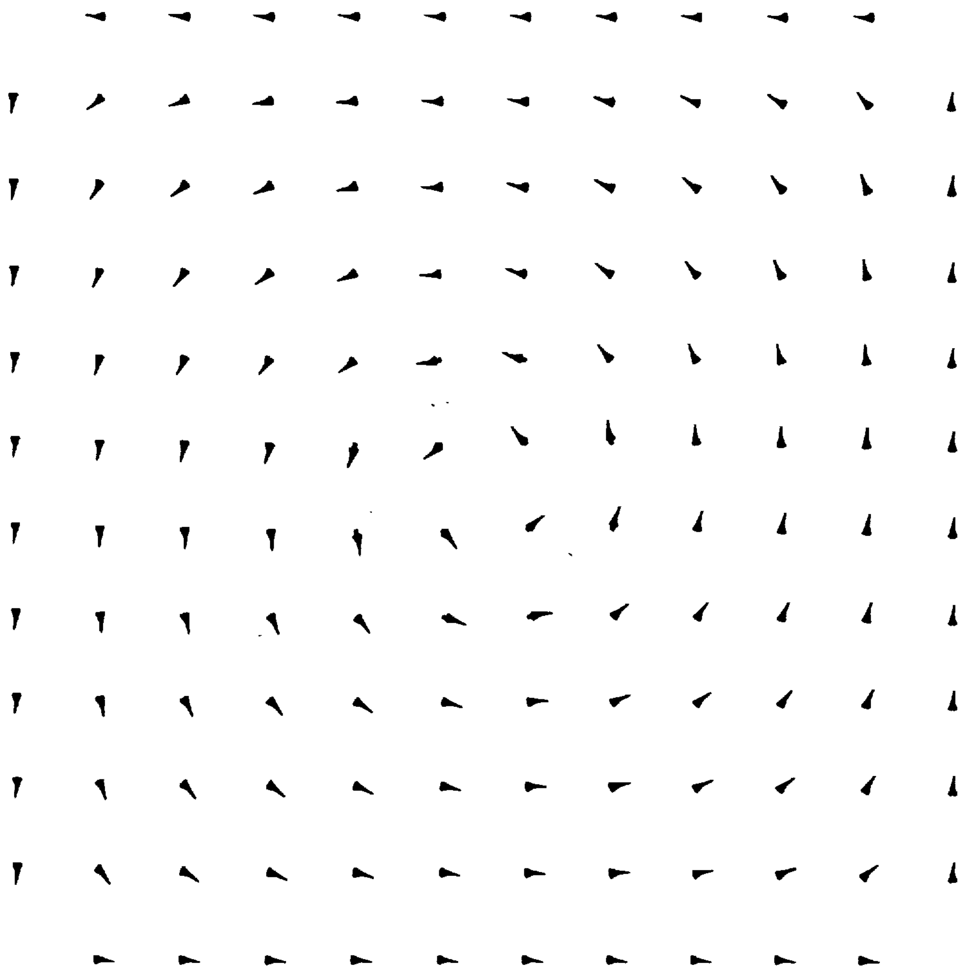
In this study only uniform values of velocity **U** are considered , but in general, any distribution of velocity is allowed.

Corners of the computational domain, adjacent to the open flow, are singular points of the solution. Typically, various types of shifts are applied there, but for the presented scheme, experiments showed that almost any set of boundary conditions works well. It was decided to use the no slip boundary conditions in these two corners.

Numerical instability is detectable after the influence of the upper plate reaches the right, bottom corner of the domain.

Oscillatory Flow.

