INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2000; **32**: 141–173

Schwarz domain decomposition for the incompressible Navier–Stokes equations in general co-ordinates

E. Brakkee[a,*], P. Wesseling[b,1] and C. G. M. Kassels[b,2]

[a] *GMD/SCAI Forschungszentrum Informationstechnik, Schloß Birlinghoven, D-53754 Sankt Augustin, Germany*
[b] *Applied Analysis Group, Delft University of Technology, Mekelweg 4, 2628 CD Delft, Netherlands*

SUMMARY

This paper describes a domain decomposition method for the incompressible Navier–Stokes equations in general co-ordinates. Domain decomposition techniques are needed for solving flow problems in complicated geometries while retaining structured grids on each of the subdomains. This is the so-called block-structured approach. It enables the use of fast vectorized iterative methods on the subdomains. The Navier–Stokes equations are discretized on a staggered grid using finite volumes. The pressure-correction technique is used to solve the momentum equations together with incompressibility conditions. Schwarz domain decomposition is used to solve the momentum and pressure equations on the composite domain. Convergence of domain decomposition is accelerated by a GMRES Krylov subspace method. Computations are presented for a variety of flows. Copyright © 2000 John Wiley & Sons, Ltd.

KEY WORDS: domain decomposition; GMRES; incompressible Navier–Stokes; boundary-fitted co-ordinates; incompressible Navier–Stokes; staggered grid; pressure-correction

1. INTRODUCTION

In recent years, much progress has been made in the application of computational fluid dynamics to engineering problems. Several approaches have been proposed for the discretization of the incompressible Navier–Stokes equations in arbitrarily shaped domains with a finite difference or finite volume method. We adopt a boundary conforming finite volume method. Methods mainly differ in grid arrangement (staggered or collocated grids), and in the choice of unknowns. Pioneering papers for the collocated grid approach are [1–3]; an interesting paper on the staggered grid approach is [4]. The various possible grid layouts and choice of primary dependent variables are reviewed in [5]. We have chosen for the staggered grid arrangement because spurious pressure oscillations are absent and artificial pressure boundary conditions are not required.

* Correspondence to: Ericsson Telecommunicatie BV/ETM, Ericssonstraat 2, 5121 ML Rijen, Netherlands.
[1] E-mail: p.wesseling@math.tudelft.nl
[2] E-mail: k.kassels@math.tudelft.nl

CCC 0271–2091/2000/020141–33$17.50

Received 20 December 1994
Revised 1 August 1998

Discretization in general co-ordinates on a staggered grid is a far from trivial matter. Several special precautions must be taken to ensure accuracy on non-smooth grids, see e.g. [6]. A central discretization is used for the discretization of the convective terms. References [7–14] describe the discretization in detail and [15] discusses the ability of the method to accurately solve a number of benchmark problems. Prior to the discretization of the incompressible Navier–Stokes equations, the flow domain is mapped onto a rectangular domain, generating boundary conforming co-ordinates, see Figure 1.

The motivation behind the boundary conforming approach is that the resulting computational grid is rectangular and structured, and that accurate discretization of boundary conditions is easy. The rectangular structured computational grid simplifies programming, because, for instance, the structure of matrices that arise in the computation is known beforehand. This property enables us to write efficient iterative solvers for these equations. The latter requires more work for, for instance, finite element discretizations on unstructured meshes.

It can be easily seen that the boundary conforming method cannot handle complex geometries, because a sufficiently smooth mapping may not exist. Therefore, we use the so-called block-structured approach, in which the global flow domain is decomposed into subdomains, each of which can be easily mapped onto a rectangle (see Figure 2). Of course, the subdomain equation systems are not independent of each other and we need to couple the solutions in the subdomains. The resulting method is called a domain decomposition method.

In a previous model study for the advection–diffusion equation [16,17] Chapter 3, we have investigated several coupling conditions and their effect on convergence of the domain decomposition algorithm. We also considered the effect of Krylov subspace acceleration on convergence. It turned out that, compared with Schwarz-type coupling conditions, improvements in convergence with variations of Neumann–Dirichlet- and Robin-type coupling conditions for the unaccelerated algorithm are possible. But these improvements are not significant when Krylov subspace acceleration is applied. Therefore, we shall use the accelerated

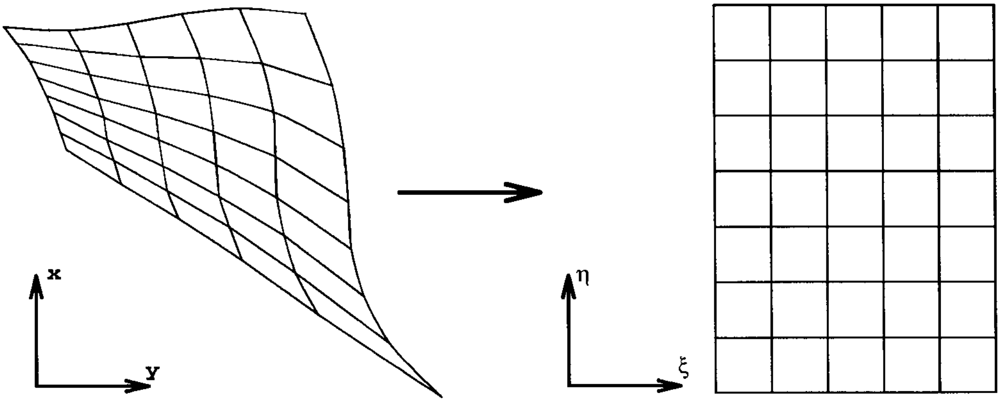


Figure 1. Transformation of the flow region to a rectangle in computational space.

