DELFT UNIVERSITY OF TECHNOLOGY

REPORT 92-102

Multigrid schemes for time-dependent incompressible Navier-Stokes equations

C.W. Oosterlee
P. Wesseling

ISSN 0922-5641

Reports of the Faculty of Technical Mathematics and Informatics no. 92-10

Delft 1992

Proposed running head: Multigrid schemes for time-dependent Navier-Stokes

Correspondence to:

C.W. Oosterlee
P.O. Box 5031
2600 GA Delft, the Netherlands.
phone: 3115-781692
fax: 3115 787209
e-mail: witaoos@dutinfh.tudelft.nl.

Abstract:
The instationary incompressible Navier-Stokes equations are discretized with a finite volume method in curvilinear coordinates on a staggered grid. Several multigrid schemes to solve time-dependent equations are presented, investigated and compared: The time marching scheme, a parabolic multigrid scheme and a multigrid waveform relaxation scheme. All methods will be based on the same robust smoother, a coupled smoothing method. Furthermore the behaviour of the BDF(2)- time discretization scheme is investigated. Some test problems are investigated, not only rectangular ones.

1 Introduction.

There are a number of methods to solve time-dependent equations. Different approaches have been suggested to treat the "time-marching" in this type of equations. A standard approach is to solve the equations one time step after another, which is called a time marching scheme. A next time step is tackled when a converged solution on the former time level is obtained. Each time step the equations are solved iteratively in space by, for example, a multigrid method. Examples of this approach are found in [11], [34] and [37]. However, the sequential nature of time marching schemes does not lend itself well for implementation on parallel machines. Therefore new methods have been developed to solve time-dependent equations more efficiently on parallel machines. In 1984 Hackbusch ([10]) introduced two multigrid approaches called parabolic multigrid methods. Here the time-direction can be seen as one of the axes in a space-time grid. The multigrid procedure updates all unknowns in this space-time grid. In these approaches the equations are not solved, but smoothed time step per time step. Some components of the multigrid methods, like prolongation and restriction, can be done in parallel. In one of the methods the smoothing algorithm is sequential. The smoothing procedure on a new time level uses updated values from the previous time levels. In the second multigrid method proposed several time steps can be smoothed simultaneously; "old" values from previous time levels are then used. This smoothing method can be efficiently implemented on parallel machines. The computation on each time level can be independently performed by a processor of a parallel computer. Results with these methods are described in [3] and [5] for the one-dimensional unsteady heat equation, in [6] and [15] for the unsteady incompressible Navier-Stokes equations in primitive variables, and in [18] for the unsteady incompressible Navier-Stokes equations in velocity-vorticity formulation. Parabolic multigrid methods or time-parallel multigrid methods for incompressible Navier-Stokes equations are described in more detail in section 3.
A different approach based on the so-called waveform relaxation methods is proposed by (amongst others) Vandewalle and described in detail in [29], [28]. In waveform relaxation methods an approximation of an unknown in space is calculated along a time interval of interest consisting of a number of time-steps. Instead of updating scalars time-step by time-step functions in time are updated. Waveform relaxation schemes can be accelerated by multigrid. This solution method also lends itself well for parallel implementation as is shown in [28] for several initial value and time periodic problems. In section 4 a multigrid waveform relaxation algorithm is presented for the incompressible Navier-Stokes equations.
In this paper we investigate the multigrid methods discussed above for the incompressible Navier-Stokes equations in general coordinates. All methods will be based on the same smoother, presented in section 2, which is robust, showed good results solving the steady incompressible Navier-Stokes equations ([21]), and posesses good possibilities for efficient implementation on parallel machines. The algorithms are not implemented as a code for parallel machines. We investigate the performance on a Convex 3820 computer on one processor, but efficiency on parallel machines is considered.

2 The discretization of the incompressible Navier-Stokes equations in general coordinates.

The incompressible Navier-Stokes equations are presented in general coordinates. Due to the discretization for boundary-fitted grids flows in domains of general shape can be calculated. In

general coordinates the unsteady incompressible Navier-Stokes equations are given (in tensor notation [2]) by:

$$U^{\alpha}_{,\alpha} = 0 \tag{1}$$

$$\frac{\partial \rho U^{\alpha}}{\partial t} + (\rho U^{\alpha} U^{\beta})_{,\beta} + (g^{\alpha\beta} p)_{,\beta} - \tau^{\alpha\beta}_{,\beta} = \rho F^{\alpha} \tag{2}$$

where $\tau^{\alpha\beta}$ represents the deviatoric stress tensor given by:

$$\tau^{\alpha\beta} = \mu(g^{\alpha\gamma} U^{\beta}_{,\gamma} + g^{\gamma\beta} U^{\alpha}_{,\gamma}) \tag{3}$$

Here U^{α} are contravariant velocity components, ρ is density, p is pressure and μ is the viscosity coefficient. F^{α} represents a body force. The coordinate transformation is given by $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\xi})$, with $\boldsymbol{x}$ Cartesian coordinates in domain Ω and $\boldsymbol{\xi}$ boundary conforming curvilinear coordinates in a transformed rectangular uniform computational domain G.
Covariant base vectors $\boldsymbol{a}_{(\alpha)}$, contravariant base vectors $\boldsymbol{a}^{(\alpha)}$, and the covariant and contravariant metric tensors $g_{\alpha\beta}$ and $g^{\alpha\beta}$ are defined as

$$\boldsymbol{a}_{(\alpha)} = \frac{\partial \boldsymbol{x}}{\partial \xi^{\alpha}}, \quad \boldsymbol{a}^{(\alpha)} = \frac{\partial \xi^{\alpha}}{\partial \boldsymbol{x}}, \quad g_{\alpha\beta} = \boldsymbol{a}_{(\alpha)} \cdot \boldsymbol{a}_{(\beta)} \quad , \quad g^{\alpha\beta} = \boldsymbol{a}^{(\alpha)} \cdot \boldsymbol{a}^{(\beta)} \tag{4}$$

