INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 22, 549–565 (1996)

A PARALLEL BLOCK-STRUCTURED MULTIGRID METHOD FOR THE PREDICTION OF INCOMPRESSIBLE FLOWS

F. DURST AND M. SCHÄFER

Lehrstuhl für Strömungsmechanik, Universität Erlangen-Nürnberg, Cauerstr. 4, D-91058 Erlangen, Germany

SUMMARY

In this paper a parallel multigrid finite volume solver for the prediction of steady and unsteady flows in complex geometries is presented. For the handling of the complexity of the geometry and for the parallelization a unified approach connected with the concept of block-structured grids is employed. The parallel implementation is based on grid partitioning with automatic load balancing and follows the message-passing concept, ensuring a high degree of portability. A high numerical efficiency is obtained by a non-linear multigrid method with a pressure correction scheme as smoother.

By a number of numerical experiments on various parallel computers the method is investigated with respect to its numerical and parallel efficiency. The results illustrate that the high performance of the underlying sequential multigrid algorithm can largely be retained in the parallel implementation and that the proposed method is well suited for solving complex flow problems on parallel computers with high efficiency.

KEY WORDS: parallel computing; multigrid method; finite volume method; block-structured grids; incompressible flow

1. INTRODUCTION

The numerical simulation of practically relevant flows often involves the handling of complex geometries and complex physical and chemical phenomena requiring the use of very fine grids and small time steps in order to achieve the necessary numerical accuracy. In recent years intensive research has been undertaken to improve the performance of flow computations in order to enlarge their applicability for a cost-effective solution of practically relevant flow problems. These improvements concern both acceleration by the use of more efficient solution algorithms such as multigrid methods (see e.g. References 1–3) as well as acceleration by the use of more efficient computer hardware such as high-performance parallel computers (see e.g. References 4 and 5). In this paper a numerical solution method for the incompressible Navier–Stokes equations is presented which combines efficient numerical techniques and parallel computing. A similar procedure for steady flows in simple orthogonal geometries is given by Perić and Schreck.[6]

The underlying numerical scheme is based on a procedure described by Perić,[7] consisting of a fully conservative second-order finite volume space discretization with a collocated arrangement of variables on non-orthogonal grids, a pressure correction method of the SIMPLE type for the iterative coupling of velocity and pressure and an iterative ILU decomposition method for the solution of the sparse linear systems for velocity components, pressure correction and temperature. For time discretization an implicit second-order scheme is employed, while a non-linear multigrid scheme, in which the pressure correction method acts as a smoother on the different grid levels, is used for convergence acceleration.

CCC 0271–2091/96/060549-17

Received May 1995
Revised June 1995

For the treatment of complex geometries and as a basis for the parallelization of the method by means of a grid-partitioning technique, the concept of block-structured grids is used. The block-structuring approach is advantageous for several reasons.

1. Complex geometries can be modelled easily.
2. Numerically efficient 'structured' algorithms can be used within each block.
3. A natural basis for the parallelization of the solution methods is provided.
4. Regions with different material properties (e.g. solid/liquid problems) or problems requiring various grid systems moving against each other can be handled in a straightforward way.

The block-structuring approach can be viewed as a compromise between flexible unstructured grids and numerically efficient structured grids.

The parallelization of the method is achieved by a grid-partitioning technique based on the block-structured grids. Depending on the number of processors, the block structure suggested by the geometry is restructured by an automatic load-balancing procedure such that the resulting subdomains can be assigned suitably to the individual processors. The major objective of the parallelization strategy was to preserve the high numerical efficiency of the sequential method in the parallel implementation. Therefore, in particular, the parallel multigrid method is implemented globally, i.e. without being affected by the grid partitioning. This ensures a close coupling of the subdomains and only a slight deterioration of the numerical efficiency compared with the corresponding sequential algorithm can be observed.

By a variety of numerical experiments for steady and unsteady flows the performance of our parallel algorithm is studied. These include investigations of the interdependence of the grid size, the time step size, the number of processors, the multigrid algorithm, the grid partitioning and the performance data of the parallel computer with respect to numerical and parallel efficiency. The results indicate that parallel computers combined with advanced numerical methods can yield the computational performance required for an efficient, accurate and reliable solution of practical flow problems in engineering and science.

We remark that the employed numerical technique has already been applied successfully to a variety of practical flow problems (see e.g. References 8–11). In this paper we concentrate more on the methodological aspects of the method to illustrate the intrinsic properties of the employed solution techniques. We restrict ourselves to two-dimensional laminar incompressible flow problems, but mention that a generalization of the underlying concepts to three-dimensional, turbulent and/or compressible flow is straightforward (indications with respect to the latter are given e.g. by Demirdžić *et al.*[12]).

2. GOVERNING EQUATIONS AND DISCRETIZATION

We consider the laminar non-isothermal time-dependent flow of an incompressible Newtonian fluid in an arbitrary two-dimensional domain. The basic conservation equations governing transport of mass, momentum and energy are given by

$$\frac{\partial(\rho v_j)}{\partial x_j} = 0, \tag{1}$$

$$\frac{\partial(\rho v_i)}{\partial t} + \frac{\partial}{\partial x_j}\left(\rho v_j v_i - \mu \frac{\partial v_i}{\partial x_j}\right) + \frac{\partial p}{\partial x_i} - \frac{\partial}{\partial x_j}\left[\mu\left(\frac{\partial v_j}{\partial x_i}\right)\right] = \rho_0 \beta (T - T_0) g_i, \tag{2}$$

$$\frac{\partial(\rho T)}{\partial t} + \frac{\partial}{\partial x_j}\left(\rho v_j T - \frac{\mu}{Pr}\frac{\partial T}{\partial x_j}\right) = 0, \tag{3}$$

where $v=(v_1, v_2)$ is the velocity vector with respect to Cartesian co-ordinates (x_1, x_2), t is the time, ρ is the density, μ is the dynamic viscosity, Pr is the Prandtl number, p is the pressure, T is the temperature, $g=(g_1, g_2)$ is the gravitational acceleration vector, β is the coefficient of thermal expansion and T_0 is a reference temperature at which μ, Pr and the reference density ρ_0 are defined. Within the above formulation the Boussinesq approximation is assumed (see e.g. Reference 13), which in no way is crucial for the derivation of the method below. It is only chosen here because of its ease of presentation.

In order to apply the above system of partial differential equations, it has to be completed with some boundary and initial conditions for v and T actually defining the flow problem. The boundary and initial conditions for the pressure p are then already determined and must not be additionally specified.

2.1. Spatial and temporal discretization

For the spatial discretization of (1)–(3) a finite volume method with a collocated arrangement of variables is employed. The basic procedure is described in detail by Demirdžić and Perić,[14] so only a brief summary is given here.

The solution domain is discretized into quadrilateral (in general non-orthogonal) finite volume cells. The transport equations (1)–(3) are then integrated over these control volumes (CVs), leading, after the application of the Gauss theorem, to a balance equation for the fluxes through the CV faces and the volumetric sources. The convection and diffusion contributions to the fluxes are evaluated using a central differencing scheme, which for the convective part is implemented using the deferred correction approach proposed by Khosla and Rubin.[15] The evaluation of the volumetric sources is performed by approximating the source term by its value at the centre of the CV.