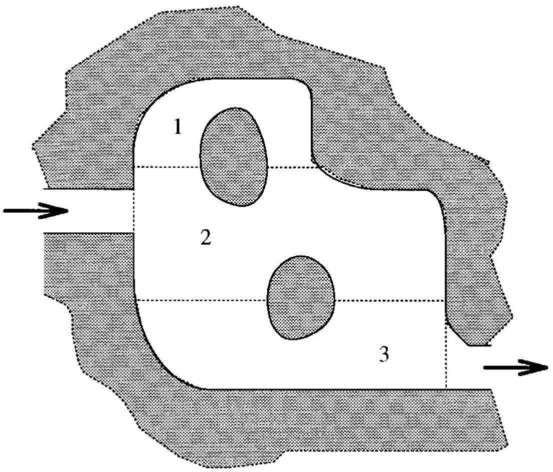


Figure 2. Flow around two islands. It is virtually impossible to map this domain onto a rectangle. Three subdomains are used here.

Schwarz [18,19] domain decomposition algorithm. Related work on the subject of influencing convergence behavior by varying the coupling conditions can be found in [20–22].

There is a vast amount of literature on the subject of domain decomposition for solving partial differential equations. Here we mention only those publications that bear directly on the type of flow computations that we envisage here. In [23], a multi-grid is extended to composite grids with an arbitrary overlap. The pressure-correction method is used for solving the incompressible Navier–Stokes equations on staggered grids. An explicit time integration scheme is used, so that only a domain decomposition problem for the pressure equation remains. In [24], a multi-grid method is used to solve the Poisson equation on arbitrary overlapping composite grids, and good convergence results are obtained. A complete composite multi-grid (CCMG) method is used where information is transferred between subdomains at each grid level. In [25], the procedures described in [23,24] are extended by considering Neumann–Dirichlet coupling conditions instead of the basic Schwarz iteration. In [26,27], the domain decomposition method consists of solving both velocity and pressure on each subdomain simultaneously. This causes a difficulty with pressure levels in the subdomains, which must be matched after the computation since they are only determined up to a constant. Since both the momentum and pressure equations are solved in each domain decomposition iteration, the method may be quite expensive. Both papers employ a constant overlap in physical space, thus ensuring good convergence properties as the mesh size is refined.

A more efficient approach is to solve the momentum and pressure equations separately on the composite domain. This was demonstrated in [28–30]. In [28,29], an explicit time integration was used like in [23], but with an extension to arbitrary overlapping grids, thus releasing the requirement of having coincident grid points at the inter-grid boundaries. In contrast to Reference [24], References [28–30] employ an incomplete composite multi-grid (ICMG) method where information is transferred between subdomains only at the finest grid level. The reason is that ICMG is much easier to implement than CCMG and also appears to be efficient. In their ICMG method, information is transferred after each V-cycle.

The solution method of [28,29] consists of an explicit solution of the momentum equations, followed by an iterative solution of the pressure equations by using a Schwarz-type method. In the latter case, special care must be taken to obtain well-posed subdomain problems. For this purpose, the subdomains are solved with Neumann boundary conditions, in combination with a mass imbalance correction (MIC) scheme, see [31,32]. In [30] the authors use an improved discretization developed in [5] and use a semi-implicit time integration instead of the fully explicit one used in [28,29]. The latter also makes Schwarz iteration for the momentum equations necessary. More details about the methods of [28–30] will be given later when more detailed comparisons with our method are made.

In [33], a non-linear multi-grid scheme is used that employs a pressure-correction method of SIMPLE-type as a smoother on the different levels for convergence acceleration. The θ time integration method is used in combination with a second-order finite volume space discretization using cell-centered collocated block-structured grids. The systems of equations for the velocities $M_1v = b_1$, pressure-correction $M_2\Delta P = b_2$, and temperature $M_3T = b_3$. are solved using a Richardson iteration with a blocked-ILU preconditioner. This works as follows. The partitioning of the global solution domain into blocks implies a partitioning of the matrices M_1, M_2 and M_3 into blocks. The blocked-ILU preconditioner is now a matrix with the same block structure as M_i but with the diagonal blocks replaced by ILU preconditioners of the diagonal blocks of M_i and the off-diagonal blocks set to zero. In other words. it is like a block-Jacobi preconditioned Richardson iteration with inaccurate subdomain solution using a few ILU iterations.

As opposed to [26,27,33], we solve the momentum and pressure equations separately over the composite domain, instead of solving these equations simultaneously in the subdomains. As opposed to [23–25,28–30,33], we do not use a multi-grid method, but solve the subdomains accurately using the GMRES method, see [34,35]. Also, we use a fully implicit time integration that is in contrast to [23–25,28,29], where an explicit time integration was used, and to [30] where a semi-implicit time integration was used.

Another important difference between our method and those of [26–30] is that we use a minimal overlap instead of an arbitrary overlap. With minimal overlap, it is meant here that the overlap area approaches zero as the mesh is refined. The reason for choosing minimal overlap is that it avoids the increasing duplication of work in overlap areas as the mesh is refined. Having minimal overlap also simplifies implementation. Recent experiences [36–38] have shown that although a large overlap generally gives better convergence rates, minimal overlap typically leads to lower computing times, even for large and ill-conditioned problems. Having minimal overlap is also the prevalent approach in engineering codes due to its convenience.

Section 2 presents the Navier–Stokes solution procedure. After that, Sections 3 and 4 describe the domain decomposition algorithm and its Krylov subspace acceleration. Section 5 gives some results for some complex geometries, like the backward-facing step problem, skewed cavity problem, and the problem of flow around a cylinder in a wall-bounded shear flow. Finally, Section 6 presents our conclusions.

2. NAVIER–STOKES SOLUTION PROCEDURE

The flow solver described in [8,12,14,39] is used as a basis for domain decomposition. A staggered grid is used in boundary-fitted co-ordinates. The discretization is co-ordinate invariant. The unknowns are the pressure (in cell centers), and the contravariant mass flux components (in the centers of cell-faces). For details, see the above-mentioned references.