The determinant of the covariant metric tensor $g_{\alpha\beta}$ is denoted by g; $\sqrt{g}$ is found to be (in two dimensions)

$$\sqrt{g} = a^{1}_{(1)} a^{2}_{(2)} - a^{2}_{(1)} a^{1}_{(2)} \tag{5}$$

Summation over repeated Greek indices is implied over their range $\{1, 2, ..., d\}$ with d the number of space dimensions, which will be two in this paper.
Finite volume integration is performed on (1) and (2). For (1) the following covariant differentiation rule is used,

$$U^{\alpha}_{,\alpha} = \frac{1}{\sqrt{g}} \frac{\partial \sqrt{g} U^{\alpha}}{\partial \xi^{\alpha}} \tag{6}$$

in which $\sqrt{g}$ is the Jacobian of the transformation.
Terms of the type $T^{\alpha\beta}_{,\beta}$ in (2) are given by,

$$T^{\alpha\beta}_{,\beta} = \frac{1}{\sqrt{g}} \frac{\partial \sqrt{g} T^{\alpha\beta}}{\partial \xi^{\beta}} + \left\{ {\alpha \atop \gamma\beta} \right\} T^{\gamma\beta} \tag{7}$$

where $\left\{ {\alpha \atop \gamma\beta} \right\}$ represents the Christoffel symbol of the second kind, defined by

$$\left\{ {\alpha \atop \gamma\beta} \right\} = \left\{ {\alpha \atop \beta\gamma} \right\} = \boldsymbol{a}^{(\alpha)} \cdot \frac{\partial \boldsymbol{a}_{(\gamma)}}{\partial \xi^{\beta}} = \frac{\partial \xi^{\alpha}}{\partial x^{\delta}} \frac{\partial^2 x^{\delta}}{\partial \xi^{\gamma} \partial \xi^{\beta}} \tag{8}$$

Discretization takes place as follows. A staggered grid ([13]) is used. Unknowns $V^{\alpha} = \sqrt{g} U^{\alpha}$ are used as primary unknowns together with the pressure. The spatial discretization of the equations is explained in [25], [20], [21] and [22]. Our discretization contains upwind discretization of the convective terms by means of the hybrid difference scheme ([24], [23], [20]). The lower spatial accuracy of a hybrid difference scheme can be compensated by using a defect correction technique ([14], [34]). A major problem for the numerical solution of the incompressible Navier-Stokes equations is the absence of a pressure term in the continuity

equation and therefore the resulting matrix (shown in figure 1) contains a zero diagonal block. There is no explicit equation for the pressure. Sometimes this problem is avoided by adopting instead of the so-called primitive formulation with velocity components and pressure as primary unknowns the vorticity-streamfunction formulation. However, this formulation is difficult to extend to three-dimensional problems. In order to solve complex flow problems in the primitive formulation three different approaches have been developed. One is called the uncoupled solution technique, in which the velocity components are calculated from the momentum equations seperately from a "pressure correction equation", a discretized form of the continuity equation combined with the momentum equations. A second order accurate pressure correction scheme is presented in [32] and is used in the ISNaS code described in [19], [25]. Pressure-Correction methods are very efficient for time-dependent problems, because one only needs to solve convection-diffusion type equations (usually with a large coefficient on the main diagonal coming from the time derivative) and a Poisson type equation for the pressure correction. They are common use in practice. For steady equations, uncoupled solution techniques of distributive iteration type ([36]), such as the so-called SIMPLE-type methods ([24], [23]) are available. In the steady case the advantage of uncoupled solution techniques is less pronounced than in the time-dependent case, because they require assumptions about variables yet to be calculated and consequently need iterations ([1]).
In a second approach, called coupled solution methods, the discretized momentum and continuity equations are solved simultaneously. A coupled solution technique is for example proposed in [30], where a coupled cell-by-cell solution procedure is proposed as smoothing technique in a multigrid solution method. In each cell the momentum equations and the continuity equation are updated by means of a point Gauss-Seidel-type iteration. This approach is efficient for steady equations ([31], [17], [20]), for unsteady equations results can be found in [37]. The method adopted in this paper is a coupled solution method.
A third approach is adopted from compressible flow solvers, where for each unknown an explicit equation exists, for velocity components the momentum equations, for density the continuity equation, and for pressure and temperature the equation of state and the enthalpy equation. This approach has been extended to incompressible flows by means of the so-called pseudo-compressibility method ([7]). An artificial time-dependent pressure term is added to the continuity equation. A disadvantage of this method is the appearance of a parameter, which is placed in front of the additional term and determines the convergence rate. For time-dependent incompressible Navier-Stokes equations this latter approach has been used in [26] and in many other papers.
The resulting matrix is sparse (as can be seen in Figure 1), due to the fact that unknowns in a momentum equation are only connected to their neighbours. The band for the momentum equations contains at most (when the grid is non-orthogonal) 19 elements, namely 13 velocity component terms and 6 pressure terms.

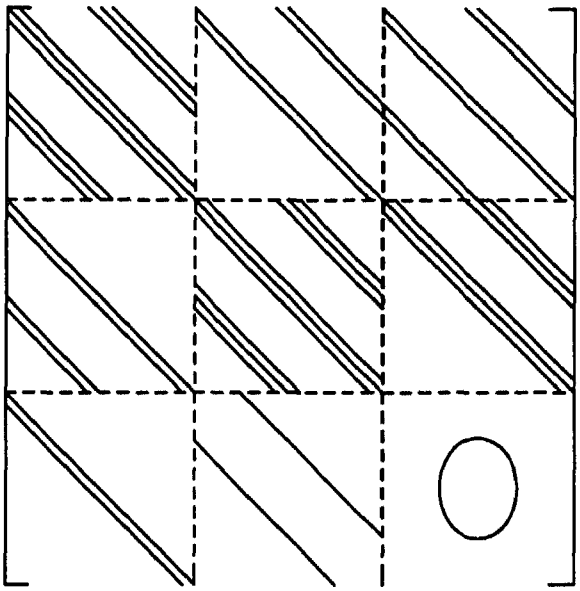

Figure 1: The matrix structure for the incompressible Navier- Stokes equations.

