DELFT UNIVERSITY OF TECHNOLOGY

REPORT 92-91

An ILU smoother for the incompressible Navier-Stokes equations in general coordinates

S. Zeng and P. Wesseling

ISSN 0922-5641

Reports of the Faculty of Technical Mathematics and Informatics no. 92-91

Delft 1992

Abstract

ILU smoothers are good smoothers for linear multigrid methods. In this paper, a new ILU smoother for the incompressible Navier-Stokes equations, called CILU (Collective ILU), is designed, based on r-transformations. Existing ILU decompositions factorize the matrix with real elements. In CILU the elements of the matrix that is factorized are submatrices, corresponding to the set of physical variables. A multigrid algorithm using CILU as smoother is investigated. Average reduction factors and limiting reduction factors are measured to explore the performance of the algorithm. The results show that CILU is a good smoother.

1 Introduction

Theoretical and practical investigations for about two decades have shown that multigrid methods are very suitable for solving large systems of algebraic equations resulting from discretization of partial differential equations. In this paper, we will present a multigrid method for the incompressible Navier-Stokes equations in general coordinates discretized on a staggered grid. A new smoother of ILU type, called CILU (Collective ILU), is introduced.

The main components in a multigrid algorithm are smoothing and coarse grid correction. The smoother should possess the smoothing property, and the coarse grid approximation should have the approximation property ([4]). In [16], the smoothing and approximation properties are studied for the incompressible Navier-Stokes equations discretized on a staggered grid in Cartesian coordinates. In general coordinates a theory is not available. Therefore, the performance of CILU is tested in numerical experiments.

Classical Jacobi or Gauß-Seidel iteration may be used for smoothing. These methods are simple to implement. However, they are not robust. They fail when the problem contains anisotropies. Examples of anisotropies are strong convection and large or small aspect ratio of grid cells, which occur often in discretizations using boundary-fitted coordinates. ILU decomposition for smoothing in multigrid methods has heen investigated by many authors; for a survey, see [12]. It is found that ILU smoothing is robust and efficient. This leads us to consider a smoother based on an ILU decomposition.

For reasons explained elsewhere ([19]), we use Galerkin coarse grid approximation. This implies that the nonlinear problem to be solved is linearized outside the multigrid algorithm.

Discrete systems approximating the Navier-Stokes equations are indefinite. So direct implementation of ILU decompositions is problematic. This problem is overcome by applying an r-transformation, as proposed in [15],[17] and [18].

This paper is arranged as follows. In section 2, the partial differential equations and the discrete system that are to be solved are described. Section 3 explains briefly the r-transformation. An incomplete LU factorization called CILU is described in section 4. In section 5, a linear multigrid algorithm is presented which covers the V-, W-, F- and A-cycles. The choices for restriction and prolongation operators are given. Using skewed driven cavity problems and L-shaped driven cavity problems as test problems, in section 6 the performance of the linear multigrid using CILU as smoother is investigated.

2 Partial Differential Equations and Discretization

The tensor formulation of the incompressible Navier-Stokes equations in general coordinates reads as follows:

$$U^{\alpha}_{,\alpha} = 0, \tag{2.1}$$

$$\frac{\partial}{\partial t}(U^{\alpha}) + (U^{\alpha}U^{\beta})_{,\beta} + (g^{\alpha\beta}p)_{,\beta} - \tau^{\alpha\beta}_{,\beta} = B^{\alpha}, \tag{2.2}$$

where $\tau^{\alpha\beta}$ is the deviatoric stress tensor and is given by

$$\tau^{\alpha\beta} = Re^{-1}(g^{\alpha\gamma}U^{\beta}_{,\alpha} + g^{\gamma\beta}U^{\alpha}_{,\gamma}), \tag{2.3}$$

with Re the Reynolds number, p the pressure, t the time, $U^\alpha, \alpha = 1, 2, \ldots, nd$ the contravariant components of velocity with nd the number of space dimensions, and B^α the contravariant component of the body force. U^α and B^α are derived from their physical counterparts $\mathbf{u}$ and $\mathbf{b}$ through the contravariant base vectors $\mathbf{a}^\alpha$ of the general coordinates by

$$U^\alpha = \mathbf{a}^\alpha \cdot \mathbf{u}, \; B^\alpha = \mathbf{a}^\alpha \cdot \mathbf{b}. \tag{2.4}$$

Furthermore, $g^{\alpha\beta}$ is the metric tensor given by $g^{\alpha\beta} = \mathbf{a}^\alpha \cdot \mathbf{a}^\beta$. For better accuracy, the variable $V^\alpha = \sqrt{g}U^\alpha$ is used instead of U^α, where $\sqrt{g}$ is the Jacobian of the mapping; this is motivated in [5], [9] and [14].

The discrete system of the above equations discretized in general coordinates on a staggered grid in two dimensions (cf. figure 2.1) by using the finite volume method ([5],[6],[14],[9])

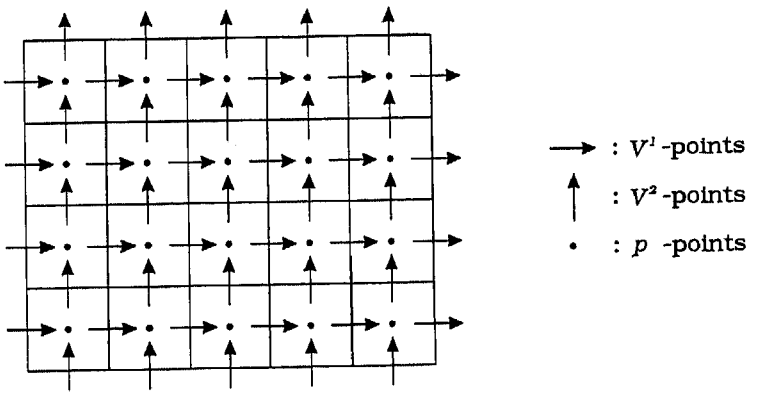


Figure 2.1: A staggered grid

can be written, for a given time interval Δt, as:

$$\begin{aligned} \frac{1}{\Delta t}\mathbf{V}^{n+1} + \theta\mathbf{Q}'(\mathbf{V}^{n+1}) + \theta\mathbf{G}\mathbf{p}^{n+1} &= \mathbf{f}'_v, \\ \mathbf{D}\mathbf{V}^{n+1} &= \mathbf{f}_c, \end{aligned} \tag{2.5}$$

with

$$\mathbf{f}'_v = \theta\mathbf{B}^{n+1} + (1-\theta)\mathbf{B}^n + \frac{1}{\Delta t}\mathbf{V}^n - (1-\theta)\mathbf{Q}'(\mathbf{V}^n) - (1-\theta)\mathbf{G}\mathbf{p}^n. \tag{2.6}$$

Here $\mathbf{V} = (V^1, V^2)$, $\mathbf{B} = (B^1, B^2)$ and $\mathbf{p}$ denote the discrete velocity, right-hand side and pressure grid functions. The superscript n indicates the time level. The parameter θ is in [0,1], and is taken to be 1 in the numerical experiments here, which gives the backward Euler method. The underlying ordering of the unkowns is

$$V_1^1, V_2^1, \cdots, V_{n_1}^1, \; V_1^2, V_2^2, \cdots, V_{n_2}^2, \; p_1, p_2, \cdots, p_{n_3}, \tag{2.7}$$

with some ordering (for example lexicographic) of the grid points. This will be called the block-wise ordering.