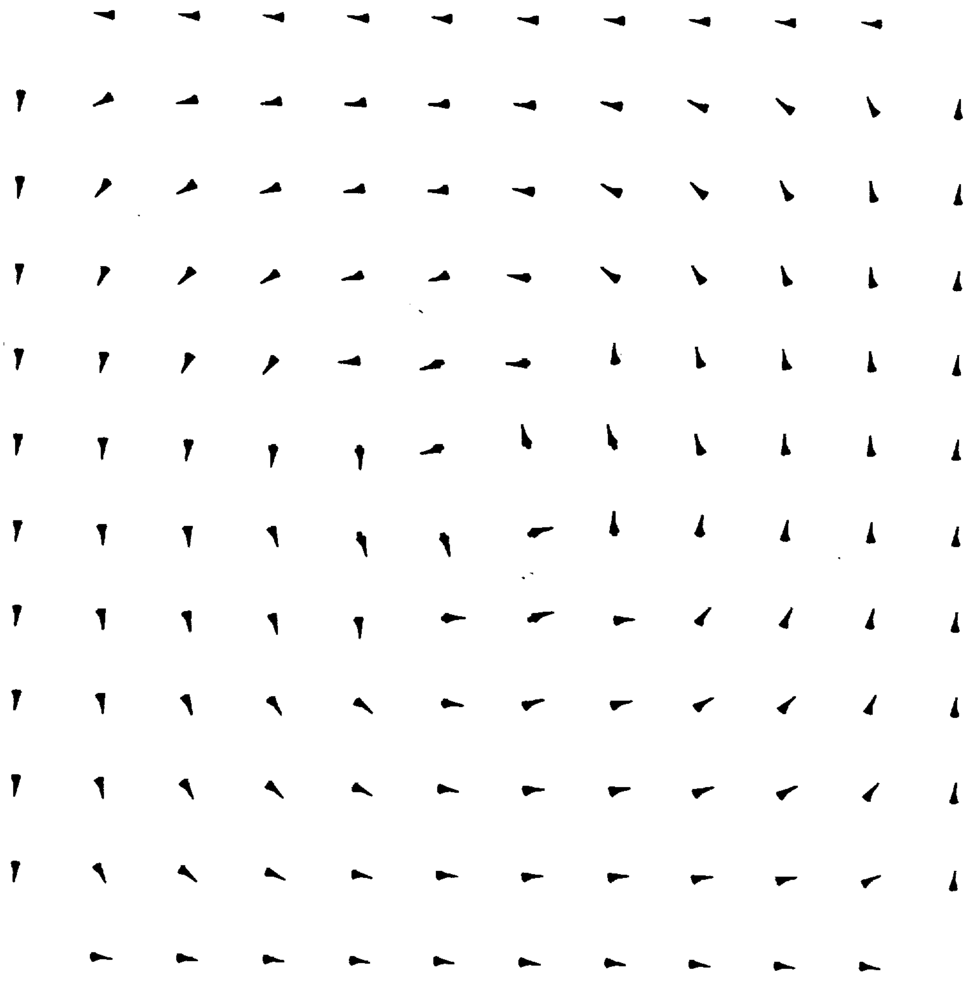
Finally an oscillatory type of flow is presented. Here, a two-dimensional, viscous, unsteady flow is enforced by a periodic variation of the vorticity at the center of the domain. The intensity of the vorticity at the core is scaled with the frequency of oscillations f , and only one dimensionless parameter, $\mathbf{Re} = \frac{\mathbf{fL^2}}{\nu}$, appears in the equations. Boundary conditions at the wall are identical to the captured wheel flow. Initial conditions correspond to the stagnant flow. Fig. 5 represents various stages of the development of the oscillatory flow.