Space discretization in general co-ordinates results in a system of differential algebraic equations that may be summarized as:

$$\begin{cases} R\dot{V}_{\mathrm{M}} = f(V_{\mathrm{M}}, V_{\mathrm{B}}) - G_{\mathrm{M}}p_{\mathrm{M}}, \\ D_{\mathrm{M}}V_{\mathrm{M}} + D_{\mathrm{B}}V_{\mathrm{B}} = 0. \end{cases} \tag{1}$$

The matrix R is a diagonal matrix with values of the density on the diagonal. V_{M} represents all non-prescribed contravariant mass flux components, and V_{B} contains velocity components that are prescribed at the external boundaries:

$$V_{\mathrm{B}} = g(t). \tag{2}$$

The vector p_{M} represents all non-prescribed pressure unknowns, f represents the viscous stress as well as the convective terms and the volume forces. The matrix G_{M} represents the discretization of the gradient operator. The operator $D = (D_{\mathrm{M}}, D_{\mathrm{B}})$ represents the discrete approximation of the divergence operator. Figure 3 presents the various stencils that arise in the space discretization in computational space.

The θ method is used for time discretization of (1), leading to:

$$R\frac{V_{\mathrm{M}}^{n+1} - V_{\mathrm{M}}^{n}}{\Delta t} = \theta f(V_{\mathrm{M}}^{n+1}, V_{\mathrm{B}}^{n+1}) + (1-\theta)f(V_{\mathrm{M}}^{n}, V_{\mathrm{B}}^{n}) - \theta G_{\mathrm{M}}p_{\mathrm{M}}^{n+1} - (1-\theta)G_{\mathrm{M}}p_{\mathrm{M}}^{n}, \tag{3.a}$$

$$V_{\mathrm{B}}^{n+1} = g^{n+1}, \tag{3.b}$$

$$D_{\mathrm{M}}V_{\mathrm{M}}^{n+1} + D_{\mathrm{B}}V_{\mathrm{B}}^{n+1} = 0. \tag{3.c}$$

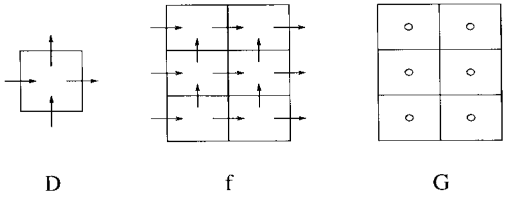


Figure 3. Discretization stencils for the divergence D, V^1 component of stress and convection f, and V^1 component of gradient G operators.

With the pressure-correction method [40–42], (3.a) and (3.b) are replaced by:

$$R\frac{V_{\rm M}^{*}-V_{\rm M}^{n}}{\Delta t}=\theta f(V_{\rm M}^{*},V_{\rm B}^{n+1})+(1-\theta)f(V_{\rm M}^{n},V_{\rm B}^{n})-G_{\rm M}p_{\rm M}^{n}, \tag{4.a}$$

$$V_{\rm B}^{*}=g^{n+1}. \tag{4.b}$$

The non-linear term $f(V_{\rm M}^{*}, V_{\rm B}^{n+1})$ is linearized, leading to:

$$f(V_{\rm M}^{*},V_{\rm B}^{n+1})\cong M(V_{\rm M}^{n})V_{\rm M}^{*}, \tag{5}$$

with M a linear operator. The resulting system for $V_{\rm M}^{*}$ is solved in two steps. Substitution of (5) into (4.a) and the introduction of $V_{\rm M}^{**}=\theta V_{\rm M}^{*}+(1-\theta)V_{\rm M}^{n}$ gives

$$S(V_{\rm M}^{n})V_{\rm M}^{**}=\frac{R}{\theta\Delta t}V_{\rm M}^{n}-G_{\rm M}p_{\rm M}^{n}, \tag{6}$$

with the momentum matrix

$$S(V_{\rm M}^{n})=\frac{R}{\theta\Delta t}-M(V_{\rm M}^{n}). \tag{7}$$

The solution $V_{\rm M}^{*}$ is obtained by

$$V_{\rm M}^{*}=\frac{1}{\theta}(V_{\rm M}^{**}-(1-\theta)V_{\rm M}^{n}). \tag{8}$$

Subtracting (4.a) from (3.a) gives:

$$R\frac{V_{\rm M}^{n+1}-V_{\rm M}^{*}}{\Delta t}=\theta\{f(V_{\rm M}^{n+1},V_{\rm B}^{n+1})-f(V_{\rm M}^{*},V_{\rm B}^{n+1})\}-\theta G_{\rm M}(p_{\rm M}^{n+1}-p_{\rm M}^{n}). \tag{9}$$

In [42] it is shown that the first term on the right-hand-side may be neglected without affecting the order of accuracy. Hence, (9) is replaced by

$$\frac{V_{\rm M}^{n+1}-V_{\rm M}^{*}}{\Delta t}=-\theta R^{-1}G_{\rm M}(p_{\rm M}^{N+1}-p_{\rm M}^{n}). \tag{10}$$

Applying the discrete divergence operator $D_{\rm M}$ to (10) gives the pressure equations

$$-\theta D_{\rm M}R^{-1}G_{\rm M}(p_{\rm M}^{n+1}-p_{\rm M}^{n})=\frac{D_{\rm M}V_{\rm M}^{n+1}-D_{\rm M}V_{\rm M}^{*}}{\Delta t}. \tag{11}$$

Because of the divergence freedom (3.c) we may also write this as

$$-\theta D_{\mathrm{M}} R^{-1} G_{\mathrm{M}}(p_{\mathrm{M}}^{n+1} - p_{\mathrm{M}}^{n}) = \frac{D_{\mathrm{M}} V_{\mathrm{M}}^{n+1} - D_{\mathrm{M}} V_{\mathrm{M}}^{*}}{\Delta t}. \tag{12}$$

The boundary conditions for the normal velocity components are already substituted in V^* so that $V_{\mathrm{B}}^{*}/V_{\mathrm{B}}^{n+1}$ and (12) simplifies to

$$D_{\mathrm{M}} R^{-1} G_{\mathrm{M}} \Delta p = \frac{DV^*}{\theta \Delta t}, \tag{13}$$

with $\Delta p = p_{\mathrm{M}}^{n+1} - p_{\mathrm{M}}^{n}$. After the solution of (13), we set $p_{\mathrm{M}}^{n+1} = p_{\mathrm{M}}^{n} + \Delta p$, and V_{M}^{n+1} is obtained from (10).