For the time discretization the so-called θ-method (see e.g. Reference 16) is employed. Applying the θ-method to the system of ordinary differential equations resulting from the spatial discretization, approximations v_h^n, p_h^n and T_h^n to the solution of (1)–(3) at the time level $t_n=n\Delta t$ $(n=1, 2, \ldots)$ are defined as solutions of non-linear algebraic systems of the form

$$\mathbf{L}_h v_h^n = 0, \tag{4}$$

$$v_h^n + \theta\Delta t[\mathbf{A}_h(v_h^n)v_h^n + \mathbf{G}_h p_h^n + \mathbf{F}_h T_h^n] = v_h^{n-1} + (\theta-1)\Delta t[\mathbf{A}_h(v_h^{n-1})v_h^{n-1} + \mathbf{G}_h p_h^{n-1} + \mathbf{F}_h T_h^{n-1}], \tag{5}$$

$$T_h^n + \theta\Delta t\mathbf{B}_h(v_h^n)T_h^n = T_h^{n-1} + (\theta-1)\Delta t\mathbf{B}_h(v_h^{n-1})T_h^{n-1}. \tag{6}$$

The discrete operators $\mathbf{A}_h$, $\mathbf{B}_h$, $\mathbf{G}_h$, $\mathbf{F}_h$ and $\mathbf{L}_h$ are defined according to the spatial discretization described above, including the corresponding discretization of the boundary conditions. The parameter h is a measure of the spatial resolution (e.g. the maximum CV diameter) and $\Delta t>0$ is the time step. The parameter $\theta\in(0, 1]$ is a blending factor for the explicit and implicit contributions of the time discretization. We remark that the cases $\theta=1$ and 0·5 correspond to the first-order fully implicit Euler scheme and the second-order implicit Crank–Nicolson scheme respectively. Both methods are unconditionally stable, but it is well known that for spatially non-smooth solutions the Crank-Nicolson scheme may cause numerical oscillations (see e.g. Reference 16), while the implicit Euler scheme does not show such a behaviour (strong A-stability). Thus varying θ within the interval [0·5, 1] also gives the possibility to control the stability of the scheme.

The time-stepping process defined by (4)–(6) is started from the initial values

$$v_h^0 = v_{0h}, \qquad T_h^0 = T_{0h}, \qquad p_h^0 = 0, \tag{7}$$

where v_{0h} and T_{0h} are suitable spatial approximations of the initial values given for v and T. Assuming that v_h^{n-1}, p_h^{n-1} and T_h^{n-1} are already computed, we are faced with the problem of solving the non-linear system (4)–(6) for v_h^n, p_h^n and T_h^n. For this a non-linear full approximation multigrid scheme with a

pressure correction algorithm of the SIMPLE type as a smoother is employed. Before describing the multigrid technique, we will take a closer look at the smoother applied on the different grid levels.

2.2. *Pressure correction smoother*

The employed smoothing procedure for the coupled set of non-linear equations (4)–(6) is based on the well-known SIMPLE algorithm proposed by Patankar and Spalding.[17] Because various modifications to the original have been introduced and many variants exist in the literature, we describe the approach employed here in a little bit more detail. A similar procedure for steady problems is described by Perić *et al.*[18]

In the following, one iteration step $k-1 \rightarrow k$ is described, assuming that $v_h^{n,k-1}$, $p_h^{n,k-1}$ and $T_h^{n,k-1}$ are already computed. The determination of $v_h^{n,k}$, $p_h^{n,k}$ and $T_h^{n,k}$ is done in several steps, leading to a decoupling with respect to the different variables. We first introduce a splitting of $\mathbf{A}_h$ of the form

$$\mathbf{A}_h(v_h^{n,k-1}) = \mathbf{A}_{\mathrm{D}h}^{n,k-1} + \mathbf{A}_{\mathrm{O}h}^{n,k-1} + \mathbf{A}_{\mathrm{N}h}^{n,k-1} \tag{8}$$

into the diagonal part $\mathbf{A}_{\mathrm{D}h}^{n,k-1}$ and off-diagonal parts $\mathbf{A}_{\mathrm{O}h}^{n,k-1}$ and $\mathbf{A}_{\mathrm{N}h}^{n,k-1}$ corresponding to orthogonal and non-orthogonal (cross-derivative) contributions of the space discretization respectively. The splitting is illustrated in Figure 1, showing the assignment for a CV with its eight neighbours involved in the local discretization.

In a first step an intermediate approximation $v_h^{n,k-1/2}$ to $v_h^{v,k}$ is obtained by solving the momentum equation (5) with the pressure term, the temperature term and the matrix coefficients formed with values of the preceding iteration $k-1$, treating the term with $\mathbf{A}_{\mathrm{N}h}^{n,k-1}$ explicitly and introducing an underrelaxation:

$$\begin{aligned} v_h^{n,k-1/2} + \theta\Delta t(\mathbf{A}_{\mathrm{D}h}^{n,k-1} + \alpha_v \mathbf{A}_{\mathrm{O}h}^{n,k-1})v_h^{v,k-1/2} &= \alpha_v S_{vh}^{n-1} + (1-\alpha_v)(v_h^{n,k-1} + \theta\Delta t \mathbf{A}_{\mathrm{D}h}^{n,k-1} v_h^{n,k-1}) \\ &\quad - \alpha_v \theta\Delta t(\mathbf{A}_{\mathrm{N}h}^{n,k-1} v_h^{n,k-1} + \mathbf{G}_h p_h^{n,k-1} + \mathbf{F}_h T_h^{n,k-1}). \end{aligned} \tag{9}$$

Here the underrelaxation factor α_v is in the interval (0, 1] and, for abbreviation, in S_{vh}^{n-1} all terms of (5) containing only values from the preceding time level $n-1$ which are not affected by the iteration process are summarized:

$$S_h^{n-1} = v_h^{n-1} + (\theta - 1)\Delta t[\mathbf{A}_h(v_h^{n-1})v_h^{n-1}\mathbf{G}_h p_h^{n-1} + \mathbf{F}_h T_h^{n-1}]. \tag{10}$$

Following the spirit of a pressure correction approach, in a second step we are now looking for corrections $\tilde{p}_h^{n,k}$ and $\tilde{v}_h^{n,k}$ to obtain the new pressure $p_h^{n,k} = p_h^{n,k-1} + \tilde{p}_h^{n,k}$ and the new velocity $v_h^{n,k} = v_k^{n,k-1/2} + \tilde{v}_h^{n,k}$ exactly fulfilling the discrete continuity equation (4). For this a momentum equation for $v_h^{n,k}$ of the form

$$\begin{aligned} v_h^{n,k} + \theta\Delta t \mathbf{A}_{\mathrm{D}h}^{n,k-1} v_h^{n,k} &= (1-\alpha_v)(v_h^{n,k-1} + \theta\Delta t \mathbf{A}_{\mathrm{D}h}^{n,k-1} v_h^{v,k-1}) + \alpha_v S_{vh}^{n-1} \\ &\quad - \alpha_v \theta\Delta t(\mathbf{A}_{\mathrm{O}h}^{n,k-1} v_h^{n,k-1/2} + \mathbf{A}_{\mathrm{N}h}^{n,k-1} v_h^{n,k-1} + \mathbf{G}_h p_h^{n,k} + \mathbf{F}_h T_h^{n,k-1}) \end{aligned} \tag{11}$$