Equation (2.5) gives rise to a sequence of systems of equations for a sequence of time levels. It is linearized with the Newton's method, for example

$$(U^\alpha U^\beta)^{n+1} = (U^\alpha)^{n+1}(U^\beta)^n + (U^\alpha)^n(U^\beta)^{n+1} - (U^\alpha U^\beta)^n \tag{2.8}$$

This gives $\mathbf{Q}'(\mathbf{V}^{n+1}) = \mathbf{Q}_1\mathbf{V}^{n+1} + \mathbf{Q}_2(\mathbf{V}^n)$ with $\mathbf{Q}_1$ linear. Note that both $\mathbf{Q}_1$ and $\mathbf{Q}_2$ are evaluated by using $\mathbf{V}^n$. The resulting system is denoted by

$$\mathbf{K}\mathbf{x} = \mathbf{f} \tag{2.9}$$

with

$$\mathbf{K} = \begin{pmatrix} \mathbf{Q} & \mathbf{G} \\ \mathbf{D} & 0 \end{pmatrix}, \ \mathbf{x} = \begin{pmatrix} \mathbf{V}^{n+1} \\ \mathbf{p}^{n+1} \end{pmatrix}, \ \mathbf{f} = \begin{pmatrix} \mathbf{f}_v \\ \mathbf{f}_c \end{pmatrix}, \tag{2.10}$$

where

$$\mathbf{Q} = \frac{1}{\Delta t} + \mathbf{Q}_1, \ \mathbf{f}_v = \mathbf{f}_v' - \mathbf{Q}_2(\mathbf{V}^n). \tag{2.11}$$

If there exists a stationary solution, then it satisfies

$$\mathbf{K}_s\mathbf{x} = \mathbf{f}_s \tag{2.12}$$

with

$$\mathbf{K}_s = \begin{pmatrix} \mathbf{Q}' & \mathbf{G} \\ \mathbf{D} & 0 \end{pmatrix}, \ \mathbf{f}_s = \begin{pmatrix} \mathbf{B} \\ \mathbf{f}_c \end{pmatrix}. \tag{2.13}$$

3 The r-Transformation

3.1 Iteration with r-Transformation

A classical iteration method solving (2.9) is given by

$$\mathbf{x}^{i+1} = \mathbf{x}^i - \mathbf{M}^{-1}(\mathbf{K}\mathbf{x}^i - \mathbf{f}) \tag{3.1}$$

with $\mathbf{M}$ a splitting of $\mathbf{K}$:

$$\mathbf{K} = \mathbf{M} - \mathbf{N}. \tag{3.2}$$

This method converges if the splitting is what is called regular [10]. The zero block in $\mathbf{K}$ makes a regular splitting impossible. A remedy is to introduce a matrix $\bar{\mathbf{K}}$, such that a regular splitting

$$\mathbf{K}\bar{\mathbf{K}} = \mathbf{M} - \mathbf{N} \tag{3.3}$$

is easy to find. This implies a splitting

$$\mathbf{K} = \mathbf{M}\bar{\mathbf{K}}^{-1} - \mathbf{N}\bar{\mathbf{K}}^{-1} \tag{3.4}$$

resulting in the following iterative method:

$$\mathbf{x}^{i+1} = \mathbf{x}^i - \bar{\mathbf{K}}\mathbf{M}^{-1}(\mathbf{K}\mathbf{x}^i - \mathbf{f}). \tag{3.5}$$

The transformation $\bar{\mathbf{K}}$ has got several names in the literature. In [1], $\bar{\mathbf{K}}$ is called distributive operator and iteration (3.5) is called distributive iteration; in [15], $\bar{\mathbf{K}}$ is called r-transformation and the iteration is called transforming iteration. Here we adopt the latter terminology. Many iterative methods can be fitted into the frame work of (3.3) and (3.5), such as the SIMPLE method of Patankar and Spalding [8] and its variants, and the DGS method of Brandt and Dinar [2] ([4],[15],[16],[17],[18]).

In practice, it may be convenient to replace $\bar{\mathbf{K}}$ by an approximation $\tilde{\mathbf{K}}$. Consequently, the iteration procedure (3.5) becomes

$$\mathbf{x}^{i+1} = \mathbf{x}^i - \tilde{\mathbf{K}}\mathbf{M}^{-1}(\mathbf{K}\mathbf{x}^i - \mathbf{f}). \tag{3.6}$$

Obviously, (3.6) converges to the solution, if it converges. Convergence may be enhanced by underrelaxation:

$$\mathbf{x}^{i+1} = \mathbf{x}^i - \omega\tilde{\mathbf{K}}\mathbf{M}^{-1}(\mathbf{K}\mathbf{x}^i - \mathbf{f}). \tag{3.7}$$

Method (3.7) is the smoothing iteration method that we use.

3.2 Construction of r-Transformation

A theory of constructing smoothers with r-transformation is given in [16]. Some applications to the Stokes and the Navier-Stokes equations can be found in [15]. We summarize some results. For use as smoother in a multigrid method, an iterative method must have the smoothing property, introduced in [4]; see [13] for an elementary introduction. In [16], it is shown generally that if $\mathbf{K}\bar{\mathbf{K}}$ is of the following block-triangular form:

$$\mathbf{K}\bar{\mathbf{K}} = \begin{pmatrix} \mathbf{A} & 0 \\ \mathbf{B} & \mathbf{C} \end{pmatrix} \tag{3.8}$$

and can be splitted regularly into $\mathbf{M} - \mathbf{N}$, then the smoothing property holds for system (3.8) if the iterative matrix

$$\mathbf{S} = \mathbf{M}^{-1}\mathbf{N} \tag{3.9}$$

with

$$\mathbf{S} = \begin{pmatrix} \mathbf{S}_{11} & 0 \\ \mathbf{S}_{21} & \mathbf{S}_{22} \end{pmatrix} \tag{3.10}$$

has the the smoothing property for its diagonal blocks. Hence, the study of the smoothing property for systems is essentially reduced to the study of of the smoothing property for single equations. Furthermore, [16] gives conditions under which the smoothing property for the perturbed method (3.6) follows from the smoothing property for (3.5). Hence, it is attractive to choose $\bar{\mathbf{K}}$ such that $\mathbf{K}\bar{\mathbf{K}}$ has the block triangular form (3.8). A possible choice for $\bar{\mathbf{K}}$ is

$$\bar{\mathbf{K}} = \begin{pmatrix} \mathbf{I} & \mathbf{K}_{12} \\ 0 & \mathbf{K}_{22} \end{pmatrix}. \tag{3.11}$$

Then we have

$$\mathbf{K}\bar{\mathbf{K}} = \begin{pmatrix} \mathbf{Q} & \mathbf{Q}\mathbf{K}_{12} + \mathbf{G}\mathbf{K}_{22} \\ \mathbf{D} & \mathbf{D}\mathbf{K}_{12} \end{pmatrix}. \tag{3.12}$$

Choosing $\mathbf{K}_{12}$ and $\mathbf{K}_{22}$ such that $\mathbf{QK}_{12} + \mathbf{GK}_{22} = 0$ results in the form given in (3.8).

There are many possibilities for choosing $\bar{\mathbf{K}}$. Wittum's theory gives us a guide. In (3.12), we do not have problems in constructing a smoother for $\mathbf{Q}$, provided that the discretization is appropriate ($\mathbf{Q}$ should be an M-matrix). What we should do then is to choose $\bar{\mathbf{K}}$ such that the smoothing property exists also for the block $\mathbf{DK}_{12}$.

Choosing

$$\bar{\mathbf{K}} = \begin{pmatrix} \mathbf{I} & -\mathbf{Q}^{-1}\mathbf{G}\mathbf{E}^{-1}\mathbf{F} \\ 0 & \mathbf{E}^{-1}\mathbf{F} \end{pmatrix} \tag{3.13}$$

with $\mathbf{E} = \mathbf{DQ}^{-1}\mathbf{G}$ results in

$$\mathbf{K}\bar{\mathbf{K}} = \begin{pmatrix} \mathbf{Q} & 0 \\ \mathbf{D} & -\mathbf{F} \end{pmatrix} \tag{3.14}$$

where $\mathbf{F}$ is still to be chosen. We discretize (2.1) and (2.2) with central differences. As a consequence, $\mathbf{Q}$ is not an M-matrix for Re sufficiently large (approximately $Re > 2/h$ in our examples with h the local mesh-size). If we choose $-\mathbf{F}$ to be an M-matrix then it will be easy (for Re small enough) to obtain a smoother for the product system $\mathbf{K}\bar{\mathbf{K}}$, as discussed before. The first choice is $\mathbf{F}$=$\mathbf{DG}$ corresponding to the distributed Gauß-Seidel method of [2]. Because of the occurrence of $\mathbf{Q}^{-1}$ and $\mathbf{E}^{-1}$ this $\bar{\mathbf{K}}$ is not practical, and is approximated by

$$\tilde{\mathbf{K}} = \begin{pmatrix} \mathbf{I} & -\mathbf{G} \\ 0 & \mathbf{DG} \end{pmatrix}. \tag{3.15}$$

The second choice is $\mathbf{F} = \mathbf{E}$, giving

$$\bar{\mathbf{K}} = \begin{pmatrix} \mathbf{I} & -\mathbf{Q}^{-1}\mathbf{G} \\ 0 & \mathbf{I} \end{pmatrix} \tag{3.16}$$

leading to iterative methods of the so-called SIMPLE type [8]. This gives

$$\mathbf{K}\bar{\mathbf{K}} = \begin{pmatrix} \mathbf{Q} & 0 \\ \mathbf{D} & -\mathbf{DQ}^{-1}\mathbf{G} \end{pmatrix}. \tag{3.17}$$

For practical purposes $\mathbf{Q}^{-1}$ in (3.16) and (3.17) has to be approximated further.