In summary, the pressure-correction method consists of three consecutive steps:

- Step 1: Solution of V_{M}^{*} from (6) and (8).
- Step 2: Solution of Δp from (13). Equation (13) is a linear 'Poisson-like' system of equations.
- Step 3: Computation of $p_{\mathrm{M}}^{n+1} = p_{\mathrm{M}}^{n} + \Delta p$ and computation of V_{M}^{n+1} from (10). This is an explicit matrix–vector operation, and guarantees divergence freedom of the velocity field.

3. DOMAIN DECOMPOSITION

The aim of the domain decomposition algorithm is to enforce the local conservation laws (mass conservation, momentum continuity) on the global domain, by applying the discretized equations across subdomain boundaries. For this reason, the pressure-correction algorithm of the previous section is applied to the global domain. Because of this, the conservation laws automatically hold across block interfaces after convergence. This is in contrast to [26,27], where specific measures had to be taken to ensure conservation and to obtain a continuous pressure field across block interfaces. Domain decomposition is used inside the pressure-correction method instead of *vice versa* as in [26,27]. At present, the grids must match at the subdomain boundaries. In the future, this restriction may have to be lifted to increase the flexibility in grid generation. Note that the domain decomposition itself is allowed to be unstructured, so that the number of subdomains that share a corner point is not restricted (see Figure 4).

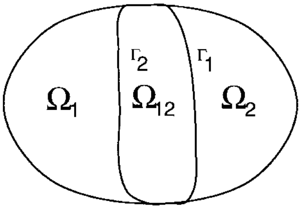


Figure 4. An overlapping decomposition of the domain.

3.1. Algebraic Schwarz iteration for the discretized Navier–Stokes equations

For the description of domain decomposition we start from the discretized equations on the global domain. Equation (6) represents the linearized momentum equations and Equation (13) represents the pressure equations on the global domain. It is not necessary to form these global discretization matrices in an implementation. It is sufficient to discretize the equations per subdomain. Equations (6) and (13) are solved using domain decomposition. The last step of the pressure-correction algorithm (10) is carried out in all subdomains independently. For the sake of argument, consider a situation of two blocks, as in Figure 5.

Both the pressure equations (13) and the momentum equations (6) are of the form

$$Au = f, \tag{14}$$

with either $A = S$ from (7) and $u = V^{**}$ for the momentum equations or $A = D_{\mathrm{M}}R^{-1}G_{\mathrm{M}}$ from (13) and $u = \Delta p$ for the pressure equations. If we decompose A into blocks such that each block corresponds to all unknowns in a single subdomain, with a small modification for the momentum equations (see further on), then

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \tag{15}$$

where A_{11} and A_{22} represent the subdomain discretization matrices and A_{12} and A_{21} represent the coupling between subdomains. The unaccelerated domain decomposition iteration for (14) is of the following form

$$u^{m+1} = (I - N^{-1}A)u^m + N^{-1}f, \tag{16}$$

where N denotes a block Gauss–Seidel or block Jacobi preconditioner, so that

$$N = N_{\mathrm{gs}} = \begin{bmatrix} A_{11} & \varnothing \\ A_{21} & A_{22} \end{bmatrix} \quad \text{or} \quad N = N_{\mathrm{jac}} = \begin{bmatrix} A_{11} & \varnothing \\ \varnothing & A_{22} \end{bmatrix}. \tag{17}$$

Since the normal mass flux components at the block interfaces belong to two subdomains, the momentum matrix S (7) is augmented before application of (16). Consider the decomposition into two blocks as in Figure 5. Suppose that the mass fluxes unknowns are divided into three sets as in Figure 5.

- $\tilde{I}_1 = I_{\mathrm{nr}} \cup I_{\mathrm{r}}$ consists of mass fluxes belonging to block 1 excluding the normal mass fluxes at the block interfaces.
- $\tilde{I}_2 = II$ consists of the normal mass fluxes at the interface.
- $\tilde{I}_3 = III_{\mathrm{r}} \cup III_{\mathrm{nr}}$ consists of the mass fluxes belonging to block 2 excluding the normal mass fluxes at the block interfaces.

With respect to these three sets of unknowns, the matrix S (7) has the block form:

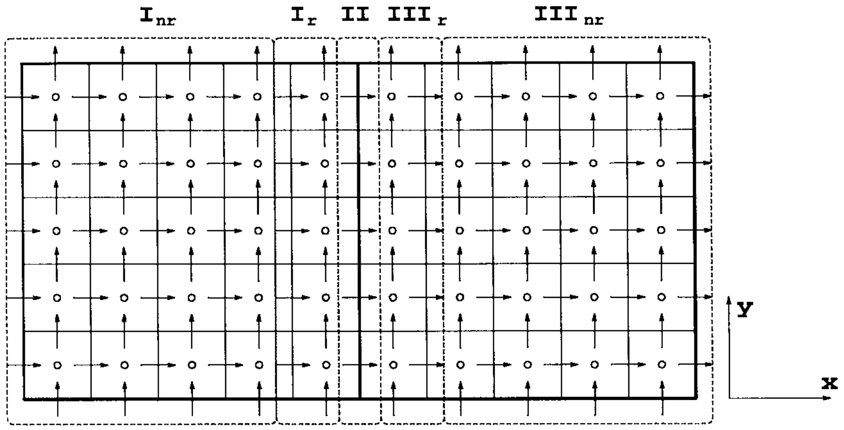


Figure 5. Different sets of unknowns. I_{nr} and I_{r} constitute set I and III_{nr} and III_{r} constitutes set III. The index set $I_1 = I_{\mathrm{nr}} \cup I_{\mathrm{r}} \cup II$ constitutes domain Ω_1 and $I_2 = II \cup III_{\mathrm{r}} \cup III_{\mathrm{nr}}$ constitutes Ω_2.