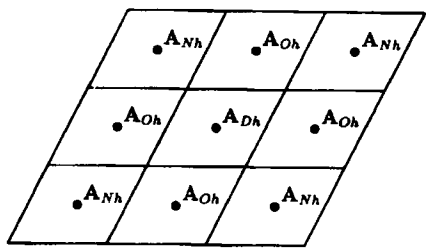

Figure 1. Splitting of the operator $\mathbf{A}_h$ into orthogonal and non-orthogonal off-diagonal parts and a diagonal part

is considered, which differs from the one for $v_h^{n,k-1/2}$, i.e. (9), by the treatment of the term with $\mathbf{A}_{Oh}^{n,k-1}$ and by taking the pressure term at the new iteration level k. Now, subtracting (9) from (11) yields

$$\tilde{v}_h^{n,k} + \theta\Delta t\bar{\mathbf{A}}_{Dh}^{n,k-1}\tilde{v}_h^{n,k} = -\alpha_v\theta\Delta t\bar{\mathbf{G}}_h\tilde{p}_h^{n,k}, \tag{12}$$

where the overbar on the operators indicates the selective interpolation technique used for making the cell face velocities dependent on the nodal pressures, which is necessary to avoid oscillatory solutions that may occur owing to the non-staggered grid arrangement.[19] Equation (12) represents an explicit expression for the new velocity in terms of the pressure correction. By applying the operator $\mathbf{L}_h$ to both sides of (12) and taking into account the validity of the continuity equation (4) for $v_h^{n,k}$, an equation for $\tilde{p}_h^{n,k}$ is obtained:

$$-\mathbf{L}_h\bar{\mathbf{G}}_h\tilde{p}_h^{n,k} = \frac{1}{\alpha_v\theta\Delta t}\mathbf{L}_h v_h^{n,k-1/2} + \frac{1}{\alpha_v}\bar{\mathbf{A}}_{Dh}^{n,k-1}\mathbf{L}_h v_h^{n,k-1/2}. \tag{13}$$

Equation (13) corresponds to a discrete Poisson equation with homogeneous Neumann boundary conditions for $\tilde{p}_h^{n,k}$. To keep the structure of the pressure correction equation the same as for the discrete momentum equation (9) and to improve its diagonal dominance, the non-orthogonal part of $\mathbf{C}_h := \mathbf{L}_h\bar{\mathbf{G}}_h$ is neglected (see e.g. Reference 20) and $\tilde{p}_h^{n,k}$ is computed from (13) with $\mathbf{C}_h$ replaced by $\mathbf{C}_{Dh} + \mathbf{C}_{Oh}$ using a splitting analogous to that in (8). If the grid is highly non-orthogonal, the influence of the neglected pressure cross-derivatives can be accounted for by the solution of a second pressure correction equation similar to that in the PISO algorithm to account for the influence of neglected velocity corrections.[21]

Once $\tilde{p}_h^{n,k}$ has been computed, $v_h^{n,k}$ can be easily obtained from (12). The validity of the continuity equation for $v_h^{n,k}$ follows directly from (12) and (13). For the new pressure also an underrelaxation with $0 < \alpha_p < 1$ is employed:

$$p_h^{n,k} = p_h^{n,k-1} + \alpha_p\tilde{p}_h^{n,k}. \tag{14}$$

Finally, the iteration step is completed by solving the energy equation

$$T_h^{n,k} + \theta\Delta t(\mathbf{B}_{Dh}^{n,k} + \alpha_T\mathbf{B}_{Oh}^{n,k})T_h^{n,k} = (1-\alpha_T)(T_h^{n,k-1} + \theta\Delta t\mathbf{B}_{Dh}^{n,k-1}T_h^{n,k-1}) + \alpha_T S_{Th}^{n-1} - \alpha_T - \Delta t\mathbf{B}_{Nh}^{n,k}T_h^{n,k-1} \tag{15}$$

for $T_h^{n,k}$, where S_{Th}^{n-1}, is defined analogously to S_{vh}^{n-1} in (10), the splitting of $\mathbf{B}_h(v_h^{n,k})$ is introduced according to that in (8) for $\mathbf{A}_h(v_h^{n,k})$ and $0 < \alpha_T \leqslant 1$ is the underrelaxation factor for the temperature.

The structure and solution of the linear systems of equations (9), (13) and (15) for the different unknowns are discussed in Section 3.2 within the framework of the employed parallelization technique. The initial approximations $v_h^{n,0}$, $p_h^{n,0}$ and $T_h^{n,0}$ for the iterative procedure are defined either as the values from the preceding time level or by interpolated values from the next coarser or finer grid within the multigrid procedure described in the next subsection.

Theoretical results concerning the convergence and smoothing properties of the considered pressure correction approach are discussed by Wittum[22] in the more general setting of transforming smoothers.

2.3. Multigrid method

The iterative pressure correction procedure described in the previous subsection removes efficiently only those Fourier components of the error whose wavelengths are comparable with the grid spacing, which usually results in a quadratic increase in computing time with the grid spacing. To keep this increase close to linear, a full approximation multigrid scheme is applied directly to the non-linear system (4)–(6), with the above pressure correction scheme acting as a smoother for the different grid levels.

To give a short summary of the multigrid technique, let us denote the system (4)–(6) on the grid with mesh size h by

$$\mathbf{K}_h(x_h) = b_h. \tag{16}$$

After a few iterations with the pressure correction smoother (pre-smoothing) an approximate solution $\tilde{x}_h$ to (16) is obtained:

$$\mathbf{K}_h(\tilde{x}_h) = b_h - r_h, \tag{17}$$

with r_h the residual. By linearization $\mathbf{K}_h$ via a Taylor expansion, the following non-linear equation for the error $e_h = x_h - \tilde{x}_h$ is obtained:

$$\mathbf{K}_h(\tilde{x}_h + e_h) - \mathbf{K}_h(\tilde{x}_h) = r_h. \tag{18}$$

The left-hand side of (18) is an approximation of the derivative of $\mathbf{K}_h$. Equation (18) is the basis for the coarse grid equation, which is defined by

$$\mathbf{K}_{2h}(\mathbf{I}_h^{2h}\tilde{x}_h + e_{2h}) - \mathbf{K}_{2h}(\mathbf{I}_h^{2h}\tilde{x}_h) = \mathbf{I}_h^{2h} r_h, \tag{19}$$

with $\mathbf{I}_h^{2h}$ a suitable restriction operator. Thus, for assembling the coarse grid equation, $\mathbf{K}_h$, $\tilde{x}_h$ and r_h have to be restricted to the coarse grid. As coarse grid variable $x_{2h} := \mathbf{I}_h^{2h}\tilde{x}_h + e_{2h}$ is used, such that the coarse grid problem reads

$$\mathbf{K}_{2h}(x_{2h}) = b_{2h}, \quad \text{with } b_{2h} = \mathbf{K}_{2h}(\mathbf{I}_h^{2h}\tilde{x}_h) + \mathbf{I}_h^{2h} r_h. \tag{20}$$

As initial value for the iterative solution of the coarse grid equation $\mathbf{I}_h^{2h}\tilde{x}_h$ is used. After having obtained an approximate solution $\tilde{x}_{2h}$ of the coarse grid problem, the error (only the error is smooth) is transferred to the fine grid by means of an interpolation operator $\mathbf{I}_{2h}^h$ and the fine grid solution is corrected:

$$x_h^* = \tilde{x}_h + \tilde{e}_h, \quad \text{with } \tilde{e}_h = \mathbf{I}_{2h}^h(\tilde{x}_{2h} - \mathbf{I}_h^{2h}\tilde{x}_h). \tag{21}$$

Additional smoothing steps (post-smoothing) are carried out to reduce high-frequency errors which may be caused by the interpolation.

It should be noted that x_{2h} is not the solution of the system which would result from a discretization of the continuous system (1)–(3) on the coarse grid, but is an approximation of the fine grid solution. In the case of convergence the coarse grid solution (where it is defined) is identical with the fine grid solution.

The extension of the described two-grid procedure to the multigrid procedure is straightforward by applying the concept recursively to (20). For the movement through the grid levels the well-known V-cycle strategy is employed, which in the case of steady problems is combined with the nested iteration technique (full multigrid) to improve the initial guesses on the finer grids (see e.g. Reference 23). The problem on the coarsest grid is not solved exactly, but is approximated by carrying out some pressure correction iterations. Since the employed pressure correction algorithm is not only a good smoother but also a good solver, this does not significantly deteriorate the rate of convergence of the multigrid procedure. For prolongation and restriction, bilinear interpolation is used. Some more details about the employed multigrid procedure can be found in the paper by Hortmann *et al.*[2]

3. BLOCK STRUCTURING AND PARALLELIZATION

Having discussed the general global solution procedure, we will now look in more detail at the employed block-structured grid approach, which is also the basis of the parallelization of the method.