Time discretization with the so-called θ-method is done as follows:

$$U_{,\alpha}^{\alpha(n+1)} = 0 \tag{9}$$

$$\frac{\rho U^{\alpha(n+1)} - \rho U^{\alpha(n)}}{\Delta t} + \theta\{(\rho U^{\alpha} U^{\beta})_{,\beta} - \tau_{,\beta}^{\alpha\beta} + (g^{\alpha\beta} p)_{,\beta}\}^{(n+1)} +$$

$$(1-\theta)\{(\rho U^{\alpha} U^{\beta})_{,\beta} - \tau_{,\beta}^{\alpha\beta} + (g^{\alpha\beta} p)_{,\beta}\}^{(n)} = \theta \rho F^{\alpha(n+1)} + (1-\theta)\rho F^{\alpha(n)} \tag{10}$$

which we abbreviate as:

$$U_{,\alpha}^{\alpha(n+1)} = 0 \tag{11}$$

$$\frac{\rho U^{\alpha(n+1)} - \rho U^{\alpha(n)}}{\Delta t} + \theta T_{,\beta}^{\alpha\beta(n+1)} + (1-\theta) T_{,\beta}^{\alpha\beta(n)} = \theta \rho F^{\alpha(n+1)} + (1-\theta)\rho F^{\alpha(n)} \tag{12}$$

For $\theta = 1$ we obtain the implicit Euler scheme, for $\theta = \frac{1}{2}$ the second order accurate Crank-Nicolson scheme. The time discretization with the BDF(2)-scheme ([9]), also called B3-scheme consists of a second order approximation of the time derivative and a fully implicit treatment of all spatial derivatives, as follows,

$$U_{,\alpha}^{\alpha(n+1)} = 0 \tag{13}$$

$$\frac{3\rho U^{\alpha(n+1)} - 4\rho U^{\alpha(n)} + \rho U^{\alpha(n-1)}}{2\Delta t} + T_{,\beta}^{\alpha\beta(n+1)} = \rho F^{\alpha(n+1)} \tag{14}$$

The time marching scheme. Each timestep the discretized equations are solved with the standard non-linear multigrid method ([4], [11]). Details are presented in [34] and [20]. The algorithm can do multigrid V, F and W iteration cycles. Prolongation and restriction operators are more or less dictated by the staggered grid arrangement and the uniform computational

grid G. Prolongation operators are derived for all variables using bilinear interpolation.
The restricted coarse grid fluxes V^α are defined to be the mean of their two neighbouring fine grid fluxes. Coarse grid pressures are defined to be the mean of the four neighbouring fine grid pressures. In evaluating the coarse grid right-hand side area weighting is used for the fine grid residuals, as follows

$$r_{i,j}^{(1)k-1} = 1/8(r_{2i-2,2j}^{(1)k} + r_{2i-2,2j-1}^{(1)k} + r_{2i,2j}^{(1)k} + r_{2i,2j-1}^{(1)k}) + 1/4(r_{2i-1,2j}^{(1)k} + r_{2i-1,2j-1}^{(1)k}) \quad (15)$$

$$(16)$$

$$r_{i,j}^{(2)k-1} = 1/8(r_{2i,2j-2}^{(2)k} + r_{2i-1,2j-2}^{(2)k} + r_{2i,2j}^{(2)k} + r_{2i-1,2j}^{(2)k}) + 1/4(r_{2i,2j-1}^{(2)k} + r_{2i-1,2j-1}^{(2)k}) \quad (17)$$

$$(18)$$

$$r_{i,j}^{(3)k-1} = 1/4(r_{2i-1,2j}^{(3)k} + r_{2i-1,2j-1}^{(3)k} + r_{2i,2j}^{(3)k} + r_{2i,2j-1}^{(1)k}) \quad (19)$$

where the superscripts $1, 2, 3$ refer to the two momentum equations and the continuity equation respectively. The grid point numbering is given in figure 2. The coarse grid approximations of the geometric quantities, needed for the discretizations on boundary-fitted grids, are calculated from the coordinate vertices from the coarse grids. This resulted in satisfactory multigrid reduction factors for several arbirary domains ([21]).

Figure 2: Definition of grid point numbering. The coarse grid cell shown is the union of the four fine grid cells shown. The grids are divided into triads with gridpoint indices: $\bar{A} = \bar{B} = \bar{C} = (i, j), \bar{D} = (i+1, j), \bar{E} = (i, j+1)$ and $A = B = C = (2i, 2j), D = E = F = (2i, 2j-1)$.

The smoothing method is called Symmetric Coupled Alternating Lines (SCAL) ([27], [21]) and is now described briefly.
All velocity components and pressures in a line of cells are updated simultaneously. SCAL is a line-by-line version of the cell-by-cell smoother SCGS introduced in [30]. SCGS has been used as smoothing method for discretizations of the steady incompressible Navier-Stokes equations in curvilinear coordinates in [17] and [20]. The smoother SCAL is robust. Many steady problems in arbirary domains, where cells with large aspect ratio occurred, have been solved succesfully ([21]).
For each line a banded system is solved and intermediate values (V^{1*}, V^{2*}, p^*) are obtained. In the steady case the new values $(V^{1(n+1)}, V^{2(n+1)}, p^{(n+1)})$ are found with underrelaxation. For the unsteady case the additional diagonal term $\frac{1}{\Delta t}$ acts as an underrelaxation term, therefore no additional relaxation is needed. After an update of unknowns along horizontal rows of cells, unknowns are updated again using vertical rows. SCAL is a zebra type smoother: first all odd (white) rowes are visited, then all even (black) rowes are visited. With special

ordering strategies acceleration can be obtained on parallel computers.
Each time-step several multigrid iterations are performed until a termination criterion is met. Then the solution is considered accurate enough and a next time step is tackled. The termination criterion used is:

$$||res_n^{(i)}||_2 < 1 \times 10^{-3}||\mathrm{RHS}_0||_2 \tag{20}$$

That is, a next timestep $(i+1)$ is started when the 2-norm of the residual after n iterations is less than 10^{-3} times the norm of the initial right-hand side. This appeared to be a good termination criterion ([33]), while it is scaling invariant and independent of the initial estimate.
An important aspect of time marching schemes is that every time-step starts with a good initial approximation of the solution vector:

$$u_h^{i+1} = u_h^i \tag{21}$$

or:

$$u_h^{i+1} = 2u_h^i - u_h^{i-1} \tag{22}$$

Comparing this coupled time marching approach using multigrid to an uncoupled time marching approach for the unsteady incompressible Navier-Stokes equations ([25]) where a pressure correction method ([32]) is used and the momentum equations and pressure equations are solved with GMRES methods ([33]) it appeared that the coupled approach was slower than the uncoupled approach for a rectangular test problem.
The pressure-correction technique can be incorporated in a parabolic multigrid method ([16]) for the unsteady case. However, to the authors of this paper it is not clear at the moment how an uncoupled solution technique could be incorporated in a waveform algorithm.
Finally it is to be noted that this coupled solution technique is robust; arbitrarily large time steps can be taken in arbitrary domains. Furthermore, all white rowes as well as all black rowes can be done in parallel, each on a single processor. Probably for many problems a red-black cell-by-cell smoother will be more efficient (well vectorizable), but certainly less robust !

3 The parabolic / time-parallel multigrid method.

The sequential process of solving equations time-step by time-step with a time marching scheme makes algorithms less efficient on parallel machines. In the following multigrid schemes based on a paper by Hackbusch ([10]) the time-axis in a space-time grid is an axis along which solutions will be updated simultaneously for a number of time-steps. A convergence criterion must be satisfied for all unknowns in this grid, so when the criterion is met all solutions on all time levels considered will be accurate enough. The nonlinear time-parallel two-grid method, for an implicit Euler scheme looks like (with T_h a discrete version of $T_{,\beta}^{\alpha\beta}$):

Nonlinear time-parallel two-grid algorithm:

begin algorithm
for time levels $n = 1$ step 1 until n_{max} **do :**
for number of iterations $\nu = 1$ step 1 until ν_{max} **do :**
for spatial indices $i \in G_h$ **do:**

• Apply a pre-smoothing iteration (sequential or parallel)

enddo

---

for time levels $n = 1$ step 1 until n_{max} **do :**
• Compute residual:

$$\text{res}^{(n+1)} = (\boldsymbol{F}_h^{(n+1)} + \frac{I}{\Delta t}\boldsymbol{u}_h^{(n)}) - (\frac{I}{\Delta t} + T_h)\boldsymbol{u}_h^{(n+1)} \tag{23}$$

enddo

---

for time levels $n = 1$ step 1 until n_{max} **do :**
• Choose $\tilde{\boldsymbol{u}}_H^{(n+1)}$.
• Apply Restriction $\boldsymbol{R}^H : \boldsymbol{u}_H^{(n)} = \boldsymbol{R}^H \boldsymbol{u}_h^{(n)}$, $\boldsymbol{R}^H : \tilde{\boldsymbol{u}}_H^{(n)} = \boldsymbol{R}^H \boldsymbol{u}_h^{(n)}$.
• Apply Restriction $\boldsymbol{R}^H$ conform (17)...(19) to $\text{res}^{(n+1)}$.
enddo

---

for $n = 1$ step 1 until n_{max} **do :**
• Solve the coarse grid equation for $\boldsymbol{u}_H^{(n+1)}$.

$$\frac{\boldsymbol{u}_H^{(n+1)} - \boldsymbol{u}_H^{(n)}}{\Delta t} + T_H \boldsymbol{u}_H^{(n+1)} = \boldsymbol{F}_H^{(n+1)} \tag{24}$$

$$= s\boldsymbol{R}^H(\text{res}^{(n+1)}) + (\frac{I}{\Delta t} + T_H)\tilde{\boldsymbol{u}}_H^{(n+1)} - \frac{I}{\Delta t}\tilde{\boldsymbol{u}}_H^{(n)} \tag{25}$$

enddo

---

for time levels $n = 1$ step 1 until n_{max} **do :**
• Prolongation:

$$\boldsymbol{u}_h^{(n+1)} = \boldsymbol{u}_h^{(n+1)} + \frac{1}{s}\boldsymbol{P}^h(\boldsymbol{u}_H^{(n+1)} - \tilde{\boldsymbol{u}}_H^{(n+1)}) \tag{26}$$

enddo

---

for time levels $n = 1$ step 1 until n_{max} **do :**
for number of iterations $\nu = 1$ step 1 until ν_{max} **do :**
for spatial indices $i \in G_h$ **do:**
• Apply a post-smoothing iteration (sequential or parallel)

enddo

---

end nonlinear time-parallel two-grid algorithm.

The parameter s is a parameter (usually chosen 1) for the nonlinear multigrid algorithm ([11]) to keep the right-hand-side of the coarse grid equation in the range of the coarse grid operator. The coarse grid equation can be solved in a similar way, i.e. by introducing a third coarser grid and by executing a number of two-grid cycles, etcetera. The introduction of a sequence of grids leads to the time-parallel multigrid algorithm, also called parabolic multigrid algorithm with different iteration cycles. Here again V-, F- and W-cycles are implemented. Note that all different stages in the multigrid algorithm can be performed in parallel in time direction. The restriction and prolongation operators are only spatial operators. In [10] it is stated that coarsening in time does not lead to an efficient algorithm. Errors which are smooth in spatial direction, but nonsmooth in time are then not improved by smoothing, because the defects are smooth in space. Further the coarse grid correction does not provide a good correction because of the coarser timestep. Therefore we do not apply grid coarsening in time.
The two different (sequential or parallel) smoothing algorithms are now described. We consider smoothers of the following type:
Sequential smoothing algorithm:

begin
for time levels $n = 1$ step 1 until n_{max} **do :**
for iteration number $\nu = 1$ step 1 until ν_{max} **do :**
for space indices $i \in G_h$ **do:**
$$((\frac{I}{\Delta t} + M_h)\boldsymbol{u}^{(n+1),\nu})_i = (N_h\boldsymbol{u}^{(n+1),\nu-1})_i + (\frac{I}{\Delta t}\boldsymbol{u}^{(n),\nu})_i + \boldsymbol{F}_i^{(n+1)}$$
enddo
end sequential smoothing.