$$S = \begin{bmatrix} S_{11} & S_{12} & S_{13} \\ S_{21} & S_{22} & S_{23} \\ S_{31} & S_{32} & S_{33} \end{bmatrix}, \quad \text{with } V^{**} = \begin{bmatrix} V_1^{**} \\ V_2^{**} \\ V_3^{**} \end{bmatrix}. \tag{18}$$

The system of equations $SV^{**} = f$ can be transformed to the equivalent system

$$\bar{S}\bar{V}^{**} = \begin{bmatrix} S_{11} & S_{12} & 0 & S_{13} \\ S_{21} & S_{22} & 0 & S_{23} \\ S_{21} & 0 & S_{22} & S_{23} \\ S_{31} & 0 & S_{32} & S_{33} \end{bmatrix} \cdot \begin{bmatrix} \bar{V}_1^{**} \\ \bar{V}_2^{**} \\ \bar{V}_2'^{**} \\ \bar{V}_3^{**} \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_2 \\ f_3 \end{bmatrix}. \tag{19}$$

The solution of (19) always satisfies $\bar{V}_2^{**} = \bar{V}_2'^{**}$ if S_{22} is invertible (see [21]) and therefore, the system (19) is equivalent to the original system of equations $SV^{**} = f$. With

$$A_{11} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix}, \qquad A_{22} = \begin{bmatrix} S_{22} & S_{23} \\ S_{32} & S_{33} \end{bmatrix}, \tag{20}$$

we obtain a system of the form (15).

The algebraic Schwarz domain decomposition algorithm [18,19] is a generalization of block Gauss–Seidel and Jacobi iterations for overlapping blocks of the matrix. Block Gauss–Seidel and Jacobi iterations are similar to Schwarz domain decomposition, in the sense that in each iteration, subdomain problems are solved using values of u from neighboring blocks. For instance, formula (16) interpreted for domain 1 becomes

$$u_1^{m+1} = A_{11}^{-1}(f - A_{12}u_2^m), \tag{21}$$

where A_{11}^{-1} represents subdomain solution and u_2^m are the values from the neighboring block. The subdomain problems, $A_{11}u_1 = \dots$ and $A_{22}u_2 = \dots$ are solved accurately using an iterative method. We use GMRES [34] with appropriate preconditioners [35,43] to solve the subdomain problems—the so-called inner iteration (16). A GMRES method is also used later on in Section 4 to accelerate domain decomposition—the so-called outer iteration. In effect, the iterative methods used for the inner and outer iterations can be replaced by other methods independent of each other.

Note that the normal mass fluxes at the block interface in region II are a part of the inner regions of the subdomains, and are solved for in each iteration.

This concludes the description of the solution of (6) and (13) using domain decomposition. Prior to the solution of (13), some precautions are necessary. A well-known problem with the pressure-correction method is that in the case of given normal inflow velocity components at all boundaries, the pressure matrix $A = D_M R^{-1} G_M$ from (13) is singular. For this reason, the right-hand-side of the pressure equation must satisfy a compatibility constraint. In such a case, we have[3]:

$$[1 \quad 1 \quad \cdots \quad 1]A = 0, \tag{22}$$

which leads to the following compatibility constraint for the right-hand-side:

$$[1 \quad 1 \quad \cdots \quad 1]\frac{1}{\theta \Delta t} D V^{n+1} = [1 \quad 1 \quad \cdots \quad 1]A\,\Delta p = 0. \tag{23}$$

This compatibility constraint is easily rewritten as:

$$\sum_{\text{all grid cells } (i,j)} D_{i,j} V^{n+1} = 0, \tag{24}$$

with $D_{i,j}$ the divergence operator in grid cell (i, j), see Figure 3. Requirement (24) is automatically satisfied in a single domain case if the boundary conditions are compatible with the incompressibility condition. However, in a general multi-domain case, the normal mass fluxes at the block interfaces belong to two different blocks in our augmented system, and are still slightly different after convergence of the domain decomposition algorithm for the mass fluxes V^{**}. It can be seen that (24) requires that

$$\sum_{\substack{\text{all normal mass fluxes}\\ \text{on the interface}}} V_{N,\text{left}}^{n+1} = \sum_{\substack{\text{all normal mass fluxes}\\ \text{on the interface}}} V_{N,\text{right}}^{n+1}, \tag{25}$$

[3] Only with special scaling of the equations are the elements of the row vector all unity.

which is not satisfied accurately. It has been observed in practice that even a slight difference in the values of the normal mass fluxes can lead to serious problems when solving the pressure equation. Therefore, we have introduced an intermediate step in the algorithm that ensures equal normal mass fluxes simply by copying their values of one block to the other.

Application of (25) does not affect the solution accuracy since the magnitude of the correction is smaller than the solution accuracy with which the domain decomposition problem for the momentum equations was solved.

The above discussion can be extended to the general multi-domain case, see also Section 3.2. Also, extensions to irregular intersections are possible, see, for example, [24,27,44].

The correction prescribed by formula (25) has a similar function as that of the mass imbalance correction (MIC) [31,32] scheme as used in [28–30], namely to obtain well-posed problems for the pressure equations. The function of our correction scheme (25) is to ensure that no mass is lost at the interfaces. The mass is not conserved on the subdomains by (25). Note that this is also not required, since the solution V^* obtained from (8) and (6) also does not need to satisfy divergence freedom. Application of (25) ensures that the global pressure equations (i.e. on the composite domain) are well-posed.