Time step 0

v2rt12.grid

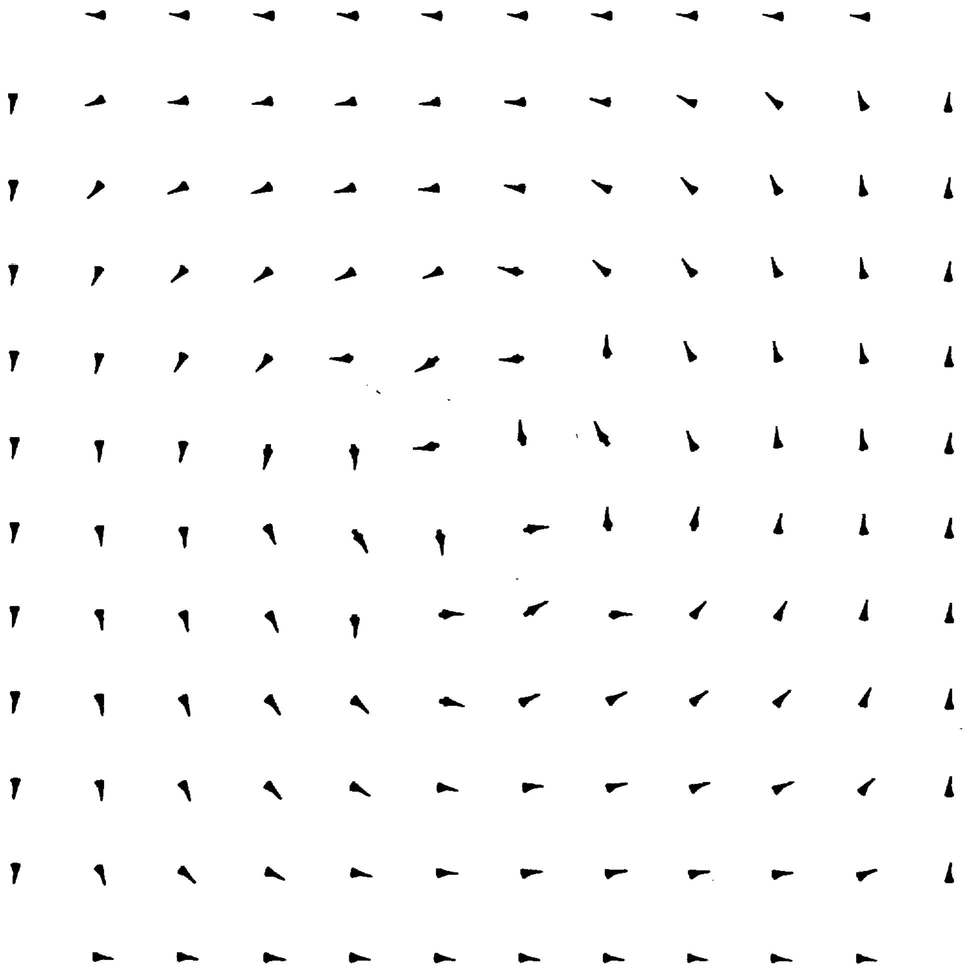
Oscillatory Flow



Time step 50

v2rt12.grid

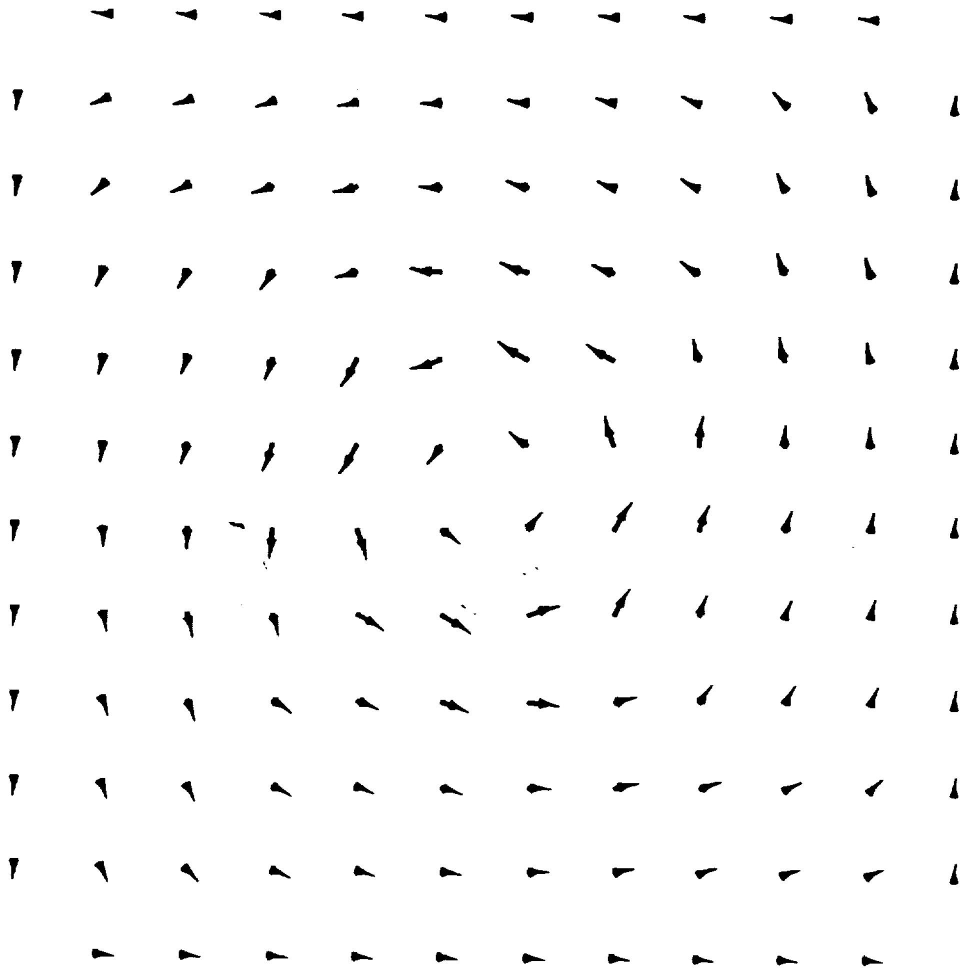
Oscillatory Flow



Time step 100

v2rt12.grid

Oscillatory Flow



Time step 150

Oscillatory Flow