Parallel smoothing algorithm:

begin
for $\nu = 1$ step 1 until ν_{max} **do :**
for $n = 1$ step 1 until n_{max} **do :**
for $i \in G_h$ **do:**
$$((\frac{I}{\Delta t} + M_h)\boldsymbol{u}^{(n+1),\nu})_i = (N_h\boldsymbol{u}^{(n+1),\nu-1})_i + (\frac{I}{\Delta t}\boldsymbol{u}^{(n),\nu-1})_i + \boldsymbol{F}_i^{(n+1)}$$
enddo
end parallel smoothing.

Note that in the latter case all time-steps can be done in parallel.
In [3] results are obtained efficiently with a parabolic multigrid method on a transputer system. A sequential smoother of Gauss-Seidel type is compared to a parallel smoother of Jacobi type for a parabolic differential equation. It was found that a good speed-up was obtained for the sequential smoother for many processors in a model problem, while the good speed-up with the parallel smoothing method was limited to a small number of processors.
In [6] a time-parallel version of the SIMPLE algorithm ([23]) is used to solve incompressible Navier-Stokes equations on a transputer system. Good efficiency is obtained for an unsteady driven cavity problem. In [18] a vorticity-velocity formulation is applied to the incompressible Navier-Stokes equations. The coarse grid equation is set up a little bit differently from [15]. The smoother is a so-called Group Explicit Iterative method (GEI), and good convergence and CPU time results for a different number of processors (varying from 1 to 4) is presented for the steady driven cavity problem solved with unsteady equations.

In this paper the incompressible Navier-Stokes equations in general coordinates will be smoothed in the time-parallel multigrid method with the sequential smoother SCAL as smoothing method on all time levels. In general coordinate systems there is a wide choice in time-dependent test problems. The computer code is restricted at present to time-dependent Dirichlet problems. In these time-parallel methods (21) or (22) can not be used to obtain a starting solution. A good initial approximation on each time level will be produced with "nested iteration" (starting on the coarsest grid). Implemented are nested iteration V- and F-cycles.

4 The multigrid waveform relaxation method.

A smoothing algorithm is the most time consuming part of a multigrid method, and it will be interesting to execute this part efficiently on a parallel machine. With a waveform relaxation method communication costs are probably lower than for the other relaxation schemes. Waveform relaxation methods update an unknown in a grid-point along a whole time interval. If an unknown in space is assigned to a processor, then during the smoothing of that unknown in time there is no need for a lot of communication with other processors, which can be costly. Originally they were developed as relaxation scheme in electrical simulation techniques ([35]). Waveform relaxation methods are found to have qualitatively the same convergence behaviour as basic iterative methods. High frequency errors are smoothed quickly, while low frequency errors are damped slowly. Therefore waveform relaxation methods are also suited for a multigrid acceleration. The non-linear multigrid waveform algorithm differs from the non-linear time-parallel method, presented in the previous section only in the smoothing algorithm. Again the restriction and prolongation operators, which are spatial operators (no coarsening is applied to time steps), can be done in parallel. In [28] the multigrid algorithm is found to perform very well on several nonlinear initial boundary value and time-periodic parabolic partial differential equations on a parallel machine. The smoother used is a red/black Gauss-Seidel waveform smoother. A comparison between the standard time marching scheme and the multigrid waveform method even on a single processor shows a competitive performance for the multigrid waveform method in many cases considered ([29]). A multigrid waveform relaxation method to solve the incompressible Navier-Stokes equations has not been found in literature yet. Here it is presented with the SCAL relaxation method as smoother:
First all horizontal "white" rows are updated (which can be done in parallel on a parallel machine) on all time levels t_0,...,t_{end}. Then all horizontal "black" rows are updated on all time levels. After a waveform sweep along horizontal rows a sweep along vertical rows will be applied.
For an implicit Euler scheme the waveform smoother looks like:
Waveform smoothing algorithm:

begin
for spatial indices $i \in G$ **do:**
for number of iterations $\nu = 1$ step 1 until ν_{max} **do :**
for time indices $n = 1$ step 1 until n_{max} **do :**

$$(\left(\frac{I}{\Delta t} - M_h\right) \boldsymbol{u}^{n+1,\nu})_i = (-N_h \boldsymbol{u}^{n+1,\nu-1})_i + (\frac{I}{\Delta t}\boldsymbol{u}^{n,\nu})_i + f_i^{n+1}$$

enddo
end waveform smoothing.

A disadvantage of waveform methods is that they require extra storage of unknowns and operators. A remedy to avoid too much storage is to use small values of n_{max}.

5 Results.

An analytical test problem. In order to investigate the accuracy of time integration schemes it is useful to construct a test problem for which the analytical solution $\boldsymbol{u}(t) = (u, v, p)$ is known. The error $\Delta \boldsymbol{u}$ is defined by:

$$\Delta \boldsymbol{u} = \boldsymbol{u}(t) - \boldsymbol{u}_h^{(n)} \tag{30}$$

The contravariant vector V^α found after n time integration steps on a domain with spatial grid-size h is transformed to the Cartesian vector $\boldsymbol{u}_h^{(n)}$. Assuming density ρ and viscosity μ constant an exact solution of the incompressible Navier-Stokes equations is defined by:

$$\begin{aligned} u &= sin(t)sin(x)sin(y) \\ v &= sin(t)cos(x)cos(y) \\ p &= sin(t)(sin(x) + cos(y)) \end{aligned} \tag{31}$$

Substitution of (31) in (2) defines the right-hand side F^α of (2). Further u and v from (31) are defined as Dirichlet boundary conditions. We choose $\rho = 1$, $\mu = 0.02$. Streamlines and isobars of the solution are presented in Figure 3. A square domain $(0, \pi) \times (0, \pi)$ is discretized