3.1. Block-structured grid partitioning

Block-structured grids, which are globally unstructured but locally structured, can be viewed as a compromise between the high geometrical flexibility of fully unstructured grids and the high numerical efficiency achieved on globally structured grids. The important characteristic of block-structured grids is that the logical structure of the individual blocks, which geometrically are in general boundary-fitted non-orthogonal grids, is rectangular, such that a regular data structure can be used for the representation of the field variables and that efficient solvers available for such regular structures can be employed with the individual blocks.

The coupling of the blocks along the block interfaces, i.e. the transfer of information among neighbouring blocks, requires special attention, especially with respect to an efficient parallelization. To deal with this problem, along the block interfaces, auxiliary control volumes containing the corresponding boundary values of the neighbouring block are introduced. The different situations that can occur for the block connections and the corresponding interface handling are illustrated in Figure 2. To ensure the coupling of the subdomains, the boundary data in the auxiliary control volumes of neighbouring blocks have to be updated from time to time during the iterative algorithm. This will be discussed in more detail in the next sub-section. In addition to these local exchanges, some global information transfer of residuals is required for convergence checking.

For the parallelization a grid-partitioning technique directly related to the block structuring is employed. The idea is to transform the block structure which results from the requirements for modelling the geometry (geometrical block structure) by some suitable mapping process to a new block structure (parallel block structure), which, in addition to the geometrical ones, also meets the requirements for an efficient implementation on a parallel computer.

In general, depending on the number of geometrical blocks, M, and the number of available processors, P, for the mapping process two situations have to be distinguished.

1. If $M > P$, the geometrical blocks are suitably grouped together so as to have as many groups as processors and the resulting groups are assigned to the individual processors.
2. If $M < P$, the geometrical blocks are partitioned in order to obtain finally a block structure with as many blocks as processors and the resulting blocks are assigned to the individual processors.

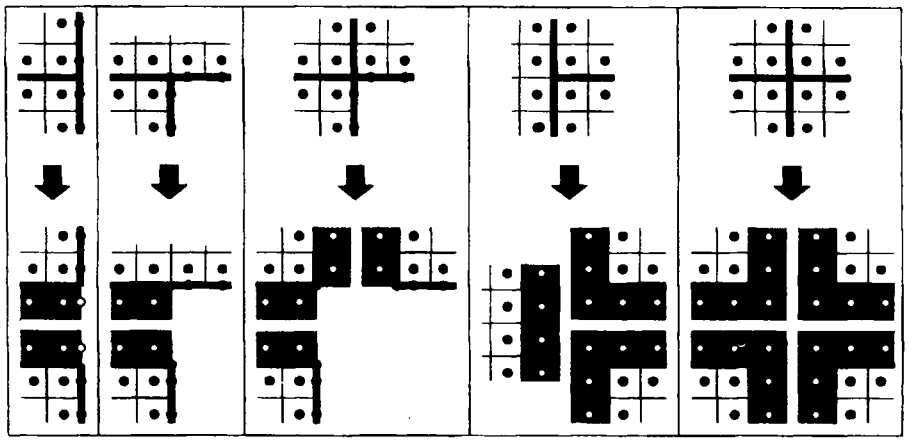

Figure 2. Block-structured grid connections and computational interface handling (the auxiliary CVs are in grey)

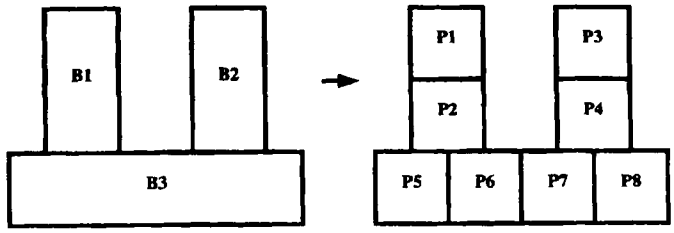

Figure 3. Example of mapping and grid partitioning when there are more processors ($P=8$) than blocks ($M=3$)

The two cases with the applied mapping and grid-partitioning strategy are illustrated schematically in Figures 3 and 4. In the case of $M=P$ no mapping has to be performed and the parallel block structure is taken to be identical with the geometrical one.

For the parallel block structure, several requirements with respect to obtaining an efficient parallel implementation can be formulated:

(a) similar number of control volumes per processor to ensure good load balancing on the parallel machine
(b) small number of neighbouring blocks located on other processors to have few communication processes
(c) short block interfaces along blocks located on different processors to have a small amount of data to transfer
(d) avoidance of steep flow gradients across block interfaces in order to retain good coupling of the subdomains for high numerical efficiency
(e) fully automatic mapping process working for arbitrary M and P which is not too time-consuming compared with the flow computation.

It is obvious that for general flow problems, as considered in this paper, not all of these requirements can be optimally fulfilled simultaneously and therefore some compromise has to be found. Our approach is mainly based on the first and last of the above criteria. Topological and flow-specific aspects are not taken into account. The automatic mapping is based on the simple criterion that the numbers of control volumes assigned to the different processors differ as little as possible.

In the case of $M > P$ the geometrical blocks are grouped into P subsets such that the numbers of CVs are similar in all subsets. This is done by taking the lexicographically first block and adding as many other blocks as long as the number of CVs in this first subset is smaller than N/P, where N is the total number of CVs. Then the next block is assigned to the next subset and the procedure is repeated successively until P subsets are formed. Of course, for $P=1$ this procedure also includes the serial case.

For the partitioning in the case of $M < P$ two different strategies are considered.

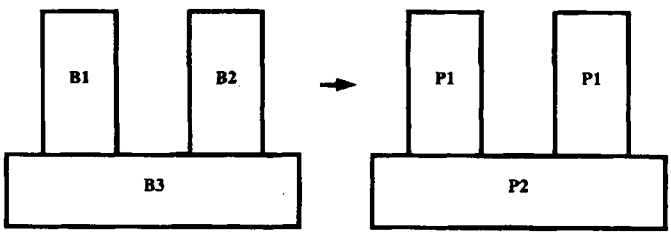

Figure 4. Example of mapping and grid partitioning when there are fewer processors ($P=2$) than blocks ($M=3$)

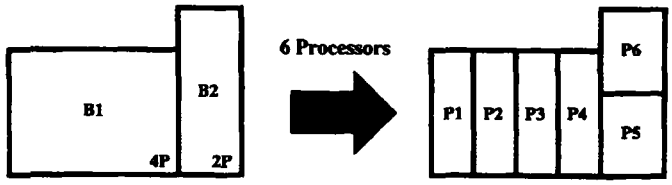

Figure 5. Automatic grid partitioning by direct mapping of geometrical block structure to parallel block structure

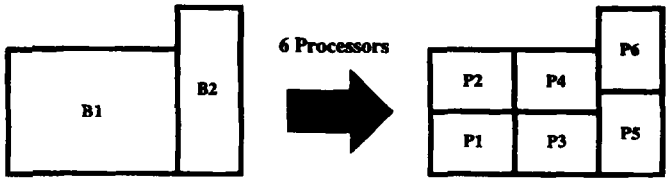

Figure 6. Automatic grid partitioning by recursive mapping of geometrical block structure to parallel block structure

1. *Direct decomposition.* The processors are assigned to the geometrical blocks according to the numbers of CVs in the blocks and the blocks are subdivided one-dimensionally in the co-ordinate direction with the largest number of CVs.
2. *Recursive decomposition.* The block with the largest number of CVs is halved in the direction with the largest number of CVs, resulting in a new block structure. The process is repeated until the number of blocks equals the number of processors.