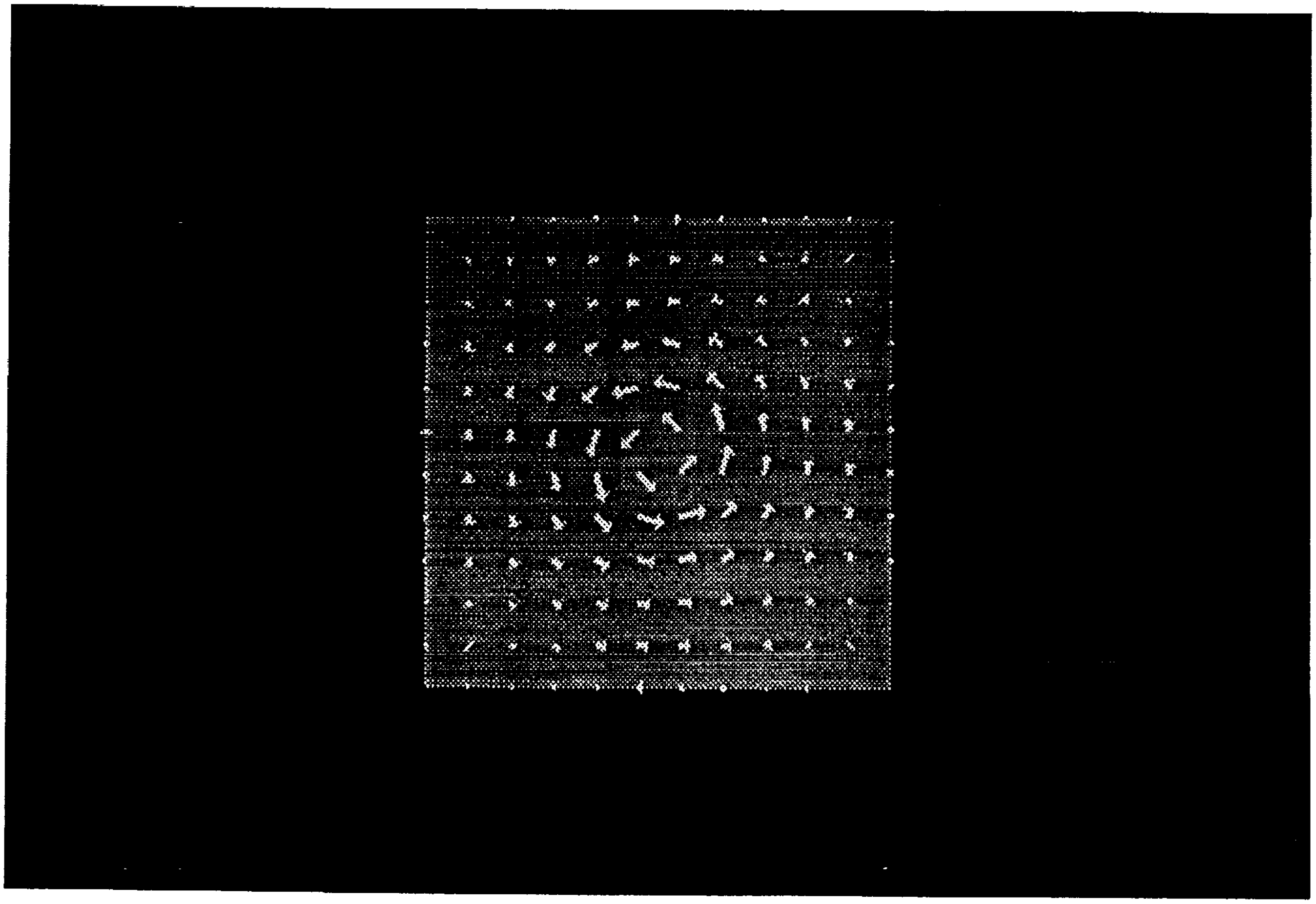
v2rt12.grid

Fig.5 Oscillatory Flow, velocity field at different time steps.

Re = 10000.

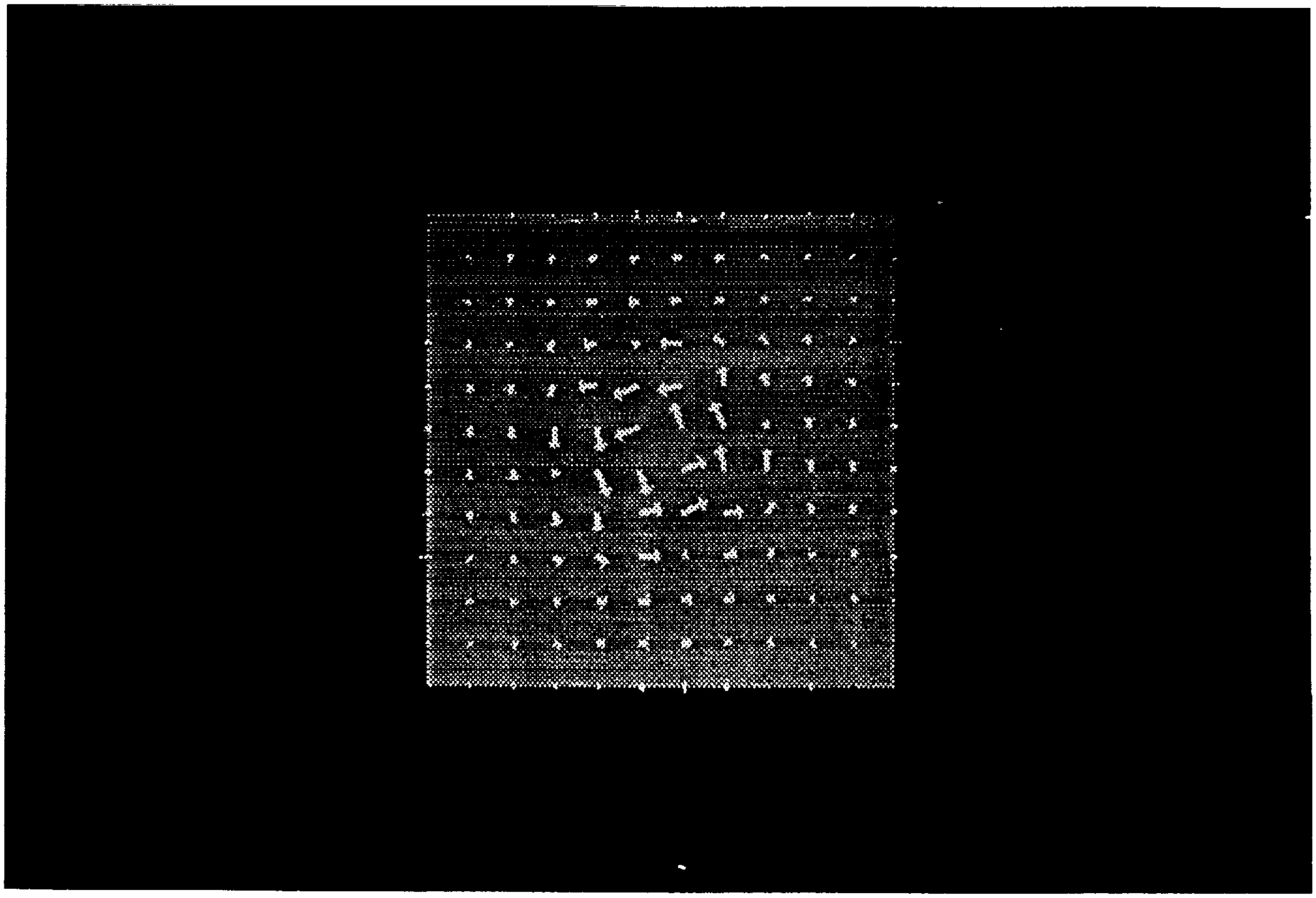
In Fig. 6 the same flow, but for **Re = 1000** is shown. The background, which is unfortunately given only in shades of gray, represents the intensity of the vorticity field. These pictures were transcribed from the color, motion graphics. The video was used to study in-phase, out-of-phase exctitation of the mixing motion.