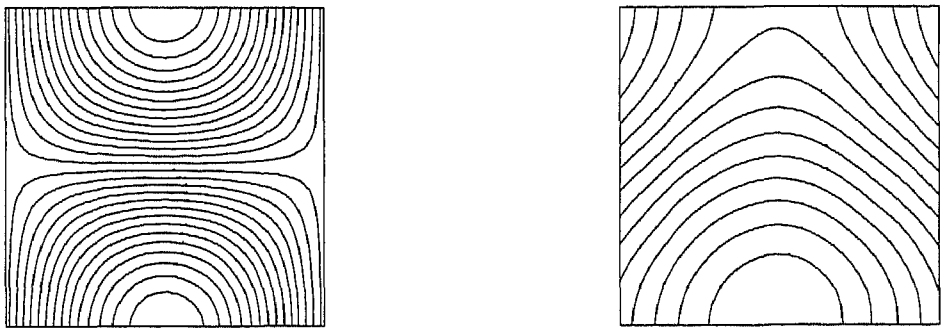

Figure 3: Streamlines an isobars of the solution of the analytical test example.

with 80×80 cells. The corresponding Reynolds number based on the maximum velocity is 50π. The l_2-norm of the u-error component is defined by:

$$||\Delta u||_{l_2} = \sqrt{\sum_{i=1}^{m} \frac{1}{m} (u_i - u_{i,h}^{(n)})^2} \tag{32}$$

where m represents the number of variable unknowns. A similar expression is found for $||\Delta v||$. $||\Delta p||$ is defined by:

$$||\Delta p|| = \sqrt{\sum_{i=1}^{m} \frac{1}{m}\{(p_i - \sum_{i=1}^{m} \frac{p_i}{m}) - (p_{i,h}^{(n)} - \sum_{i=1}^{m} \frac{p_{i,h}^{(n)}}{m})\}^2} \tag{33}$$

The determination of the error in the pressure (taking the difference between the solution and the mean value of the pressure) avoids extra errors coming from the fact that the pressure is determined up to a constant.
In Table 1 the error in the time discretization found with the time marching scheme (very similar results are found with the other multigrid schemes, of course) is presented for three schemes, the first order accurate implicit Euler scheme, the second order accurate Crank-Nicolson scheme and the second order accurate BDF(2) scheme. BDF(2) is started with an implicit Euler step. Time discretization is performed from $t_0 = 0$ to $t_{end} = 1.5$ with respectively 5,10, 20 and 40 time steps ($dt = 0.3, ..., 0.0375$). From Table 1 we can observe

Table 1: The l_2-norm of the errors for the test problem on a 80×80-grid, with (n) the number of time-steps between $t_0 = 0$ and $t_{end} = 1.5$.

(n)	Δu	Schemes		
		$\theta = 1$	$\theta = \frac{1}{2}$	BDF(2)
5	Δu	$.1337 \times 10^{-2}$	$.1099 \times 10^{-3}$	$.2940 \times 10^{-3}$
	Δv	$.7069 \times 10^{-3}$	$.6332 \times 10^{-4}$	$.1604 \times 10^{-3}$
	Δp	$.8332 \times 10^{-1}$	$.1169 \times 10^{-1}$	$.7574 \times 10^{-2}$
10	Δu	$.6777 \times 10^{-3}$	$.2268 \times 10^{-4}$	$.7048 \times 10^{-4}$
	Δv	$.3562 \times 10^{-3}$	$.1365 \times 10^{-4}$	$.3919 \times 10^{-4}$
	Δp	$.4213 \times 10^{-1}$	$.2398 \times 10^{-2}$	$.1666 \times 10^{-2}$
20	Δu	$.3354 \times 10^{-3}$	$.6075 \times 10^{-5}$	$.1772 \times 10^{-4}$
	Δv	$.1815 \times 10^{-3}$	$.4457 \times 10^{-5}$	$.1065 \times 10^{-4}$
	Δp	$.2238 \times 10^{-1}$	$.6297 \times 10^{-3}$	$.4989 \times 10^{-3}$
40	Δu	$.1719 \times 10^{-3}$	$.3035 \times 10^{-5}$	$.5520 \times 10^{-5}$
	Δv	$.9018 \times 10^{-4}$	$.2859 \times 10^{-5}$	$.4106 \times 10^{-5}$
	Δp	$.1067 \times 10^{-1}$	$.3752 \times 10^{-3}$	$.2907 \times 10^{-3}$

that, before the spatial discretization error is reached (for $n = 40$) the different schemes are indeed first and second order accurate in time (the error is a factor 4 smaller for $n = 10$ compared to $n = 5$ for BDF(2) and $\theta = \frac{1}{2}$). Because BDF(2) shows results similar to the Crank-Nicolson scheme, and because BDF(2) is an L-stable scheme, while the Crank-Nicolson scheme is A-stable but not L-stable ([12]) we will continue with the first order Euler scheme and the second order BDF(2)-scheme. (BDF(2) requires extra storage of one solution from an previous time level, but that is not a problem here, because the parabolic and waveform relaxation multigrid time iteration schemes also need these solutions to be stored.) The average number of iterations per time-step to reach the desired accuracy from (20) on the 80×80-grid for the BDF(2) scheme is presented in Table 2. The sequential computational complexity of the marching, parabolic and waveform schemes is about the same. For the time marching scheme we compared the V(0,1)-cycle (meaning V-cycle with 0 pre- and 1 post-smoothing iteration) with V(0,2), F(0,1) and F(0,2). V(0,1) appeared to be the fastest cycle, followed

by V(0,2). The F-cycles are much more expensive and therefore not taken into account here for the other multigrid schemes. For the parabolic and waveform multigrid algorithms we compared V(0,1) and V(0,2) with FV(0,1) and FV(0,2) (meaning a V-cycle scheme, with an F-cycle nested iteration). Again for this test problem the CPU time for the V(0,1)-cycle was smallest, followed by FV(0,1) and V(0,2). From Table 2 we also observe a reduction, due to the good starting solution on each time level in the number of iterations for the time marching scheme when the number of time levels equals 20.
Clearly the smoothing algorithm is though robust certainly fairly expensive on a vector computer. An iteration of the time marching scheme indeed was as expensive as an iteration of the other schemes. It is satisfying to observe that a more or less equal number of iter-

Table 2: The average number of iterations per time-step to satisfy the termination criterion for different multigrid schemes (- = not considered).