The principles of the two strategies are illustrated schematically in Figures 5 and 6. Which strategy is preferable depends on the problem geometry, the block structure used to model it and the number of available processors. An advantage of the recursive decomposition is that, at least theoretically, as many processors as there are control volumes on the coarsest grid can be used for the parallel computation. If direct decomposition is used, the number of processors assigned to a block may not be larger than the largest number of control volumes in one direction in this block. (However, from the numerical point of view it is not reasonable to use too many processors for a grid of given size.) A comparison of the results of direct and recursive mapping can be found in the paper by Schäfer *et al.*[24]

Although the mapping approaches considered for the two cases are relatively simple, they are very fast and have turned out to ensure good load balancing for a wide range of applications.

3.2. Parallel linear system solver

In order to work efficiently on a parallel computer, the crucial point in the multigrid pressure correction method described in Section 2 is the choice of the numerical method for the solution of the sparse linear systems (9), (13) and (15) which arise during the iterative solution process for the different unknown variables. The parallelization of the other components of the method (assembly of the systems, prolongation, restriction, etc.) is straightforward. Since there are no recurrences and owing to the auxiliary CVs, these operations can be done (in principle) simultaneously for all CVs and locally on the individual processors.

For the discussion of the parallelization approach of the linear system solver, it is important to take a closer look at the structure of the systems, which is similar for all of them. Numbering the unknowns in a natural way according to the block structure, i.e. lexicographically within each block and then employing a block-by-block subsequential numbering, the coefficient matrix for all systems, which we denote commonly by $\mathbf{M}y = b$, gets a block-structure of the form

$$\mathbf{M} = \begin{pmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} & \cdot & \cdot & \cdot & \mathbf{B}_{1M} \\ \mathbf{B}_{12} & \mathbf{B}_{22} & \cdot & & & \cdot \\ \cdot & \cdot & \cdot & \cdot & & \cdot \\ \cdot & & \cdot & \cdot & \cdot & \cdot \\ \cdot & & & \cdot & \cdot & \mathbf{B}_{M-1,M} \\ \mathbf{B}_{M1} & \cdot & \cdot & \cdot & \mathbf{B}_{M,M-1} & \mathbf{B}_{MM} \end{pmatrix}, \tag{22}$$

where M is the number of blocks. The matrices $\mathbf{B}_{ii}$, $i = 1, \ldots, M$, on the diagonal have the usual pentadiagonal structure resulting from a five-point discretization in one subdomain (the non-orthogonal contributions are neglected or treated explicitly and contained in the right-hand sides). The matrices $\mathbf{B}_{ij}$, $i, j = 1, \ldots, M$, $i \neq j$, represent the coupling of the blocks. If block i is a neighbour to block j, these matrices have non-zero entries in the main diagonal, otherwise all entries vanish. According to this structure, $\mathbf{M}$ can be split into a local part

$$\mathbf{M}_{\mathrm{L}} = \begin{pmatrix} \mathbf{B}_{11} & & & \\ & \mathbf{B}_{12} & & 0 \\ & & \cdot & \\ & & & \cdot \\ & 0 & & \cdot \\ & & & \mathbf{B}_{MM} \end{pmatrix} \tag{23}$$

and a coupling part

$$\mathbf{M}_{\mathrm{C}} = \mathbf{M} - \mathbf{M}_{\mathrm{L}}. \tag{24}$$

In the present approach, as linear system solver, a parallel variant of the strongly implicit method of Stone[25] is employed. It is based on an incomplete LU decomposition and in its sequential version has proven to work very efficiently, especially in combination with multigrid techniques (see e.g. Reference 26). The serial version of the method is defined by an iteration process of the form

$$y^{n+1} = y^n - \mathbf{H}^{-1}(\mathbf{M}y^n - b), \tag{25}$$

with an incomplete decomposition $\mathbf{H} = \mathbf{LU}$ of $\mathbf{M}$ into lower and upper triangular matrices $\mathbf{L}$ and $\mathbf{U}$ respectively. For our parallel version of the method an incomplete decomposition of $\mathbf{M}_{\mathrm{L}}$ instead of $\mathbf{M}$ is used:

$$\mathbf{H} = \begin{pmatrix} \mathbf{L}_1\mathbf{U}_1 & & & \\ & \mathbf{L}_2\mathbf{U}_2 & & 0 \\ & & \cdot & \\ & & & \cdot \\ & 0 & & \cdot \\ & & & \mathbf{L}_M\mathbf{U}_M \end{pmatrix}. \tag{26}$$

Thus an iteration step

$$y_i^{n+1} = y_i^n - (\mathbf{L}_i\mathbf{U}_i)^{-1}\left(\sum_{j=1}^{M} \mathbf{B}_{ij}y_j^n - b_i\right) \tag{27}$$

can be carried out concurrently for all $i = 1, \ldots, M$, if y^n is available in the auxiliary CVs for computing $\mathbf{B}_{ij}y_j^n$ for $i \neq j$. To achieve this, one exchange of the block boundary values belonging to different processors is required in each iteration. For the local decompositions $\mathbf{L}_i\mathbf{U}_i$, $i = 1, \ldots, M$, in each block

the serial version of the strongly implicit method is applied. We remark that the present approach is closely related to the so-called multicoloured basic iterative methods discussed e.g. by Schäfer.[27]

4. NUMERICAL RESULTS

The following numerical investigations concerning the algorithmic aspects of the proposed method are performed on a Parsytec MultiCluster-3 transputer system with T805 processors under the operating system Parix1.2. How the performance of this system relates to more recent and more powerful systems can be seen from our study in Section 4.3, where different actual parallel computer platforms are involved.

4.1. Steady flow

First we study the properties of our solution method for a steady problem with respect to the interaction of the multigrid technique, the nested iteration and the parallelization. For this a buoyancy-driven flow in a cavity with a complex obstacle is considered. In Figure 7 the geometry with the numerical grid and the parallel block structure obtained from the direct mapping for 16 processors are shown together with the computed streamlines. The cavity walls are at low temperature and the obstacle walls are at high temperature, resulting in a Rayleigh number $Ra = 500$ based on the minimum distance between cavity wall and the obstacle. The Prandtl number is $Pr = 6{\cdot}7$. For the multigrid method a coarsest grid with 512 CVs is used (in Figure 7 the third level with 4096 CVs is shown) and V-cycles with five pre-smoothing, five post-smoothing and 10 coarse grid iterations are employed. The underrelaxation factors are 0·7 for velocities, 0·3 for pressure and 0·9 for temperature.

In Table I the computing times and the numbers of fine grid iterations are given for the single-grid and the multigrid method, each with and without nested iteration, for different numbers of processors and grid sizes. Within the two groups of results for $P = 1$, 4 and 16 processors and $P = 2$, 8 and 32 processors the number of control volumes per processor remains constant.