4 CILU Decomposition

4.1 Incomplete decomposition

The residual amplification matrix of (3.6) is $\mathbf{I} - \mathbf{K}\tilde{\mathbf{K}}\mathbf{M}^{-1}$. Hence $\mathbf{M}$ should be close to $\mathbf{K}\tilde{\mathbf{K}}$ but easily invertible. With incomplete decomposition one chooses

$$\mathbf{M} = (\mathbf{L} + \mathbf{D})\mathbf{D}^{-1}(\mathbf{D} + \mathbf{U}) \tag{4.1}$$

with $\mathbf{L}$ and $\mathbf{U}$ strictly lower and upper triangular matrices and $\mathbf{D}$ a diagonal matrix. A possible choice for $\mathbf{L}$, $\mathbf{D}$ and $\mathbf{U}$ is as follows. Let G be a non-zero pattern, and let $\mathbf{L}$, $\mathbf{D}$, $\mathbf{U} \neq 0$ only on G. Then we require

$$\mathbf{M}_{ij} = (\mathbf{K}\tilde{\mathbf{K}})_{ij}, \quad (i,j) \in G \tag{4.2}$$

from which $\mathbf{L}$, $\mathbf{D}$ and $\mathbf{U}$ follow. It is known that the rate of convergence of the resulting iterative method depends on the ordering of the unknowns. We will number the cells in lexicographic order. The V^1-unknown in the left cell face, V^2 in the lower face and p in the center are grouped together in a 3-vector $\mathbf{u}_i = (V^1, V^2, p)_i$ with i the number of the cell. Because of this collective treatment of the three unknowns, we call the resulting method collective ILU decomposition (CILU). This collective ordering induces a 3×3 block matrix representation of $\mathbf{K}\tilde{\mathbf{K}}$. Such block matrices are called cell blocks. A typical row, say number i, has non-zero elements at postions $(i, i\pm I\pm 1, i\pm I, i\mp I\pm 1, i\pm 1, i+I-2, i-2, i-2I, i-2I+1)$, where I is the number of cells in the ξ^1-direction. The structure of the stencil of $\mathbf{K}\tilde{\mathbf{K}}$ is given by

$$[\mathbf{K}\tilde{\mathbf{K}}] = \begin{bmatrix} * & * & * & * & & & & & \\ * & * & \underline{*} & * & & & & & \\ & * & * & * & & & & & \\ & & * & * & & & & & \end{bmatrix}. \tag{4.3}$$

Here each $*$ represents a 3×3 matrix. The $*$ with an undercore corresponds to cell number i. We choose $G = (i, i \pm I \pm 1, i \pm I, i \mp I \pm 1, i \pm 1)$ and introduce the following abbreviations:

$$\begin{aligned} z_i &= \mathbf{H}_{i,i-I-1}, \quad a_i = \mathbf{H}_{i,i-I}, \quad b_i = \mathbf{H}_{i,i-I+1}, \\ c_i &= \mathbf{H}_{i,i-1}, \quad d_i = \mathbf{H}_{i,i}, \quad q_i = \mathbf{H}_{i,i+1}, \\ f_i &= \mathbf{H}_{i,i+I-1}, \quad g_i = \mathbf{H}_{i,i+I}, \quad p_i = \mathbf{H}_{i,i+I+1}, \end{aligned} \tag{4.4}$$

where $\mathbf{H} = \mathbf{K}\tilde{\mathbf{K}}$. The non-zero elements of $\mathbf{L}$, $\mathbf{D}$ and $\mathbf{U}$ (which are also 3×3 matrices) in the location of z_i, $a_i, \cdots, p_i$ are called $\omega_i, \alpha_i, \beta_i, \gamma_i, \delta_i, \mu_i, \zeta_i, \eta_i$ and τ_i, respectively. Equation (4.2) leads to the following recursion:

$$\begin{aligned} \omega_i &= z_i, \\ \alpha_i &= a_i - \omega_i \delta_{i-I-1}^{-1} \mu_{i-I-1}, \\ \beta_i &= b_i - \alpha_i \delta_{i-I}^{-1} \mu_{i-I}, \\ \gamma_i &= c_i - \omega_i \delta_{i-I-1}^{-1} \eta_{i-I-1} - \alpha_i \delta_{i-I}^{-1} \zeta_{i-I}, \\ \delta_i &= d_i - \omega_i \delta_{i-I-1}^{-1} \tau_{i-I-1} - \alpha_i \delta_{i-I}^{-1} \eta_{i-I} - \beta_i \delta_{i-I+1}^{-1} \zeta_{i-I+1} - \gamma_i \delta_{i-1}^{-1} \mu_{i-1}, \\ \mu_i &= q_i - \alpha_i \delta_{i-I}^{-1} \tau_{i-I} - \beta_i \delta_{i-I+1} \eta_{i-I+1}, \\ \zeta_i &= f_i - \gamma_i \delta_{i-I}^{-1} \eta_{i-I}, \\ \eta_i &= g_i - \gamma_i \delta_{i-1}^{-1} \tau_{i-1}, \\ \tau_i &= p_i. \end{aligned} \tag{4.5}$$

All terms with indices pointing outside the grid are defined to be zero.

4.2 Approximation of $\mathbf{K}\tilde{\mathbf{K}}$

Temporarily using the blockwise ordering (2.7), we choose the following distribution operator (cf. (3.16))

$$\bar{\mathbf{K}} = \begin{pmatrix} \mathbf{I} & -\mathbf{Q}^{-1}\mathbf{G} \\ 0 & \zeta\mathbf{I} \end{pmatrix}, \tag{4.6}$$

where the parameter ζ will be used to enhance multigrid convergence. This gives

$$\mathbf{K}\tilde{\mathbf{K}} = \begin{pmatrix} \mathbf{Q} & (\zeta-1)\mathbf{G} \\ \mathbf{D} & -\mathbf{D}\mathbf{Q}^{-1}\mathbf{G} \end{pmatrix}. \tag{4.7}$$

Because $\mathbf{Q}^{-1}$ is not readily available we approximate $\mathbf{Q}$ by $\tilde{\mathbf{Q}} = diag(\mathbf{Q})$. This gives

$$\tilde{\mathbf{K}} = \begin{pmatrix} \mathbf{I} & -\tilde{\mathbf{Q}}^{-1}\mathbf{G} \\ 0 & \zeta\mathbf{I} \end{pmatrix}, \quad \mathbf{K}\tilde{\mathbf{K}} = \begin{pmatrix} \mathbf{Q} & -\mathbf{Q}\tilde{\mathbf{Q}}^{-1}\mathbf{G} + \zeta\mathbf{G} \\ \mathbf{D} & -\mathbf{D}\tilde{\mathbf{Q}}^{-1}\mathbf{G} \end{pmatrix}. \tag{4.8}$$

CILU smoothing can now be summarized as

$$\mathbf{x}^{i+1} = \mathbf{x}^i - \omega\tilde{\mathbf{K}}\mathbf{M}^{-1}(\mathbf{K}\mathbf{x}^i - \mathbf{f}), \tag{4.9}$$

where the matrix $\mathbf{M}$ will be the $(\mathbf{L}+\mathbf{D})\mathbf{D}^{-1}(\mathbf{D}+\mathbf{U})$ decomposition of the following approximation $\widetilde{\mathbf{K}\tilde{\mathbf{K}}}$ of $\mathbf{K}\tilde{\mathbf{K}}$, using the collective ordering:

$$\widetilde{\mathbf{K}\tilde{\mathbf{K}}} = \begin{pmatrix} \mathbf{Q} & (\zeta-1)\mathbf{G} \\ \mathbf{D} & -\mathbf{D}\tilde{\mathbf{Q}}^{-1}\mathbf{G} \end{pmatrix}. \tag{4.10}$$

Based on numerical experiments we have found $\omega = 0.7$ to be a suitable choice. The choice of ζ will be discussed later.