(n)	cycle	BDF(2)		
		Marching	Parabolic	Waveform
5	V(0,1)	5	5	5.5
	V(0,2)	4	4	4
	F(0,1)	4.5	-	-
	F(0,2)	4	-	-
	FV(0,1)	-	4	4
	FV(0,2)	-	2	2.5
10	V(0,1)	5	5	5.5
	V(0,2)	4	4	4
	F(0,1)	4	-	-
	F(0,2)	3.5	-	-
	FV(0,1)	-	4	4
	FV(0,2)	-	2	2.5
20	V(0,1)	4	5	5.5
	V(0,2)	3	3	3.5
	F(0,1)	3	-	-
	F(0,2)	2.5	-	-
	FV(0,1)	-	4	4.5
	FV(0,2)	-	2	2.5

ations is needed to satisfy the termination criterion for the parabolic multigrid scheme and the multigrid waveform relaxation scheme. Also approximations of the convergence factor are investigated:

$$\mu_n^{(i)} \equiv \| \frac{res_n^{(i)}}{res_{n-2}^{(i)}} \|^{1/2} \quad \text{for large n} \tag{34}$$

For i always the last time level is chosen. There the approximate convergence factor was generally not smaller than for earlier time-steps. In many cases $\mu_n^{(i)}$ is found to be approximately constant (for $n \cong 20$), in which case we have found the asymptotic convergence factor. The number of grid points are chosen to be 16×16, 32×32 and 64×64, the number of time levels is 40. Table 3 presents the approximate convergence factors. Again the multgrid waveform relaxation method shows a convergence factor for a large number of time-steps (40 is this

case) almost as good as the other two methods.

An unsteady skewed driven cavity. Another test example used here is the unsteady flow in a driven cavity, which has the shape of a parallelogram , with skewness angle 45^o. Investigations and reference results for the cavity studied here are described for the steady case in [8] and [21]. First we show some possibilities of our approach. An unsteady cavity flow at Reynolds number 1000 is presented on a 64×64-grid. The top wall will move with a speed depending on the time as follows:

$$\begin{aligned} t \leq 100: \quad & u = 1 \\ 100 < t < 200: \quad & u = -cos(\frac{\pi t}{100}) \\ t \geq 200: \quad & u = -1 \end{aligned} \tag{35}$$

Figure 4 shows streamlines of flowpatterns at certain times from one steady state to an other obtained with the BDF(2)-scheme. It is clear that most phenomena will happen from $t = 150$ to $t = 160$, when the flow direction is reversed. We chose variable time steps: $dt^{(1)} = 100$, $dt^{(i)} = 10$ for $i = 2,..,6$ (to $t = 150$); $dt^{(i)} = 1$ for $i = 7,...16$ (to $t = 160$) and $dt^{(i)} = 10$ for $i > 16$.

The test problem investigated numerically is described in [16]. The topwall of the skewed cavity is moving with velocity $u = sin(t)$ from $t_0 = 0$ to $t_{end} = 1.5$ ($\cong \pi/2$). The number of time steps is 10, 20 and 40. For this example the behaviour of an F-cycle for a higher Reynolds number is investigated: $Re = 1000$. We obtained the average number of iterations for an F(0,1)-cycle with starting solution from (21), and for the other schemes for an FF(0,1)-cycle to satisfy termination criterion (20) and the convergence factors from (34) with $n = 20$. Table 4 presents these results. The same results are found for the implicit Euler scheme.

6 Conclusions.

Three multigrid methods for the time-dependent incompressible Navier-Stokes equations equations in general coordinates have been compared for several test problems, namely a time marching scheme, a parabolic multigrid method and a multigrid waveform relaxation method. For all methods an essential part of the algorithm, the smoother, was based on the same method. Contrary to solving steady incompressible Navier-Stokes equations no additional underrelaxation is required to solve the unsteady equations. The smoother is a robust method, though fairly expensive (on a vector computer). Three time discretization schemes have been compared, the implicit Euler scheme, the Crank-Nicolson scheme and the BDF(2) scheme. The latter scheme showed second order accuracy, while it satisfies better stability conditions than the Crank-Nicolson scheme.

Further the storage requirement of the time marching scheme is lowest, for the multigrid waveform relaxation method it is highest. Approximate convergence factors and the average number of iterations per time-step to satisfy a termination criterion are compared. Satisfying results were obtained for all three methods. Their sequential computational complexity was about the same, but their differ markedly in their parallelization potential. It will be interesting to study the behaviour of the smoothers on a parallel machine.

References

[1] Ch. Arakawa, A. Demuren, W. Rodi, B. Schönung, *Application of multigrid methods for the coupled and decoupled solution of the incompressible Navier-Stokes equations.* In: M.

Deville (ed.), Proc. 7th GAMM conf. on Num. Methods in Fluid. Mech., Notes on Num. Fluid Mech. **20,** 1–8, Vieweg, Braunsweig, (1988).

[2] R. Aris, *Vectors, tensors and the basic equations of fluid mechanics.* Prentice-Hall, Inc., Englewood Cliffs, N.J. (1962).

[3] P. Bastian, J. Burmeister and G. Horton, *Implementation of a parallel multigrid method for parabolic differential equations.* In: W. Hackbusch (ed.), *Parallel algorithms for partial differential equations*, Proc. 6th GAMM seminar Kiel, Vieweg, Wiesbaden, (1990).

[4] A. Brandt, Guide to multigrid development. *In:* W. Hackbusch and U. Trottenberg (eds.), *Multigrid Methods*, Lecture notes in Mathematics **960**, pp. 220–312, Springer, Berlin (1982).

[5] J. Burmeister, *Paralleles Lösen diskreter parabolischer Probleme mit Mehrgittertechniken* Master's Thesis, Kiel, (1985).