Several conclusions can be drawn from the results. In all cases the multigrid method is significantly superior to the corresponding single-grid computation. While the single-grid method shows its typical linear increase in iteration numbers with grid refinement, the iteration numbers for the multigrid method change only slightly. When the number of processors is increased by the same factor as the

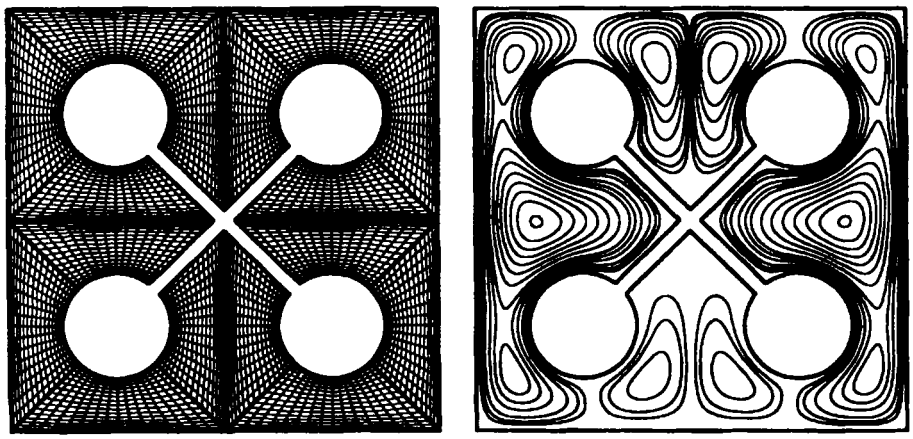

Figure 7. Geometry, intermediate grid (third level, 4096 CVs), parallel block structure (16 processors) and computed streamlines for buoyancy-driven flow in a cavity with a complex obstacle

Table I. CPU times in minutes and numbers of fine grid iterations (in parentheses) for single-grid (SG) and multigrid (MG) methods with and without nested iteration (NI) for different numbers of processors and control volumes for buoyancy-driven flow in a cavity with a complex obstacle

	N/P = 4096			N/P = 8192		
Method	$P=1$	$P=4$	$P=16$	$P=2$	$P=8$	$P=32$
SG	40 (171)	133 (592)	500 (2225)	265 (592)	977 (2215)	3929 (8619)
SG + NI	17 (58)	50 (201)	139 (556)	99 (201)	268 (550)	748 (1496)
MG	10 (31)	13 (41)	14 (42)	26 (41)	26 (42)	31 (46)
MG + NI	9 (21)	10 (24)	12 (26)	20 (24)	22 (26)	21 (21)

number of control volumes, the multigrid method gives the solution in nearly the same computing time, but owing to the increase in iteration numbers, this is not the case for the single-grid computations. If one compares for instance the results for $P=16$ and 8, which correspond to equal grid sizes, one can see that a very good parallel efficiency is obtained. We have speed-ups of 97·9% for SG, 96·4% for SG + NI, 92·8% for MG and 91·6% for MG + NI. The decrease in speed-up when adding NI and/or MG is due to an increase in work on coarser grids, where the proportion of communication relative to arithmetic operations is larger. The reduction of iteration numbers and computing times due to the nested iteration is larger for the single grid than for the multigrid case. The full multigrid method is faster than MG without nested iteration. Thus the additional effort due to the increased number of coarse grid computations is compensated by the reduction in the number of iterations.

4.2. *Unsteady flow*

For the investigation of the behaviour of our method for complex unsteady flows we consider as a second test problem the time-dependent natural convection in a square cavity with a circular obstacle. The cavity walls are at a fixed temperature and on the circular wall a time-dependent temperature is prescribed. The configuration with the corresponding boundary conditions is illustrated in Figure 8 together with the numerical grid used for the computation. The Prandtl number is $Pr=6{\cdot}7$ and the initial conditions at $t=0$ are $v_1=v_2=T=0$. The temporal flow behaviour can be seen from Figure 9, where the maximum vertical velocity is plotted against time. The quite complex flow structure is

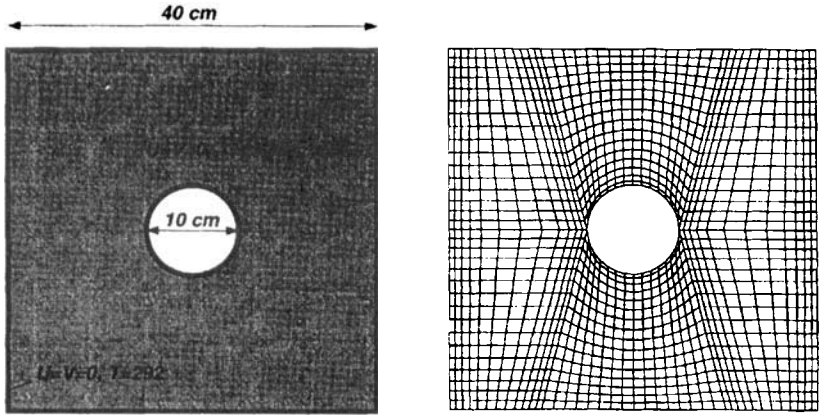

Figure 8. Configuration and boundary conditions for natural convection flow in a square cavity with a circular obstacle and numerical grid

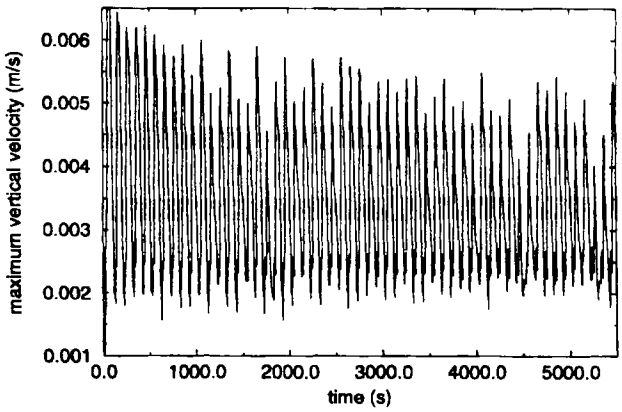

Figure 9. Maximum vertical velocity against time for natural convection flow in a square cavity with a circular obstacle

illustrated in Figure 10, showing the predicted streamlines for two points of time corresponding to the temporal maximum and minimum values of the maximum vertical velocity respectively.

For the multigrid method a coarsest grid with 80 CVs is used (in Figure 8 the third level with 1280 CVs is shown) and V-cycles with three pre-smoothing, three post-smoothing and three coarse grid iterations are employed. The underrelaxation factors are 0·7 for velocities, 0·3 for pressure and 0·9 for temperature. In Table II the numbers of fine grid iterations are indicated for computing the flow from the initial state to $t = 200$ s for various grid sizes, time step sizes and processor numbers. The corresponding computing times are given in Table III. As in the steady case, the iteration numbers increase only slightly with the processor number. The fine grid iteration numbers decrease when the grid is refined, which may be due to the fact that more coarser grid levels are involved in the computation and less work has to be done on the fine grid. The smaller the time step size, the smaller is the iteration number per time step, but the total number of iterations, owing to the larger number of time steps, increases and therefore the total computing time is also larger. Of coarse, with a smaller time step a higher accuracy is also obtained. Comparing for instance the computing times for 5120 CVs and $P = 6$ with those for 20,480 CVs and $P = 24$, one can further see that a four times larger problem can be solved in approximately the same time with four times more processors.