5 The Multigrid Algorithm

5.1 The Linear Multigrid Algorithm

Adaptd from [11] and [13], the structure diagram of the linear multigrid algorithm which includes the V-, W-, F- and A-(adaptive) cycle is given in figure 5.1. The linear system

$$L^{l_f}\varphi^{l_f} = f^{l_f} \tag{5.1}$$

is to be solved, with l_f the finest grid index; l is the grid index, *nmg*, *nsc*, *npre*, *npost* are the number of multigrid iterations, the number of iterations on the coarsest grid, the number of pre-smoothings and the number of post-smoothings, respectively; *cycle* chooses a multigrid strategy from the V-, W-, F- and A- cycles; *tolf* is the accuracy tolerance factor: if the residual norm on the finest grid is smaller than the product of this factor and the norm of the right-hand side, the multigrid iteration terminates. The parameter *maxgam* controls the number of visits to a grid, which is useful when the A-cycle is used. If the number of visits to a grid coming from the next coarser grid exceeds *maxgam*, then the next finer grid has to be visited. The parameters δ, η and *tolc* have effect only when the A-cycle is used. δ is called the residual norm tolerance factor: when the residual norm on a grid is smaller than the product of δ and the residual norm on the finer grid, a coarse grid correction takes place to the finer grid, otherwise a restriction is done. The parameter η is referred to as the smoothing rate tolerance. When the smoothing factor, defined later, is larger than η, then smoothing stops.

$\mathbf{P}^l$ stands for prolongation of corrections from grid $l-1$ to l, and $\mathbf{R}^l$ represents restriction of the residual from grid $l+1$ to l. In order to avoid redundant computations, variables *llast* and *lnrm* are introduced; *llast* prohibits redundant smoothing, when the A-cycle is employed and η is exceeded; *lnrm* provents the residual norm from being computed if computation of the residual norm just took place on the same grid. The smoothing algorithm SA should take the structure given in figure 5.2. Here $S^l(\varphi^l, f^l)$ represents one smoothing. The coarse grid solver SAC can be the same as SA, solving the system on the coarsest grid in *nsc* iterations, or can use other iterative or direct solution methods.

To investigate the multigrid algorithm with CILU as smoother, we design the following tests. Starting from the finest grid from the zero solution, 2 time steps, each accompanied by two (multigrid) interations, are performed first to give an initial start for the solution. Then 20 multigrid iterations are carried out, in which we measure the average reduction factor and the limiting reduction factor, which are to be explained later. The multigrid cycle uses the W-cycle, with one pre- and one post-smoothing. The coarsest grid is fixed at 2×2. For transfer operators, we distinguish between the prolongation operators for the computation of coarse grid matrices and those for the computation of coarse grid correction. In the formulation of the coarse grid matrices by means of Galerkin coarse grid approximation, two versions will be used for the prolongation of V^α. Version 1 is so-called hybrid interpolation (explained in [19]), which is a mixture of piecewise constant and bilinear interpolation. Version 2 is bilinear interpolation. Piecewise constant interpolation is used for p. The prolongation operator in the computation of coarse grid correction is of version 2. The restriction operator is the adjoint of the version 1 prolongation operator. See [19] and [20] for the details about the choice of prolongation and restriction operators and an efficient formulation of Galerkin coarse grid approximation for systems of equations.

Figure 5.1: Linear MG, including the V-, W-, F- and A-cycle

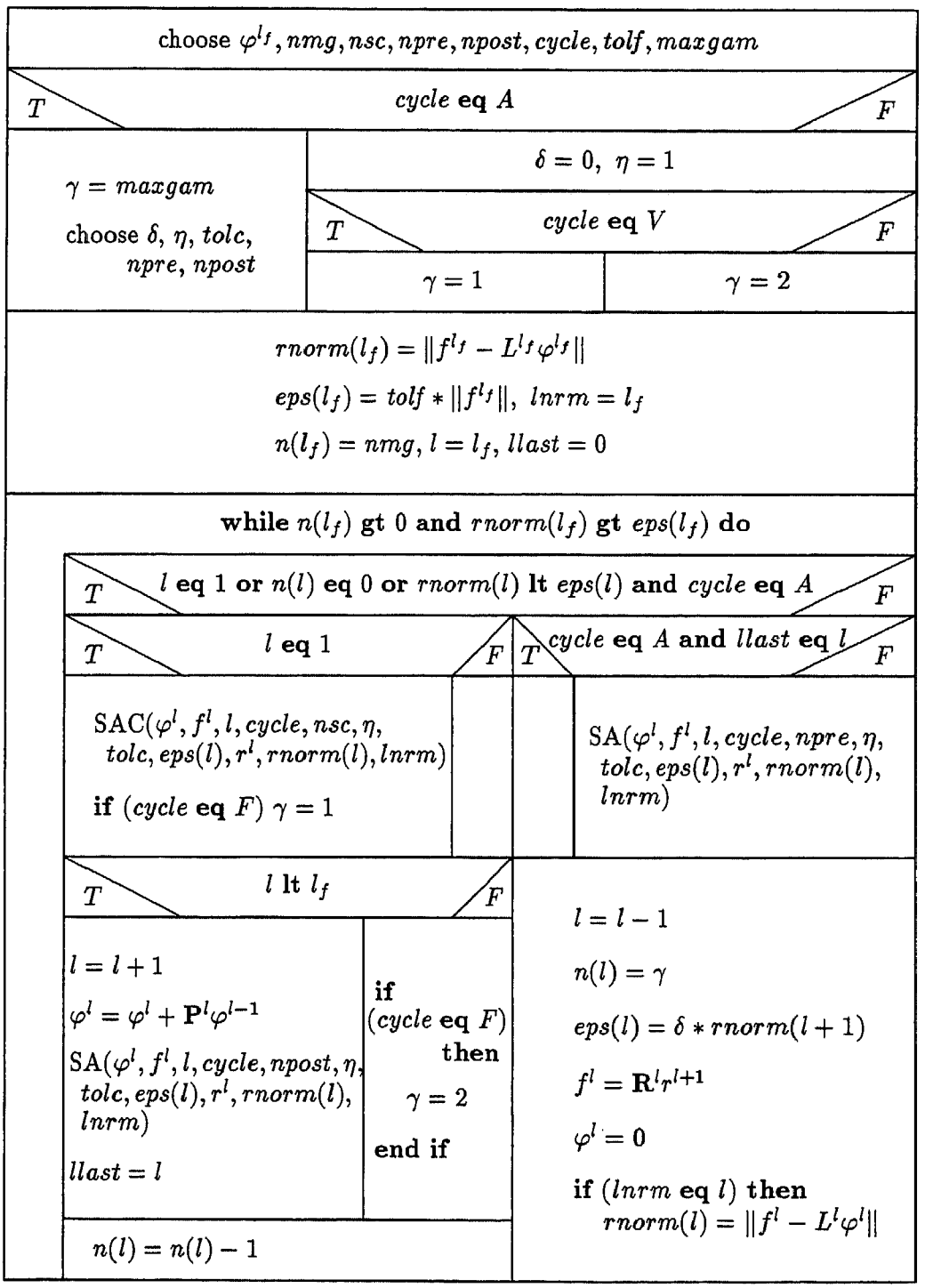
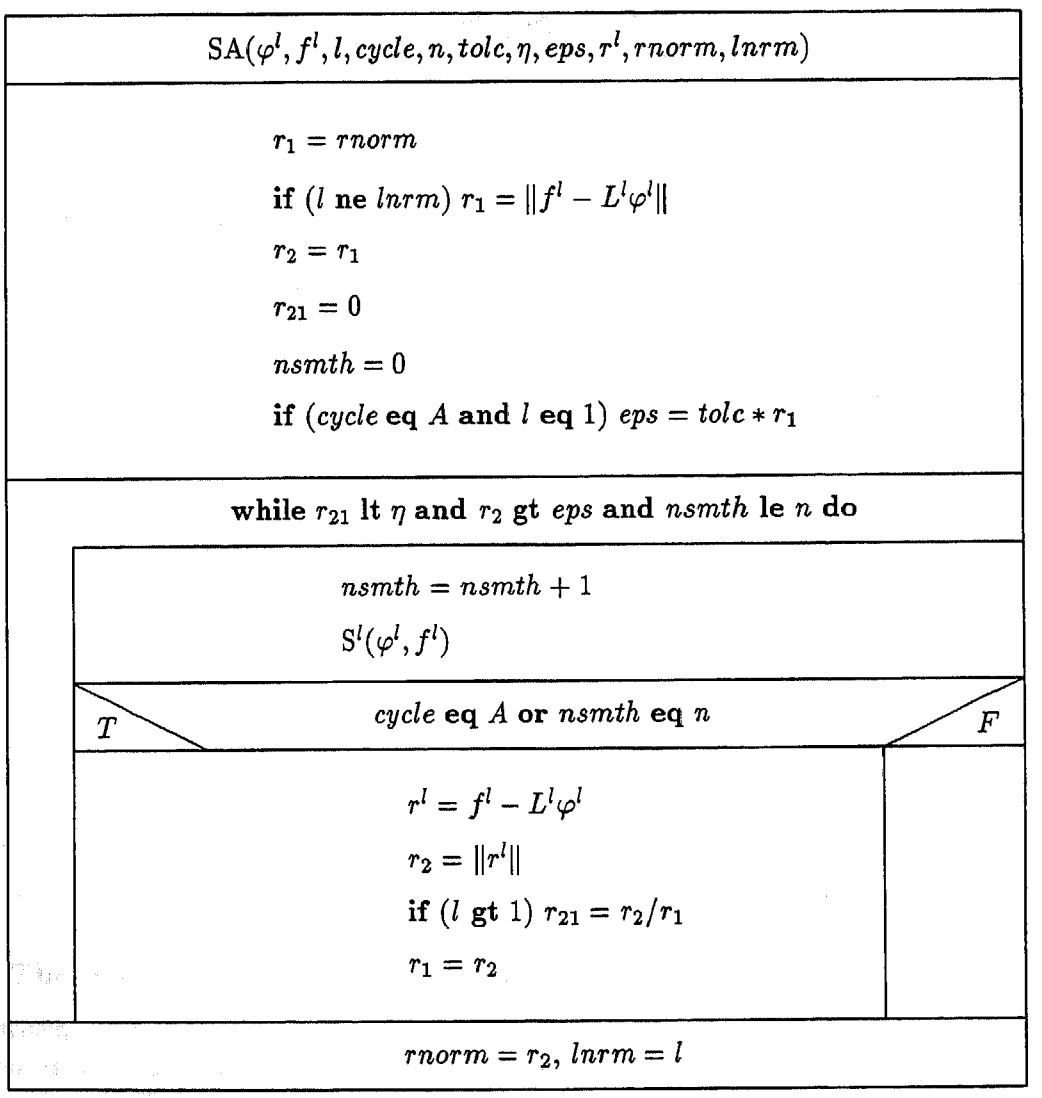