v1rt12: time step 000

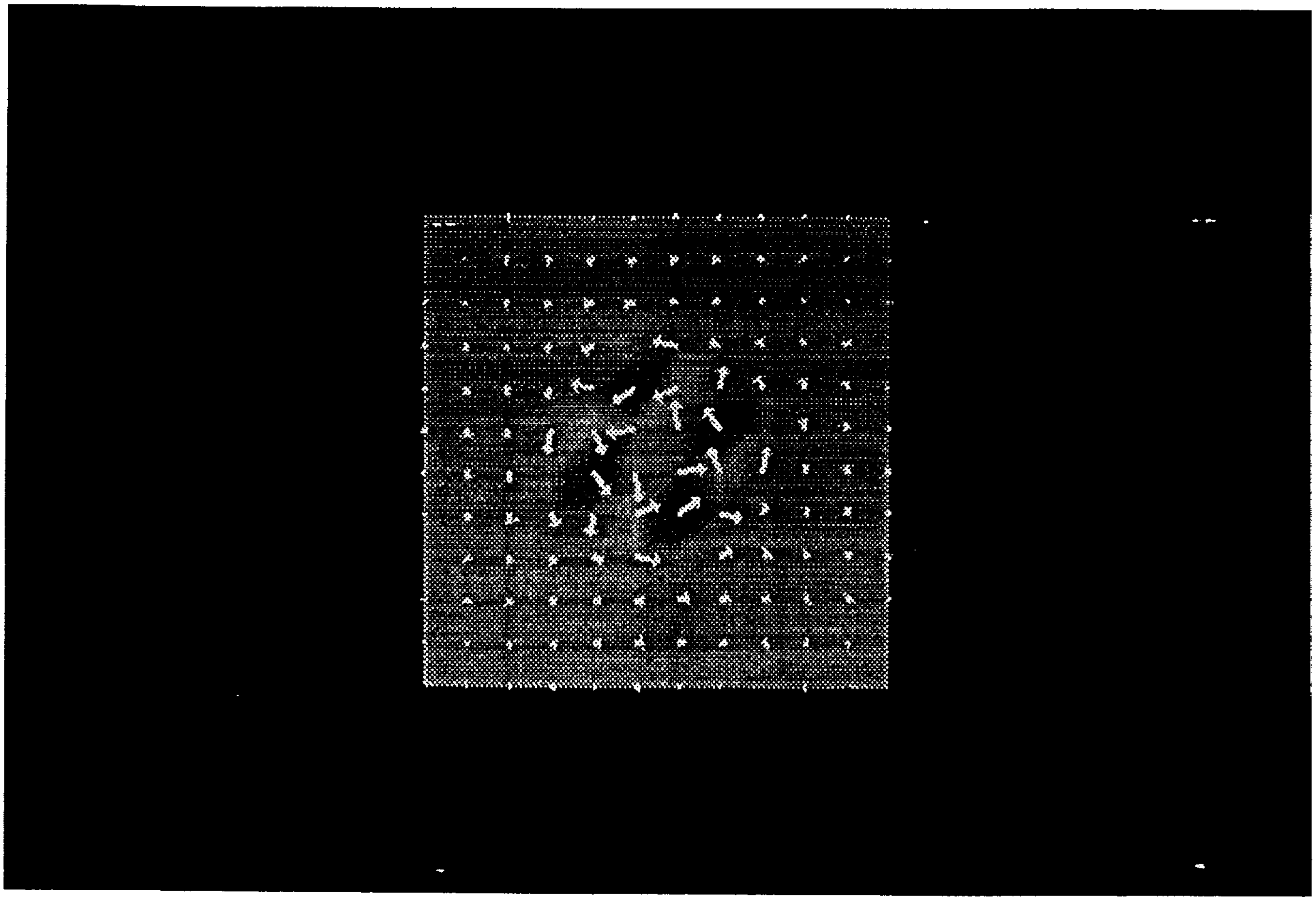


ORIGINAL PAGE IS
OF POOR QUALITY

v1rt12: time step 050



v1rt12: time step 100



ORIGINAL PAGE IS
OF POOR QUALITY

v1rt12: time step 150

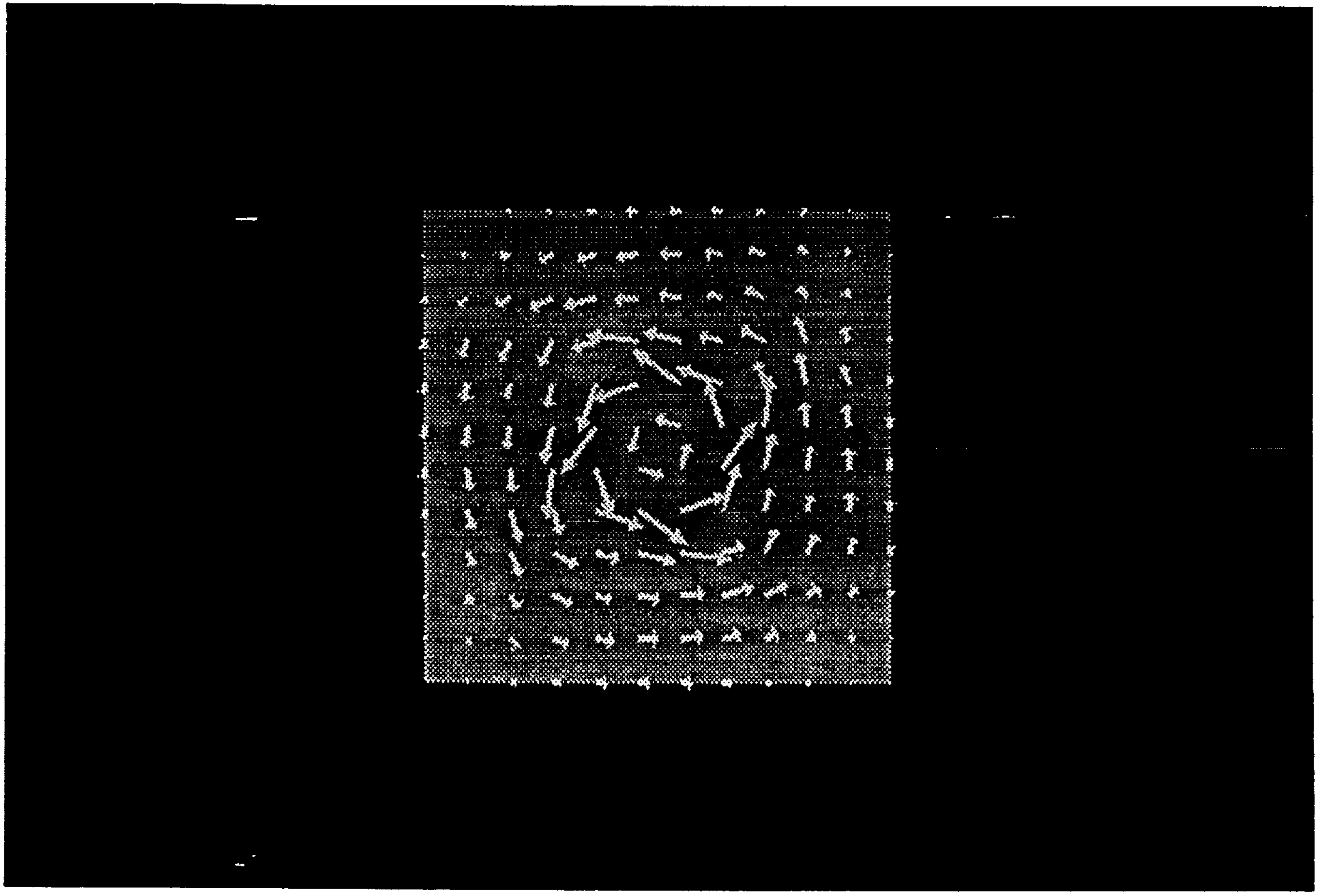


Fig.6 Oscillatory Flow, velocity field overlayed on the vorticity field, at different time steps.

Re = 1000.

It was observed that the numerical instability becomes important well into the chaotic regime of the mixing flow.

Discussion

The aim of the presented study was to show the practical advantages of the compact scheme. All cases were limited to two-dimensional phenomena, although there were no such constraint in the method of solution. The code used to obtain presented results is somewhat limited by the specific formulation of the fluid dynamic problem. The method is fully explaineded in [1], where , the potential advantages of *the compact scheme* are discussed. However, before this potential can be fully exploited, the computational efficiency has to be improved. The main issue is the characteristics of