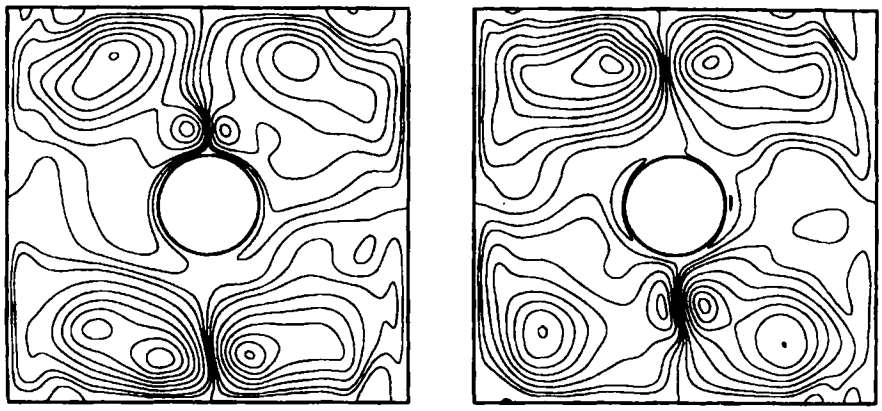

Figure 10. Predicted streamlines for two points of time corresponding to the temporal maximum and minimum values of the maximum vertical velocity for natural convection flow in a square cavity with a circular obstacle

Table II. Numbers of fine grid iterations for different time step sizes, numbers of CVs and processor numbers for natural convection flow in a square cavity with a circular obstacle

	$\Delta t = 4{\cdot}0$			$\Delta t = 2{\cdot}0$			$\Delta t = 1{\cdot}0$		
	$P=6$	$P=12$	$P=24$	$P=6$	$P=12$	$P=24$	$P=6$	$P=12$	$P=24$
1280 CVs	1523	1523	1528	2656	2661	2674	4753	4754	4755
5120 CVs	1410	1411	1423	2529	2529	2539	4617	4632	4628
20480 CVs	1200	1204	1212	2001	2018	2034	3717	3721	3733
81920 CVs	—	965	986	—	1801	1811	—	2847	2895

Table III. Computing times in hours for different time step sizes, numbers of CVs and processor numbers for natural convection flow in a square cavity with a circular obstacle

	$\Delta t = 4{\cdot}0$			$\Delta t = 2{\cdot}0$			$\Delta t = 1{\cdot}0$		
	$P=6$	$P=12$	$P=24$	$P=6$	$P=12$	$P=24$	$P=6$	$P=12$	$P=24$
1280 CVs	0·6	0·4	0·4	1·0	0·7	0·7	1·9	1·3	1·2
5120 CVs	2·0	1·2	0·8	3·6	2·1	1·5	6·5	3·9	2·8
20480 CVs	6·4	3·5	2·1	10·5	5·7	3·4	19·8	10·8	6·4
81920 CVs	—	10·4	5·7	—	19·4	10·5	—	29·9	16·3

For the above test problem we have also studied the performance of the parallel multigrid method in comparison with the corresponding parallel single-grid method. The flow was computed with both methods for three different processor numbers with a fixed ratio of CVs per processor. The computing times, the numbers of fine grid iterations and the acceleration factors obtained with the multigrid method are given in Table IV for different time step sizes. One can see the increase in the acceleration factor with the grid size. Again with the multigrid method one can solve a larger problem in nearly the same amount of computing time when the processor number is increased by the same factor as the number of CVs. For larger time steps the multigrid acceleration factors does not depend significantly on the time step size. If the step size becomes smaller than a certain limit, i.e. if the asymptotic range of convergence with respect to time is reached, the acceleration factor decreases with the time step size. This limit value decreases with increasing grid size.

Table IV. Computing times in hours and numbers of fine grid iterations (in parentheses) for multigrid and single-grid methods for different time step sizes, numbers of CVs and processor numbers (number of CVs per processor is fixed) and corresponding acceleration factors (with respect to computing time) for natural convection flow in a square cavity with a circular obstacle

Grid	Δt	SG	MG	SG/MG
20480 CVs	4·0	25·5 (3887)	11·2 (1200)	2·3
($P=3$)	2·0	43·1 (6558)	18·6 (2001)	2·3
	1·0	55·3 (8417)	34·6 (317)	1·6
81920 CVs	4·0	44·5 (5937)	10·4 (965)	4·3
($P=12$)	2·0	87·1 (11614)	19·4 (1801)	4·5
	1·0	130·5 (17396)	29·9 (2847)	4·4
327680 CVs	4·0	74·9 (9727)	10·9 (953)	6·9
($P=48$)	2·0	—	19·6 (1721)	—
	1·0	—	30·2 (2658)	—

4.3. Parallel computer aspects

Finally, the influence of the arithmetic and communication performance of the parallel computer used for the computation on the efficiency of our method is studied. For this a flow around a circular cylinder in a channel is considered, which is computed on various parallel machines with different numbers of processors for various grid sizes. In Figure 11 the geometry is shown together with the parallel block structures used for the computations with $P = 4$, 16 and 64 processors. The coarsest grid has 256 CVs in each case and up to seven grid levels (corresponding to 1,048,567 CVs) are considered. In Figure 11 the third grid level with 4096 CVs is shown. Concerning the parallel computers, several Parsytec machines (GC/PowerPlus, PowerXPlorer, MultiCluster-3) under the operating system Parix as well as a Kendall Square Research KSR-1 with TCGMSG communication library are included in the comparison.

In Table V the computing times for one V-cycle, which can be considered for both steady and unsteady problems as the basic unit of our multigrid algorithm, are given for the different cases. For all grid levels the V-cycle consists of five pre- and post-smoothing iterations and 10 coarse grid iterations. The relative processor speeds for the machines can be seen by comparing the corresponding one-processor results, while the relative communication performance can be estimated by a comparison of the ratios of the one- and four-processor results for the different machines. The values obtained by proceeding in this way, taking the values for the GC/PowerPlus as a reference (i.e. processor speed $S = 100$, communication speed $C = 100$), are also indicated in Table V. Of course, this gives only a very rough estimate of the performance of the machine, to which a variety of other aspects are contributing, but for our purpose these values are entirely sufficient.

In terms of parallel efficiency one can see an increase with decreasing processor speed S and with increasing communication speed C, because the proportion of communication relative to arithmetic work becomes smaller. Of course, the computing time, which is the most crucial measure in practice for any algorithm on any parallel machine, reduces with increasing processor speed and increasing communication speed.

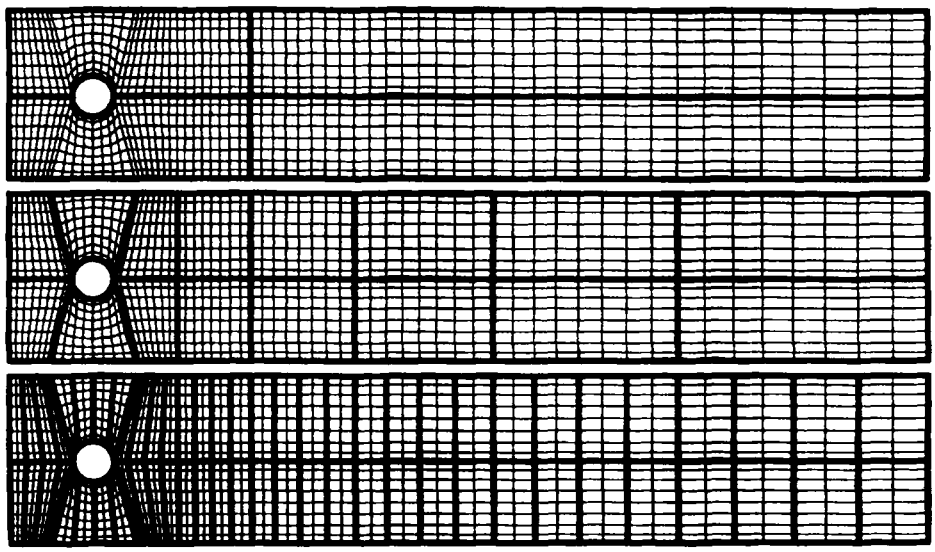

Figure 11. Geometry and parallel block structures for 4, 16 and 64 processors for flow around circular cylinder

Table V. CPU times for one V-cycle for different grid sizes and processor numbers on various parallel computers for flow around a circular cylinder

Computer	P	Number of CVs						
		256	1024	4096	16384	65536	262144	1048567
GC/PowerPlus	1	0·87	17·0	60·7	223·5	—	—	—
($S=1200$, $C=100$)	4	1·58	17·0	34·1	81·6	246·2	—	—
	16	2·90	26·6	43·1	69·3	128·6	322·0	—
	64	10·43	94·3	142·2	197·6	265·1	382·2	668
PowerXPlorer	1	1·02	18·8	66·9	247·4	—	—	—
($S=91$, $C=152$)	4	1·22	14·0	32·2	86·3	274·6	—	—
MC-3	1	12·95	255·4	953·2	—	—	—	—
($S=6$, $C=471$)	4	5·00	77·9	263·6	947·4	—	—	—
	16	3·71	38·0	97·1	284·8	969·4	—	—
KSR-1	1	4·62	76·0	278·1	1082·7	—	—	—
($S=22$, $C=287$)	4	2·92	23·7	67·8	247·4	972·5	—	—

5. CONCLUSIONS

We have presented a parallel implicit finite volume multigrid algorithm for the numerical prediction of flows in complex geometries. The results in the previous sections have shown that the applied block-structuring technique is suitable for handling both complex geometries and parallel processing. It has turned out that the employed parallelization approach based on block-structured grid partitioning results in an efficient parallel implementation mostly retaining the high numerical efficiency of the powerful sequential multigrid solution procedure.