Figure 5.2: The structure of the smoother SA



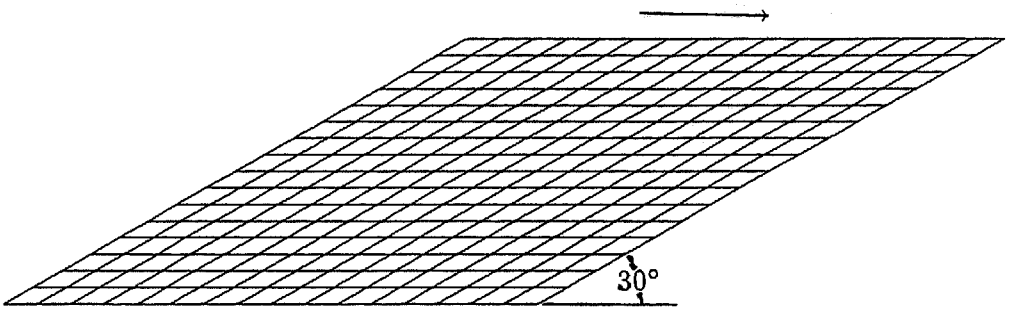


Figure 6.1: The skewed driven cavity problem and the grid used

6 Test Problems and Results

Let $\mathbf{r} = \mathbf{f} - \mathbf{K}\mathbf{x}$ be the residual of equation (2.9), and let $r = \|\mathbf{r}\|$ with $\|\cdot\|$ the l_2-norm. After linearization, a number of multigrid iterations is carried out, after which V^α and p are updated outside of multigrid iterations. Let r_0 be the initial residual norm on the finest grid, and r_n be the residual norm on the finest grid after n multigrid iterations. The average reduction factor $\bar{\rho}_n$ is defined by

$$\bar{\rho}_n = \left(\frac{r_n}{r_0}\right)^{\frac{1}{n}}. \tag{6.1}$$

The reduction factor at the i-th iteration is defined by

$$\rho_i = \frac{r_i}{r_{i-1}}. \tag{6.2}$$

If ρ_i has a limit for i tending to infinity, then it is the asymptotic reduction factor. Let $\mathbf{r}_s = \mathbf{f}_s - \mathbf{K}_s\mathbf{x}$ be the residual of equation (2.12) and $r_s = \|\mathbf{r}_s\|$. A steady state is reached if

$$\frac{r_s^t}{r_s^0} \leq \epsilon \ll 1 \tag{6.3}$$

is satisfied, with r_s^0 being r_s at the initial time level and r_s^t being r_s at time level t.

6.1 The Skewed Driven Cavity Problem

The driven cavity problem is chosen first (cf. figure 6.1). A bench-mark solution for this problem is available recently in [3], where the grid used is a collocated grid. This problem is also solved in [7] by a nonlinear multigrid method for the steady case on a staggered grid. Here we do not want to solve the differential equations very accurately, since our purpose is to investigate the performance of the multigrid algorithm with the CILU smoother. In accordance with [7], the Reynolds numbers will be 100 and 1000, respectively. The numbering of cells is lexicographic. Figures 6.2 and 6.3 give the streamlines obtained after 20 time steps and agree well with the solutions presented in [3] and [7]. The time step $\Delta t = 1$, and one

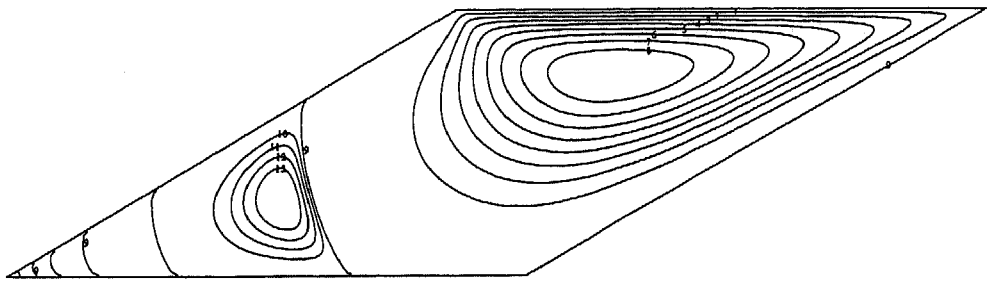


Figure 6.2: Streamlines for the skewed driven cavity problem, $Re = 100$, $\Delta t = 1$, 20 time steps, $r_s^t/r_s^0 < 3.570 \times 10^{-12}$, on 128×128 grid

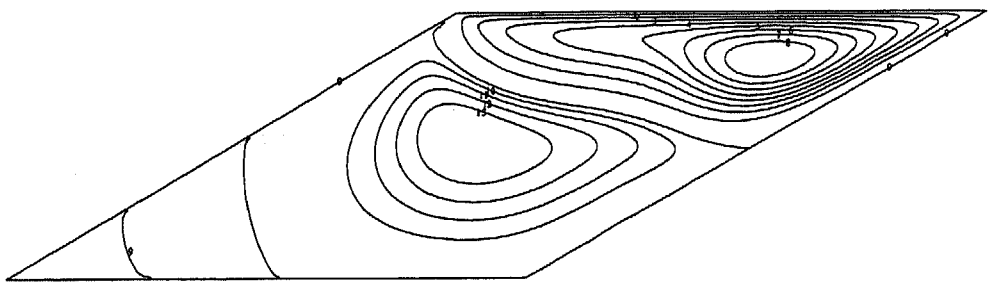


Figure 6.3: Streamlines for the skewed driven cavity problem, $Re = 1000$, $\Delta t = 1$, 20 time steps, $r_s^t/r_s^0 < 6.424 \times 10^{-5}$, on 128×128 grid

multigrid iteration is performed for each time step. Larger time steps are not used because it is found that with a larger time step, r_s^t decreases at a slower speed than with a smaller time step and therefore the termination time for achieving the same r_s^t is larger with a larger Δt. Furthermore, convergence problems arise after several time steps with larger time steps for high Reynolds number cases; Δt has to be taken sufficiently small in order to maintain the diagonal dominance and improve smoothing. This can be seen in the next test problem. Table 6.1 presents reduction factors for 5 successive iterations on various grids before rounding error takes effect, taking the solution obtained after 2 time steps with $\Delta t = 5$ as the initial solution for the linear multigrid iterations. If the effect of rounding error does not come into play, the reduction factors for the last five iterations are presented.