[6] J. Burmeister and G. Horton, *Time-parallel multigrid solution of the Navier-Stokes equations.* In: Multigrid methods III, W. Hackbusch and U. Trottenberg (eds.), pages 155–166. International Series of Numerical Mathematics **98**, Birkhäuser, Basel, (1991).

[7] A.J. Chorin, A numerical method for solving incompressible viscous flow problems. *J. Comp. Phys.* **2,** 12–26, (1967).

[8] I. Demirdzic, Z. Lilek and M. Peric, Fluid flow and heat transfer test problems for non-orthogonal grids: Bench-mark solutions. *Int. J. Num. Methods in Fluids* **15**, 329–354, (1992).

[9] C.W. Gear, *Numerical initial value problems for ordinary differential equations.* Prentice Hall, Englewood Cliffs, (1971).

[10] W. Hackbusch, *Parabolic Multigrid Methods.* In: R. Glowinski, J.R. Lions (eds.) *Computing methods in Applied Sciences and Engineering VI,* Proc. 6th Int. Symp. on Comp. Meth. in Appl. Sciences and Eng. 20–45, North Holland, Amsterdam, (1984).

[11] W. Hackbusch, *Multi-grid methods and applications.* Springer-Verlag, Berlin (1985).

[12] E. Hairer, G. Wanner, *Solving ordinary differential equations II,* Springer, Berlijn (1991).

[13] F. Harlow, J. Welch, Numerical calculation of time-dependent viscous incompressible flow. *Phys. Fluids* **8,** 2182–2189 (1965).

[14] P.W. Hemker Mixed defect correction iteration for the accurate solution of the convection diffusion equation. *In:* W. Hackbusch and U. Trottenberg (eds.), *Multigrid Methods*, Lecture notes in Mathematics **960**, pp. 485–501, Springer, Berlin (1982).

[15] G. Horton, *Time-parallel multigrid solution of the Navier-Stokes equations.* In: C. Brebbia, ed. *Applications of supercomputers in Engineering.* Elsevier, Amsterdam, (1991).

[16] G. Horton, The time-parallel multigrid method. *Comm. Appl. Num. Methods* **8,** 585–596 (1992)

[17] D.S. Joshi and S.P.Vanka, Multigrid calculation procedure for internal flows in complex geometries. *Num. Heat Transfer* **20,** 61–80, (1991).

[18] S. Murata, N. Satofuka and T. Kushiyama, Parabolic Multi-grid Method for incompressible viscous flows using a group explicit relaxation scheme. *Comp. and Fluids*, **19**, 33–41, (1991).

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

[20] C.W. Oosterlee and P. Wesseling, A multigrid method for an invariant formulation of the incompressible Navier-Stokes equations in general coordinates. *Comm. Appl. Num. Methods* **8**, 721–734 (1992)

[21] C.W. Oosterlee and P. Wesseling, *A robust multigrid method for a discretization of the incompressible Navier-Stokes equations in general coordinates* Report 92/14, TU Delft, Fac. Math. Inf., Delft (1992). To appear in Impact of Comp. in Science and Eng.

[22] C.W. Oosterlee and P. Wesseling, *Benchmark solutions for the incompressible Navier-Stokes equations in general coordinates on staggered grids.* Report 92/67, TU Delft, Fac. Math. Inf., Delft (1992). Submitted to Int. J. Num. Methods in Fluids.

[23] S. V. Patankar 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).

[24] S.V. Patankar, *Numerical heat transfer and fluid flow.* McGraw-Hill, New York (1980).

[25] A. Segal, P. Wesseling, J. van Kan, C.W. Oosterlee and C.G.M. Kassels, Invariant discretization of the incompressible Navier-Stokes equations in boundary fitted co-ordinates. *Int. J. Num. Methods in Fluids* **15**, 411–426 (1992).

[26] W.Y. Soh and J.W. Goodrich, Unsteady solution of incompressible Navier-Stokes equations. *J. Comp. Phys.* **79**, 113–134, (1988).

[27] M. C. Thompson and J. H. Ferziger, An adaptive multigrid technique for the incompressible Navier- Stokes equations. *J. Comp. Phys.* **82**, 94–121 (1989).

[28] S. Vandewalle, *The parallel solution of parabolic partial differential equations by Multigrid waveform relaxation methods.* Ph. D. Thesis, Leuven University, (1992).

[29] S. Vandewalle, R. Piessens, Efficient parallel algorithms for solving initial-boundary value and time-periodic parabolic partial differential equations. *SIAM J. Sci. Stat. Comp.*, **13**, 1330–1346, (1992).

[30] S.P. Vanka, Block-implicit calculation of steady turbulent recirculating flows. *Int. J. Heat Mass Transfer* **28**, 2093–2103 (1985).

[31] S.P. Vanka, Block-implicit multigrid solution of Navier-Stokes equations in primitive variables. *Journ. Comp. Physics* **65**, 138–158 (1986).

[32] J.J.I.M. van Kan, A second-order accurate pressure correction method for viscous incompressible flow. *SIAM J. Sci. Stat. Comp.* **7**, 870–891, (1986).

[33] C. Vuik, *Solution of the discretized incompressible Navier-Stokes equations with the GMRES method* Report 91/24 , TU Delft, Fac. Math & Inf. Delft (1991). To appear in Int. J. Num. Meth. Fluids.

[34] P. Wesseling, *An introduction to multigrid methods.* John Wiley, Chichester (1992).

[35] J. White and A. Sangiovanni-Vincentelli, *Relaxation techniques for the simulation of VLSI Circuits.* Kluwer Academic Publishers, Boston, (1987).

[36] G. Wittum, Multi-grid methods for Stokes and Navier-Stokes equations with transforming smoothers: algorithms and numerical results. *Numer. Math.* **54**, 543–563 (1989).

[37] Zeng Shi and P. Wesseling, *Numerical solution of a bifurcation problem for the Boussinesq equations at low Prandtl number by a multigrid method.* Report 89/67, TU Delft, Fac. Math & Inf. Delft, (1989).