The efficiency of the parallel implementation for a specific problem size depends strongly on the parallel hardware and system software, especially on the ratio of communication to calculation performance. If this ratio is balanced, high efficiencies are achieved. The multigrid method, in spite of its slightly lower efficiency in terms of parallel computing, for both steady and unsteady flows is clearly superior to the corresponding single-grid method. It is also more robust with respect to the decoupling of the subdomains due to the grid partitioning. The problem of determining an optimal partitioning for general geometries is a very complicated task, but for a wide range of applications relatively simple strategies such as considered in this work give satisfactory results. The numerical examples have shown that adequate efficiencies on actual parallel computers and high acceleration compared with serial flow computations are achieved.

In general the results indicate that the increase in computer power due to MIMD parallel computers, combined with the acceleration of the multigrid technique, yields an improved computational performance, enlarging significantly the possibilities for reliable simulation of complex practical flow problems in engineering and science.

ACKNOWLEDGEMENTS

The work was financed by the Bayerische Forschungsstiftung in the Bavarian Consortium of High-Performance Scientific Computing (FORTWIHR) and the Deutsche Forschungsgemeinschaft in the special programme Flow Simulation with High-Performance Computers. This support is gratefully acknowledged.

REFERENCES

1. L. Bai, K. Mitra, M. Fiebig and A. Kost, 'A multigrid method for predicting periodically fully developed flow', *Int. j. numer methods fluids*, **18**, 843–852 (1994).
2. M. Hortmann, M. Perić and G. Scheuerer, 'Finite volume multigrid prediction of laminar natural convection: benchmark solutions,' *Int. j. numer. methods fluids*, **11**, 189–207 (1990).
3. F. S. Lien and M. A. Leschziner, 'Multigrid acceleration for recirculation laminar and turbulent flow computed with a non-orthogonal, collocated finite-volume scheme', *Comput. Methods Appl. Mech. Eng.*, **118**, 351–371 (1994).
4. R. B. Pelz, A. Ecer and J. Häuser (eds), *Parallel Computational Fluid Dynamics '92*, North-Holland, Amsterdam, 1993.
5. K. G. Reinsch, W. Schmidt, A. Ecer, J. Häuser and J. Periaux (eds), *Parallel Computational Fluid Dynamics '91*, North-Holland, Amsterdam, 1992.
6. M. Perić and E. Schreck, 'Computation of fluid flow with a parallel multigrid solver,' *Int. j. numer. methods fluids*,**16**, 303–327 (1993).
7. M. Perić, 'A finite volume method for the prediction of three-dimensional fluid flow in complex ducts,' *Ph.D. Thesis*, University of London, 1985.
8. U. Bückle, F. Durst, B. Howe and A. Melling, 'Investigation of a floating element flowmeter', *Flow Meas. Instrum.*, **4**, 215–225 (1992).
9. F. Durst, L. Kadinskii, M. Perić and M. Schäfer, 'Numerical study of transport phenomena in MOCVD reactors using a finite volume solver', *J. Cryst. Growth*, 612–626 (1993).
10. M. Hortmann, I. Janzig, L. Kadinski, M. Schäfer and H. Scheidat, 'A parallel multigrid algorithm for flow computations with thermal solid/fluid interactions,' in C. Taylor (ed.), *Numerical Methods in Laminar and Turbulent Flow VIII*, Pineridge, Swansea, 1993, pp. 1459–1470.
11. M. Hortmann and M. Schäfer, 'Numerical prediction of laminar flow in plane, bifurcating channels', *Comput. Fluid Mech.*, **2**, 65–82 (1994).
12. I. Demirdžić, Ž. Lilek and M. Pireć, 'A colocated finite volume method for predicting flows at all speeds', *Int. j. numer. methods fluids*, **15**, (1991).
13. E. A. Spiegel and G. Veronis, 'On the Boussinesq approximation for a compressible fluid', *Astrophys. J.*, **131**, 442–447 (1960).
14. I. Demirdžić, and M. Pireć, 'Finite volume method for prediction of fluid flow in arbitrary shaped domains with moving boundaries', *Int. j. numer. methods fluids*, **10**, 771–790 (1990).
15. P. K. Khosla and S. G. Rubin, 'A diagonally dominant second-order accurate implicit scheme', *Comput. Fluids*, **2**, 207–209 (1974).
16. C. Cuvelier, A. Segal and A. A. van Steenhoven, *Finite Element Methods and Navier–Stokes Equations*, Reidel, Dordrecht, 1986.
17. S. V. Patanakar and D. B. Spalding, 'A calculation procedure for heat, mass and momentum transfer in three dimensional parabolic flows,' *Int. J. Heat Mass Transfer*, **15**, 1787–1806 (1972).
18. M. Perić, R. Kessler and G. Scheuerer, 'Comparison of finite-volume numerical methods with staggered and colocated grids,' *Comput. Fluids*, **16**, 389–403 (1988).
19. C. M. Rhie and W. L. Chow, 'Numerical study of the turbulent flow past an airfoil with trailing edge separation', *AIAA J.*, **21**, 1525–1532 (1983).
20. M. Perić, 'Analysis of pressure–velocity coupling on nonorthogonal grids', *Numer. Heat Transfer B*, **17**, 63–82 (1990).
21. R. I. Issa, 'Solution of the implicity discretized fluid flow equations by operator-splitting,' *J. Comput. Phys.*, **62**, 40–65 (1986).
22. G. Wittum, 'On the convergence of multi-grid methods with transforming smoothers', *Numer. Math.*, **57**, 15–38 (1990).
23. W. Hackbusch. *Multi-Grid Methods and Applications*, Springer, Berlin, 1985.
24. M. Schäfer, E. Schreck and K. Wechsler, 'An efficient parallel solution technique for the incompressible Navier–Stokes equations,' in F.-K. Hebeker, R. Rannacher and G. Wittum (eds), *Numerical Methods for the Navier–Stokes Equations*, NNFM Vol. 47, Vieweg, Braunschweig, 1994, pp. 228–238.
25. H. Stone, 'Iterative solution of implicit approximations of multi-dimensional partial differential equations,' *SIAM J. Numer. Anal.*, **5**, 530–558 (1968).
26. M. Schäfer and E. Schreck, 'ILU as a solver in a parallel multi-grid flow prediction code', in *Incomplete Decompositions (ILU)—Algorithms, Theory, and Applications*, NNFM Vol. 41, Vieweg, Braunschweig, 1993, pp. 149–158.
27. M. Schäfer, 'Numerical solution of the time-dependent axisymmetric Boussinesq equations on processor arrays', *SIAM J. Sci. Stat. Comput.*, **13**, 1377–1393 (1992).