The dependence of multigrid convergence on ζ is given in table 6.2. The prolongation operators for **RAP** are of version 2. Apparently, ζ should not be too large. The rate of convergence is not very sensitive to ζ.

6.2 The L-Shaped Driven Cavity Problem

This problem is proposed in [7] and is illustrated in figure 6.4. In order to get rid of wiggles in the solution, a smooth grid generated by a bi-harmonic grid generator is used ([7]). The computational domain is depicted in figure 6.5. We find that the multigrid algorithm does

Table 6.1: Reduction factors for the skewed driven cavity problem, $\zeta = 2$

$Re = 100$, **P** for **RAP** version 1				
Grid	16×16	32×32	64×64	128×128
i	16	16	16	16
ρ_i	.3424	.5951	.7314	.8061
ρ_{i+1}	.3416	.5951	.7313	.8072
ρ_{i+2}	.3409	.5950	.7312	.8081
ρ_{i+3}	.3403	.5948	.7311	.8088
ρ_{i+4}	.3497	.5946	.7310	.8093
r_0	$.1209 \times 10^{-2}$	$.1680 \times 10^{-2}$	$.1398 \times 10^{-2}$	$.1308 \times 10^{-2}$
r_{i+4}	$.5165 \times 10^{-12}$	$.2729 \times 10^{-7}$	$.1175 \times 10^{-5}$	$.4784 \times 10^{-5}$
$\bar{\rho}$	.3400	.5761	.7018	.7554
$Re = 1000$, **P** for **RAP** version 1				
i	16	16	16	16
ρ_i	.4887	.4779	.5924	.6046
ρ_{i+1}	.4855	.4811	.5985	.5886
ρ_{i+2}	.4901	.4817	.6036	.6152
ρ_{i+3}	.5095	.4796	.6091	.6621
ρ_{i+4}	.5205	.4765	.6144	.6868
r_0	$.8282 \times 10^{-3}$	$.1261 \times 10^{-2}$	$.9889 \times 10^{-3}$	$.5778 \times 10^{-3}$
r_{i+4}	$.7269 \times 10^{-9}$	$.4235 \times 10^{-9}$	$.1384 \times 10^{-8}$	$.2706 \times 10^{-7}$
$\bar{\rho}$	.4979	.4746	.5097	.6074
$Re = 100$, **P** for **RAP** version 2				
i	16	16	16	16
ρ_i	.3466	.3504	.3213	.4825
ρ_{i+1}	.3481	.3447	.3506	.4829
ρ_{i+2}	.3493	.3409	.4266	.4832
ρ_{i+3}	.3505	.3396	.4573	.4833
ρ_{i+4}	.3515	.3407	.4381	.4832
r_0	$.9608 \times 10^{-3}$	$.1270 \times 10^{-2}$	$.8138 \times 10^{-3}$	$.5088 \times 10^{-3}$
r_{i+4}	$.8049 \times 10^{-13}$	$.1306 \times 10^{-12}$	$.2364 \times 10^{-11}$	$.5613 \times 10^{-11}$
$\bar{\rho}$	.3134	.3167	.3742	.4001
$Re = 1000$, **P** for **RAP** version 2				
i	16	16	16	16
ρ_i	.4735	.4733	.4105	.4396
ρ_{i+1}	.4839	.4724	.4156	.4541
ρ_{i+2}	.5059	.4721	.4087	.4643
ρ_{i+3}	.5191	.4713	.4146	.4706
ρ_{i+4}	.5206	.4717	.4118	.4741
r_0	$.8115 \times 10^{-3}$	$.1435 \times 10^{-2}$	$.1245 \times 10^{-2}$	$.5575 \times 10^{-3}$
r_{i+4}	$.6937 \times 10^{-9}$	$.2613 \times 10^{-9}$	$.2429 \times 10^{-10}$	$.2036 \times 10^{-11}$
$\bar{\rho}$	.4973	.4603	.4116	.3786

Table 6.2: The dependence of multigrid convergence on ζ in the skewed driven cavity problem, finest grid=128 x 128, **RAP** version 2

$Re = 100$, $r_0 = .5088 \times 10^{-3}$							
ζ, i	ρ_i	ρ_{i+1}	ρ_{i+2}	ρ_{i+3}	ρ_{i+4}	r_{i+4}	$\bar{\rho}$
1,16	.7707	.7719	.7726	.7730	.7731	$.6219 \times 10^{-7}$	.6373
1.5,16	.4116	.4127	.4138	.4148	.4157	$.1752 \times 10^{-11}$	.3774
2,16	.4825	.4829	.4832	.4833	.4832	$.5613 \times 10^{-11}$	.4001
4,16	.5712	.5729	.5728	.5745	.5908	$.2079 \times 10^{-9}$	.4793
5,16	.6841	.7250	.7467	.7330	.6975	$.6709 \times 10^{-8}$	.5702
6,*	div						
$Re = 1000$, $r_0 = .5575 \times 10^{-3}$							
1,*	div						
1.5,16	.4097	.4108	.4118	.4127	.4136	$.8955 \times 10^{-12}$	.3633
2,16	.4396	.4541	.4643	.4706	.4741	$.2036 \times 10^{-11}$	.3786
4,16	.4619	.4628	.4648	.4715	.4812	$.6364 \times 10^{-10}$	.4497
5,16	.6322	.6711	.6851	.6760	.6490	$.8905 \times 10^{-9}$	.5131
6,*	div						

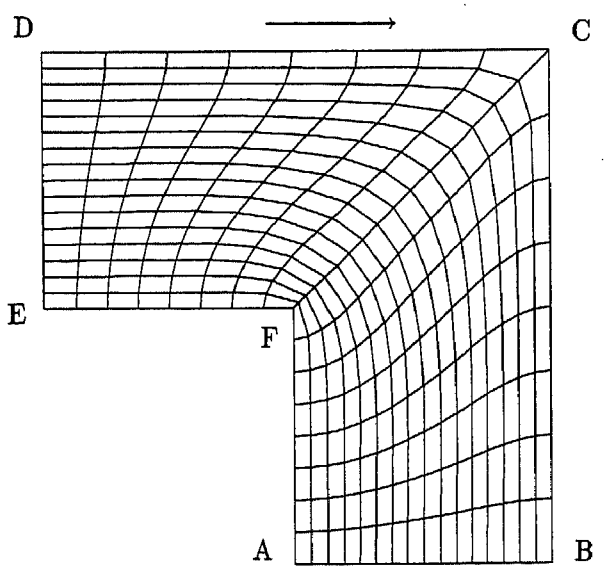


Figure 6.4: The L-shaped driven cavity problem with a grid generated by a biharmonic grid generator

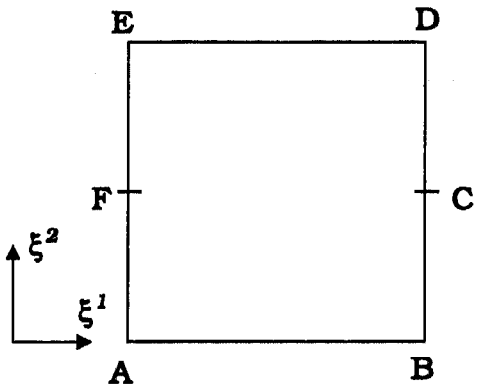


Figure 6.5: The computational domain for the L-shaped driven cavity problem (for the physical domain, see figure 6.4)

not work well or fails for lower Reynolds numbers and smaller mesh sizes, if the numbering of cells is the lexicographic ordering as used for the skewed driven cavity problem. The reason is explained in [13] (see section 7.8 and the references therein) for anisotropic convection-diffusion equations. Therefore, instead of the lexicographic ordering, a backward lexicographic ordering is employed, in which the numbering of cells takes place first in the reverse direction of the ξ^1-direction and then in the reverse direction of the ξ^2-direction, starting from corner D. The Reynolds numbers are 100 and 1000, respectively. Figures 6.6 and 6.7 give the streamlines for the two cases and are in good agreement with those given in [7]. One multigrid iteration is employed for each time step. Note that the time steps for $Re = 100$ and $Re = 1000$ are different. Compared with the time step for the skewed driven cavity problems, the time step for $Re = 1000$ has to be smaller, otherwise the multigrid algorithm fails after a few time steps, because of lack of diagonal dominance, as discussed before. Of course the time step can be larger for low Reynolds numbers, for example for $Re = 100$ here. But in accordance with the case for $Re = 1000$, we take $\Delta t = .5$ for both cases in measuring reduction factors. The reduction factors and the dependence of multigrid convergence on ζ are presented in tables 6.3 and 6.4.