The global pressure equations are solved using Schwarz domain decomposition iteration. In each Schwarz iteration for the pressure, the subdomain problems are solved using values of Δp from the neighboring subdomain; in other words, using Dirichlet-type boundary conditions. This means that the subdomain pressure equations are automatically well-posed, and no compatibility conditions for the subdomain problems apply.

In [33], no correction for mass imbalance is needed because the unknowns velocity, pressure and temperature reside at the cell-centers, so that different blocks do not have unknowns at the block interfaces in common.

The method of [28–30] is somewhat different. In these, the authors solve subdomain problems that resemble well-posed Navier–Stokes problems where the normal velocity is prescribed everywhere on the interface and external boundary. The most natural choice is then to use Neumann boundary conditions (see [28]). This, as explained above, requires that the compatibility condition (24) is satisfied for the subdomains. To satisfy (24), the authors correct the boundary mass fluxes using the MIC scheme. As they observe, the applied corrections are relatively small (less than 1%, see [29]). Because their method focuses on subdomain mass conservation, it will not preserve mass at the block interfaces up to machine precision.

The boundary mass fluxes needed to solve a pressure subdomain problem in [28–30] are obtained by computing

$$(V_{\mathrm{M}}^{n+1})_{\mathrm{boundary}} = (V_{M}^{*,i} - \theta \Delta t R^{-1} G_{\mathrm{M}}(p_{\mathrm{M}}^{n+1,i} - p_{\mathrm{M}}^{n}))_{\mathrm{interpolated}}, \tag{26}$$

where i denotes the iteration index. In other words, by carrying out the correction (10) of V^* in each pressure Schwarz iteration and interpolating this to the interface. This procedure thus mixes pressure-correction with Schwarz—or Navier–Stokes solution methods with linear algebra—and may therefore be difficult to implement. During iteration of the pressure, the velocity field must be continuously updated.

In our method, momentum and pressure equations are treated separately, with an intermediate correction by (25) between the solution of these two systems. Formula (25) ensures that no mass is lost at the block interfaces and the pressure-correction method itself ensures local mass conservation. The scheme used in [28–30] only ensures mass conservation on the subdomains (not at the block interfaces) and can probably only be applied in those cases where the velocity is prescribed at all external boundaries. It is not clear how their method can be applied to cases where, on some parts of the boundary, the normal velocity component is not imposed. The correction (25), on the other hand, can always be applied.

3.2. Extension to the general multi-domain case

The explanation of the domain decomposition algorithm was restricted to a two-block situation and started from an (imaginary) discretization on the global domain. However, in our implementation we cannot form this imaginary matrix. For instance, a problem occurs when less than four or more than four blocks intersect at the same point, see Figure 6. We wish to allow this to happen.

An accurate discretization in such difficult points would require an immense administration in the program and probably also large rewrites of our code (in finite element fashion). This is not desirable and simple solutions to this problem are wanted.

To avoid this problem, the domain decomposition algorithm communicates with neighboring blocks only through the common subdomain interface and not through the corners. The consequence of this is that we must do something special at corner points. Consider as an example the bottom-right subdomain of Figure 6. The discretization at the top-left corner point extends across the subdomain boundaries, and we see that the stencil requires an unknown in the corner of the domain (marked *), see Figure 7.

In the multi-block configuration of Figure 6, these unknowns do not correspond to an actual unknown from a neighboring block. This problem is solved by eliminating the

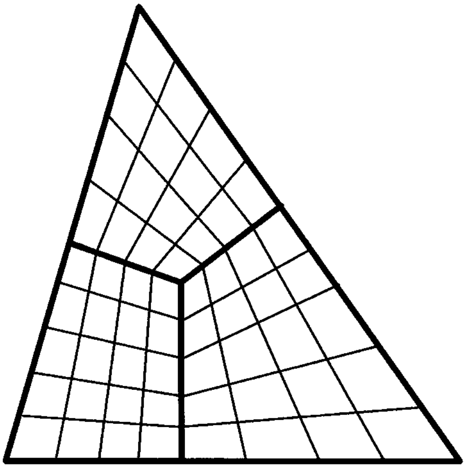


Figure 6. Three blocks intersecting in the same point.

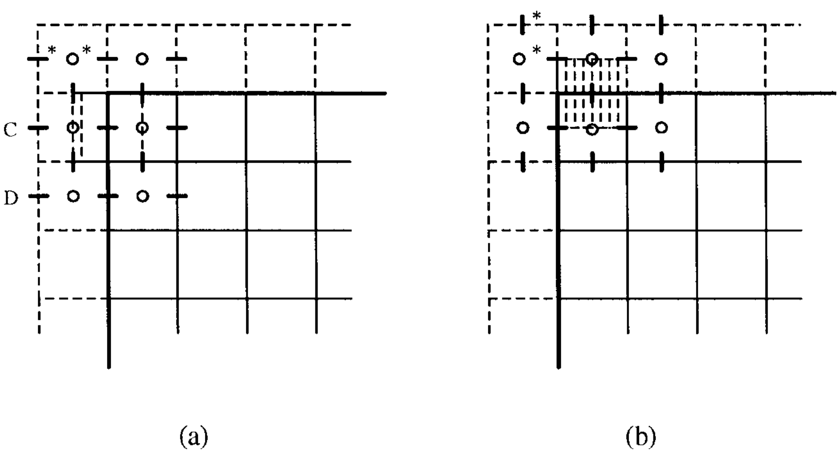


Figure 7. Discretization at a corner point. (a) Momentum and pressure-1 component, (b) momentum and pressure-2 component. The asterisks indicate unknowns to be eliminated.

corresponding unknowns in blocks with which we want to avoid communication by bilinear extrapolation from the four mass fluxes within the domain under consideration. Pressures are eliminated by linear extrapolation from the other two pressures along the domain boundary with which we do allow communication.