the matrix $\mathcal{A}$ of the system. There is a strong indication, [1], [6] and [10] that the compact scheme might benefit from the symmetric, positive-definite matrices, although this avenue was not pursued in this report. However, the fact that it is now a coupled system of equations, defined on staggered grids plus some other aspects of the system of algebraic equations offer significant adavantages, that would have to explored sometime in the future. At the present time, however, the scheme offers a viable alternative to other, more popular methods. Finally, a chaotic flow was obtained using only 144 Cartesian elements, this best illustrates the potential of the scheme .

Bibliography

[1] Rose, M. E. "Compact Finite Volume Methods for Partial Differential Equations -I. Boundary Value Problems," Journal of Scientific Computing, to appear.

[2] Peyret, R., and Taylor, T. D., "Computational Methods for Fluid Flow," Springer Verlag, New York, 1983.

[3] Kaczmarz, S., "Anganaherte Auflosung von Systemen linearer Gleichungen," Bull. Acad. Polon, Science et Lettres, A, 355, 1937.

[4] Gatski, T. B., Grosch, C. E., and Rose M. E., " The Numerical Solutions of Navier-Stokes Equations for Three-Dimensional, Unsteady, Incompressible Flows by Compact Schemes. J. Comp. Phys. (copy obtained from authors).

[5] Dongarra, J. J., Moler, C. B., Bunch, J. R., and Stewart, G. W., "Linpack User's Guide," SIAM, Philadelphia,1979

[6] Rose, M. E., "A Finite Difference Solutions of Stokes-Type Flows," (Preliminary Report), copy obtained from the author.

[7] Farouk, F., and Fusegi, T., "A Coupled Solution of the Vorticity-Velocity Formulation of the Incompressible Navier-Stokes Equations," International Journal for Numerical Methods in Fluids, Vol.5,1017-1034 (1985).

[8] Kreiss, H. O., "On Difference Approximations of the Dissipative Type for Hyperbolic Equations," Comm. Pure Appl. Math 17, (1964).

[9] Rose, M. E., "Finite Volume Methods for PDE," workshop on compact schemes conducted at NC A & T State University, Fall 1989.

[10] Rose, M. E.,"Compact Finite Methods for the Diffusion Equation," draft obtained from the author.