In this test problem, the parameter ζ must be greater than 1 and can be rather large. From both the skewed driven cavity and the L-shaped driven cavity problems, it is clear that the optimal value of ζ is problem-dependent and an appropriate choice of ζ improves the multigrid performance. But in both cases $\zeta = 2$ would give satisfactory convergence.

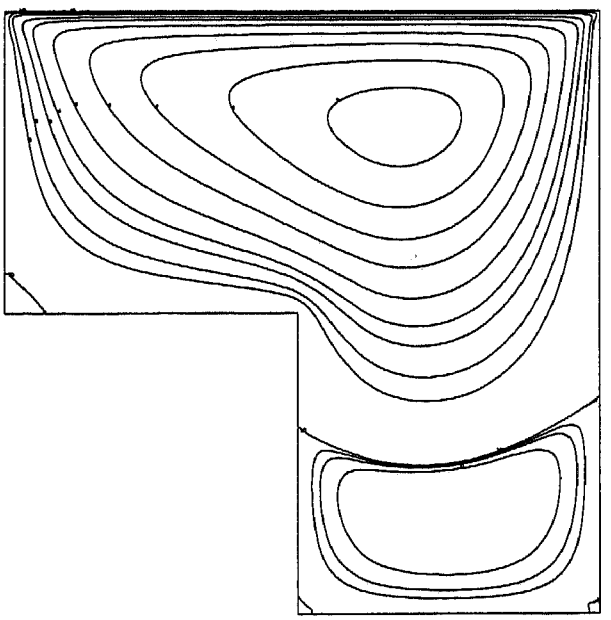


Figure 6.6: Streamlines for the L-shaped driven cavity problem, $Re = 100$, $\Delta t = 1$, 20 times steps, $r_s^t / r_s^0 < 1.905 \times 10^{-9}$, on 128×128 grid

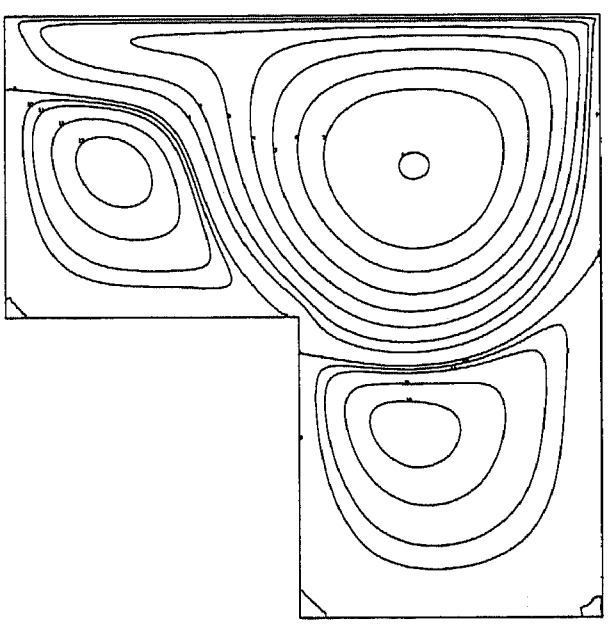


Figure 6.7: Streamlines for the L-shaped driven cavity problem, $Re = 1000$, $\Delta t = .2$, 100 times steps, $r_s^t / r_s^0 < 1.172 \times 10^{-4}$, on 128×128 grid

Table 6.3: Reduction factors for the L-shaped driven cavity problem, $\zeta = 2$

$Re = 100$, **P** for **RAP** version 1				
Grid	16×16	32×32	64×64	128×128
i	16	16	16	16
ρ_i	.3281	.3364	.3268	1.054
ρ_{i+1}	.3296	.3384	.3322	1.054
ρ_{i+2}	.3309	.3401	.3367	1.054
ρ_{i+3}	.3319	.3415	.3404	1.054
ρ_{i+4}	.3328	.3428	.3433	1.054
r_0	$.3178 \times 10^{-3}$	$.3315 \times 10^{-3}$	$.3518 \times 10^{-3}$	$.1931 \times 10^{-2}$
r_{i+4}	$.1117 \times 10^{-13}$	$.1744 \times 10^{-13}$	$.2170 \times 10^{-13}$	$.6013 \times 10^{-2}$
$\bar{\rho}$	.3001	.3062	.3087	1.058
$Re = 1000$, **P** for **RAP** version 1				
i	16	16	16	16
ρ_i	.3061	.2792	.3307	.3395
ρ_{i+1}	.3081	.2869	.3343	.3406
ρ_{i+2}	.3095	.2857	.3370	.3417
ρ_{i+3}	.3104	.2894	.3390	.3427
ρ_{i+4}	.3110	.2956	.3407	.3436
r_0	$.1241 \times 10^{-3}$	$.1939 \times 10^{-3}$	$.1468 \times 10^{-3}$	$.9756 \times 10^{-4}$
r_{i+4}	$.6374 \times 10^{-15}$	$.3265 \times 10^{-14}$	$.3407 \times 10^{-13}$	$.1066 \times 10^{-13}$
$\bar{\rho}$	.2726	.2893	.3111	.3176
$Re = 100$, **P** for **RAP** version 2				
i	16	16	16	16
ρ_i	.3256	.3338	.3392	.5285
ρ_{i+1}	.3274	.3360	.3418	.2465
ρ_{i+2}	.3290	.3377	.3438	.2215
ρ_{i+3}	.3303	.3392	.3453	.4931
ρ_{i+4}	.3313	.3404	.3465	.3166
r_0	$.3096 \times 10^{-3}$	$.3104 \times 10^{-3}$	$.3076 \times 10^{-3}$	$.4837 \times 10^{-3}$
r_{i+4}	$.9552 \times 10^{-14}$	$.1471 \times 10^{-13}$	$.1531 \times 10^{-13}$	$.1873 \times 10^{-13}$
$\bar{\rho}$	.2982	.3046	.3054	.3016
$Re = 1000$, **P** for **RAP** version 2				
i	16	16	16	16
ρ_i	.3126	.2737	.3200	.3384
ρ_{i+1}	.3143	.3098	.3255	.3395
ρ_{i+2}	.3156	.3135	.3312	.3405
ρ_{i+3}	.3164	.2879	.3360	.3414
ρ_{i+4}	.3169	.2834	.3391	.3423
r_0	$.1208 \times 10^{-3}$	$.1859 \times 10^{-3}$	$.1438 \times 10^{-3}$	$.9281 \times 10^{-4}$
r_{i+4}	$.8155 \times 10^{-15}$	$.5311 \times 10^{-14}$	$.1032 \times 10^{-13}$	$.1025 \times 10^{-13}$
$\bar{\rho}$	.2764	.2970	.3110	.3178

Table 6.4: The dependence of multigrid convergence on ζ in the L-shaped driven cavity problem, finest grid=128 x 128, **RAP** version 2