The reason that this extrapolation procedure works is that the discretization stencils at only a few points are disturbed. These local disturbances have a global effect due to the viscous terms, but the resulting disturbance of the global solution is a rapidly decaying (Green's) function of the distance to these points.

A detailed analysis of the local discretization error when using this extrapolation scheme shows that the resulting discretization is not consistent anymore. Therefore, it needs to be checked whether or not this approach lowers the order of accuracy of the resulting solution. We investigate this by numerical experiments on a representative model problem. We solve a Poisson equation perturbed with a mixed derivative term:

$$\begin{cases} -\Delta\phi + \gamma \dfrac{\partial^2 \phi}{\partial x_1\, \partial x_2} = (-2+\gamma)\mathrm{e}^{x_1+x_2}, \\ \phi|_{\partial\Omega} = \mathrm{e}^{x_1+x_2}, \end{cases} \quad \text{on } \Omega = (0, 1) \times (0, 1). \tag{27}$$

The right-hand-side and boundary conditions were chosen such that the solution $\phi(x_1, x_2) = \mathrm{e}^{x_1+x_2}$ contains all powers of x_1 and x_2 in a Taylor series expansion. A second-order cell-centered discretization is used on a uniform Cartesian mesh. The domain is decomposed into a rectangular array of 2 × 2 blocks. We compare the solution ϕ_1^h, which is obtained by single-block solution (without extrapolation at internal corners), and the solution ϕ_2^h, which is obtained by the multi-block solution with the extrapolation procedure at the internal corners described above. The claim that we want to verify is that ϕ_2^h is just as accurate as ϕ_1^h or that $\delta^h = \phi_2^h - \phi_1^h = O(h^2)$. For this purpose we use the Euclidian norm $\| * \|_{\mathrm{E}}$, which describes the average perturbation, defined as

$$\|y\|_{\mathrm{E}} = \sqrt{\frac{1}{n} \sum_{i=1,\ldots,n} y_i^2} \tag{28}$$

and the maximum norm $\| * \|_\infty$,

$$\|y\|_\infty = \max_{i=1,\ldots,n} |y_i| \tag{29}$$

for $y \in \mathbb{R}^n$.

Table I shows some results for $\|\delta^h\|$ for different values of the parameter γ. It is clear from this table that the ratio $\|\delta^{2h}\|/\|\delta^h\| \approx 4$ for both norms. Therefore, the order of the perturbation δ^h caused by extrapolation at internal corners is the same as the order of accuracy of the numerical solution. Hence, the extrapolation procedure at the internal corners is allowed.

4. KRYLOV SUBSPACE ACCELERATION

The domain decomposition procedure as described in Section 3 converges slowly and is particularly sensitive to variations in the number of blocks, subdomain sizes, and types of external boundary conditions. Nevertheless, as we will show, it is a good preconditioner and

Table I. Perturbation of the solution caused by extrapolation at internal corners

h	$\gamma = 0.25$		$\gamma = 0.50$	
	$\|\delta^h\|_{\mathrm{E}}$	$\|\delta^{2h}\|_{\mathrm{E}}/\|\delta^h\|_{\mathrm{E}}$	$\|\delta^h\|_{\mathrm{E}}$	$\|\delta^{2h}\|_{\mathrm{E}}/\|\delta^h\|_{\mathrm{E}}$
0.1	$1.42 \cdot 10^{-4}$		$3.25 \cdot 10^{-4}$	
0.05	$1.96 \cdot 10^{-5}$	7.23	$5.25 \cdot 10^{-5}$	6.18
0.025	$3.19 \cdot 10^{-6}$	6.16	$1.05 \cdot 10^{-5}$	5.01
0.0125	$7.36 \cdot 10^{-7}$	4.25	$2.45 \cdot 10^{-6}$	4.27
	$\|\delta^h\|_\infty$	$\|\delta^{2h}\|_\infty/\|\delta^h\|_\infty$	$\|\delta^h\|_\infty$	$\|\delta^{2h}\|_\infty/\|\delta^h\|_\infty$
0.1	$7.67 \cdot 10^{-3}$		$1.85 \cdot 10^{-3}$	
0.05	$2.00 \cdot 10^{-4}$	3.83	$4.98 \cdot 10^{-4}$	3.71
0.025	$5.20 \cdot 10^{-5}$	3.85	$1.34 \cdot 10^{-4}$	3.72
0.0125	$1.40 \cdot 10^{-5}$	3.71	$3.60 \cdot 10^{-5}$	3.72

$\delta^h = \phi_2^h - \phi_1^h$ measures the effect of extrapolation on the numerical solution.

we will use it as such in combination with a GMRES [34] Krylov subspace method. Such a GMRES acceleration procedure gives much more robustness with respect to these parameters, see [16,17].

Krylov subspace methods are frequently used to accelerate domain decomposition methods but mostly in conjunction with an approximate solution of the subdomains, see, for example, [45] and many of the papers on iterative substructuring methods in [46–50]. We use an accurate solution of the subdomains, which means that the Krylov subspace acceleration uses in effect a block Gauss–Seidel/Jacobi preconditioning of the global system of equations.[4] The block Gauss–Seidel/Jacobi preconditioned global system of equations has the following block form:

$$Bv = \begin{bmatrix} I & R \\ 0 & D \end{bmatrix} \begin{bmatrix} v_{\mathrm{nr}} \\ v_{\mathrm{r}} \end{bmatrix} = \begin{bmatrix} g_{\mathrm{nr}} \\ g_{\mathrm{r}} \end{bmatrix}, \tag{30}$$

where the components of v_{r} correspond to unknowns in the regions I_{r} and III_{r} in Figure 5, and those of v_{nr} correspond to the remaining unknowns. This block form can be derived as follows. The block Jacobi or Gauss–Seidel iteration is written as in (16) with $N = N_{\mathrm{gs}}$ or $N = N_{\mathrm{jac}}$. As can be seen from Figure 5 and the discretization stencils in Figure 3, u^{m+1} depends only on the components of u^m corresponding to regions I_{r} and III_{r} in Figure 5. We write