$Re = 100$, $r_0 = .4837 \times 10^{-3}$							
ζ, i	ρ_i	ρ_{i+1}	ρ_{i+2}	ρ_{i+3}	ρ_{i+4}	r_{i+4}	$\bar{\rho}$
1,*	div						
1.5,16	.6650	.6855	.6890	.7198	.7366	$.1207 \times 10^{-8}$	.5246
2,16	.5285	.2465	.2215	.4931	.3166	$.1873 \times 10^{-13}$	.3016
4,10	.1919	.1951	.1934	.1934	.1978	$.2582 \times 10^{-13}$	.1846
6,10	.1504	.1550	.1566	.1561	.1554	$.1940 \times 10^{-14}$	.1534
10,10	.1308	.1358	.1406	.1454	.1685	$.3214 \times 10^{-15}$	.1350
20,10	.1476	.1500	.1523	.1553	.1933	$.2704 \times 10^{-15}$	.1333
$Re = 1000$, $r_0 = .9281 \times 10^{-4}$							
1,*	div						
1.5,16	.6405	.7172	.7516	.7658	.7729	$.9667 \times 10^{-11}$	.4476
2,16	.3384	.3395	.3405	.3414	.3423	$.1025 \times 10^{-13}$	.3178
4,14	.2188	.2196	.2202	.2215	.2232	$.5794 \times 10^{-16}$	.2099
6,13	.1840	.1842	.1845	.1846	.2058	$.1683 \times 10^{-16}$	.1780
10,12	.1730	.1728	.1726	.1730	.1856	$.2123 \times 10^{-16}$	.1622
20,12	.1787	.1786	.1785	.1788	.1910	$.2378 \times 10^{-16}$	.1633

7 Conclusions

Based on collective incomplete LU factorization with r-transformation, a new smoother, called CILU, is presented for the incompressible Navier-Stokes equations in general coordinates. Instead of working with scalar elements as ordinary ILU, CILU works with elements that are 3×3 matrices. Apart from the underrelaxation factor ω, another parameter ζ is introduced to enhance smoothing performance. A multigrid algorithm using CILU as smoother is investigated numerically, using the skewed driven cavity and the L-shaped driven cavity problems as test problems. The performance of the multigrid algorithm is studied by measuring the limiting reduction factor and the average reduction factor on various grids and for different choices of prolongation operators in the computation of coarse grid matrices by means of Galerkin coarse grid approxiamtion (RAP). Two versions are used for the prolongation operators: in version 1, the prolongation operators for the velocities are the so-called hybrid interpolations, and that for the pressure is a piecewise constant interpolation; in version 2, the prolongation operators for the velocities are a bilinear interpolation, and that for the pressure remains the same as in version 1. The multigrid schedule is the W-cycle with one pre- and one post-smoothing, and the coarsest grid is fixed at 2×2.

The numerical experiments show that with version 2, the reduction factors are almost independent of mesh sizes and slightly dependent on the Reynolds number. But with version 1, the reduction factors grow with refining mesh sizes, and are mostly larger than those obtained with version 2, and the algorithm works better for the low Reynolds number case than for the high Reynolds number case. So the multigrid algorithm with prolongation operators from version 2 seems to be more promising.

The effect of the parameter ζ is investigated on 128×128 grids. The results show that a proper choice of ζ improves the multigrid performance, sometimes very much as in the L-shaped driven cavity prolem. The optimal value of ζ is problem-dependent, but a fixed choice $\zeta = 2$ seems to be a good compromise.

The well-known anisotropy of ILU smoothers is encountered here in the L-shaped driven cavity problem, where meshes are stretched more in a direction than in another. This problem is cured by simply changing the ordering of cells.

Due to central differencing of the partial differential equations, the time step should be sufficiently small for high Reynolds numbers. Otherwise the algorithm may fail after several time steps.

To sum up, CILU smoother is a good smoother.

Acknowledgement The authors would like to thank A. Segal and K. Kassels for their efforts in making the ISNaS code available for this computation. The authors are also grateful to C.W. Oosterlee and E. Brakkee for providing test problems and grids, as well as some useful discussions.

References

[1] Brandt, A., Multigrid techniques: 1984 guide with applications to fluid dynamics. GMD-Studien, **85**, Gesellschaft für Mathematik und Datenverarbeitung MBH Bonn, Germany, 1984.

[2] Brandt, A. and N. Dinar, *Multigrid solutions to flow problems.* In S. Parter (ed.): Numerical Methods for Partial Differential Equations, Academic Press, New York, 53–147, 1979.

[3] Demirdžić, I., Ŭ. Lilek and M. Perić, Fluid flow and heat transfer test problems for non-orthogonal grids: Bench-mark solutions. Int. J. Numer. Methods in Fluids, **15**: 329–354, 1992.

[4] Hackbusch, W., *Multi-grid methods and applications.* Springer, Berlin, 1985.

[5] Mynett, A.E., P. Wesseling, A. Segal and C.G.M. Kassels, *The ISNaS incompressible Navier-Stokes solver: invariant discretization.* Applied Scientific Research, **48**: 175–191, 1991.

[6] Van Kan, J., C.W. Oosterlee, A. Segal and P. Wesseling, *Discretization of the incompressible Navier-Stokes equations in general coordinates using contravariant velocity components.* Report 91-09, TWI, TU Delft, 1991.

[7] Oosterlee, C.W., P. Wesseling, A. Segal, E. Brakkee, *Benchmark solutions for the incompressible Navier-Stokes equations in general coordinates on staggered grids.* Report 92-67, TWI, TU Delft, 1992.

[8] Patankar, S.V. and D.B. Spalding, *A calculation procedure for heat and mass transfer in three-dimensional parabolic flows.* Int. J. Heat Mass Transfer, **15**: 1787–1806, 1972.

[9] Segal, A., P. Wesseling, J. van Kan, C.W. Oosterlee and K. Kassels, *Invariant discretization of the incompressible Navier-Stokes equations in boundary fitted co-ordinates.* Int. J. Numer. Methods in Fluids, **15**: 411–426, 1992.

[10] Varga, R.S., *Matrix iterative analysis.* Prentice-Hall, Englewood, Cliffs, NJ, 1962.

[11] Wesseling, P., *Two remarks on multigrid methods.* In W. Hackbusch (ed.): Robust Multigrid Methods. Proc. 4-th GAMM-Seminar, Kiel, Germany. Notes on Numer. Fluid Mech. **23**: 209–216, Vieweg, Braunschweig, 1988.

[12] Wesseling, P., *A survey of Fourier smoothing analysis results.* In: W. Hackbusch and U. Trottenberg (eds.): Multigrid Methods III. Proc. 3rd European Conf. on Multigrid Methods, Bonn, Oct.1–4, 1990. In: K.-H. Hoffmann, H.D. Mittelmann and J.Todd (eds.): International Series of Numerical Mathematics, **98**, 1991.

[13] Wesseling, P., *An introduction to multigrid methods.* John Wiley & Sons, Chichester, 1992.

[14] Wesseling, P., A. Segal, J. van Kan, C.W. Oosterlee and C.G.M. Kassels, *Finite volume discretization of the incompressible Navier-Stokes equations in general coordinates on staggered grids.* Comp. Fluid Dyn. J. **1**: 27–33, 1992.

[15] Wittum, G., *Multi-grid methods for Stokes and Navier-Stokes equations—Transforming smoothers: Algorithm and Numerical Results.* Numer. Meth. **54**: 543–563, 1989.

[16] Wittum, G., *On the convergence of multi-grid methods with transforming smoothers—Theory with Applications to the Navier-Stokes Equations.* Numer. Math. **57**: 15–38, 1990.

[17] Wittum, G., *The use of fast solvers in computational fluid dynamics.* In P. Wesseling (ed.): Proc. 8-th GAMM-Conference on Numer. Methods in Fluid Mech., September 27-29, 1989, Delft, the Netherlands. Notes on Fluid Numer. Mech. **29**: 574–581, Vieweg, Braunschweig, 1990.

[18] Wittum, G., *R-transforming smoothers for the incomressible Navier-Stokes equations.* In W. Hackbusch (ed.): Proc. 5-th GAMM-Seminar, January 20-22, 1989, Kiel, Germany. Notes on Numer. Fluid Mech. **30**: 153–162, Vieweg, Braunschweig, 1990.

[19] Zeng, S. and P. Wesseling, *Galerkin coarse grid approximation for the incompressible Navier-Stokes equations in general coordinates.* Report 92-35, TWI, TU Delft, 1992.

[20] Zeng, S. and P. Wesseling, *An efficient algorithm for the computation of Galerkin coarse grid approximation for the incompressible Navier-Stokes equations.* Report 92-40, TWI, TU Delft, 1992.