$$u = \begin{bmatrix} w \\ v \end{bmatrix}, \tag{31}$$

with w the components of u in regions I_{nr}, III_{nr} and II and v the components of u in regions I_{r} and III_{r}. The injection operators Q_w and Q_v are defined as

$$Q_w^T u = w, \qquad Q_v^T u = v, \tag{32}$$

so that the fact that u^{m+1} depends only on v can be expressed as

$$(I - N^{-1}A)u = (I - N^{-1}A)Q_v v. \tag{33}$$

Furthermore, since $u = Q_w w + Q_v v$, it follows from (33) that

$$N^{-1}AQ_w w = Q_w w, \tag{34}$$

and therefore,

$$N^{-1}Au = N^{-1}AQ_w w + N^{-1}AQ_v v = Q_w w + N^{-1}AQ_v v = N^{-1}f. \tag{35}$$

[4] As mentioned before in Section 3, a GMRES method with appropriate preconditioners [45,44] is used to solve the subdomain problems (inner iteration), but this is independent of the GMRES method used for the outer iteration, which is explained in this section.

By multiplying the left- and right-hand-sides of (35) with $[Q_w \; Q_v]^T$ we get

$$\begin{bmatrix} Q_w^T \\ Q_v^T \end{bmatrix} N^{-1}Au = \begin{bmatrix} I & Q_w^T N^{-1}AQ_v \\ \varnothing & Q_v^T N^{-1}AQ_v \end{bmatrix} \begin{bmatrix} w \\ v \end{bmatrix} = \begin{bmatrix} Q_w^T \\ Q_v^T \end{bmatrix} N^{-1}f, \tag{36}$$

which proves the block form (30) with $R = Q_w^T N^{-1}AQ_v$ and $D = Q_v^T N^{-1}AQ_v$.

From the block form (30) we see that we can solve for the interface (or relevant) variables v_r independently of the other variables. This means that we can reduce the vector length in GMRES to the length of v_r, which only contains unknowns along the interface. It is precisely this fact that makes GMRES acceleration efficient. For a detailed description, see [16,17]. Table II shows the number of domain decomposition iterations and total time needed to solve a Poisson equation on a rectangle divided into 3 × 3 blocks. Clearly, the accelerated algorithm provides a significant speedup, and due to the reduction in vector length by using only v_r, the amount of additional work by the GMRES acceleration is negligible, see [16,17] for details.

5. NUMERICAL RESULTS

This section presents results for several flow problems in complex geometries. In all computations we have used a relative stopping criterion for solving the subdomain problems (with GMRES) and the domain decomposition acceleration (with GMRES):

$$\|r^m\| \leq \epsilon \|r^0\|, \tag{37}$$

with ϵ the required relative accuracy and r^k the residual vector after k steps of the GMRES algorithm. As mentioned earlier, the subdomains should be solved accurately, otherwise the reduction of vector length in GMRES to the number of interface variables v_r would not have been possible. In the reported experiments we have used the same relative accuracy of $\epsilon = 10^{-4}$ for solving subdomains as for the GMRES acceleration. This accuracy for the subdomains proved to be sufficient.

The stopping criterion used for the stationary solution is as follows. If u^m denotes a computed quantity (velocity field or pressure) at a certain time step, then a relative accuracy of ϵ has been reached for this quantity when the following holds:

Table II. Additional work in the accelerated algorithm

Subdomain grid		5 × 5	10 × 10	20 × 20	30 × 30	40 × 40	50 × 50
Unaccelerated	No. iterations	135	110	119	149	183	220
	Time	32	51	231	919	2727	7062
	Time/iteration	0.2	0.5	1.9	6.2	14.0	32.1
Accelerated	No. iterations	16	15	15	16	17	18
	Time	5	9	37	119	258	598
	Time/iteration	0.3	0.6	2.4	7.4	15.2	33.2

Computing time on a HP 9000/720 workstation measured in seconds.

$$\|u^{m+1}-u^m\|_\infty = \frac{1-\rho}{\rho}\,\epsilon\,\|u^{m+1}\|_{\mathrm{E}}, \tag{38}$$

where $\| * \|_\infty$ denotes the maximum norm and $\| * \|_{\mathrm{E}}$ denotes the Euclidean norm as before. The quantity ρ is defined by

$$\rho = \frac{\|u^{m+1}-u^m\|_{\mathrm{E}}}{\|u^m-u^{m-1}\|_{\mathrm{E}}}. \tag{39}$$

The computation stops when (38) is satisfied for both the momentum and pressure equations. For a derivation of this stopping criterion see [17] pp. 132–133.

Three different problems will be computed. The backward-facing step problem is discussed in Section 5.1. This problem is used to compare multi-block efficiency and accuracy with that of single-block solution. Section 5.2 shows that our multi-block method also functions well in the presence of skewed grids. Furthermore, it is a good test to verify the extrapolation procedure introduced in Section 3.2. We end with the problem of a cylinder in a wall-bounded shear flow in Section 5.3. This problem is only used to demonstrate the capability of our multi-block method to handle more complex geometries.

To obtain accurate results for these problems we use a central discretization, even at the higher Reynolds numbers. Numerical techniques that use artificial diffusion (e.g. upwinding) are less suited for these problems. Especially at higher Reynolds numbers, the artificial diffusion can become more significant with respect to the physical diffusion, which in turn can lead to inaccurate results (e.g. inaccurate reattachment lengths).

5.1. Flow over a backward-facing step

The backward-facing step problem is a long-standing benchmark to examine the accuracy of numerical methods. It models the phenomenon of flow separation of internal flows caused by a sudden expansion of the test-section geometry. Numerical results have been presented in [5,51,52]. A good paper containing experimental results is [53].

Figure 8 shows the backward-facing step geometry and boundary conditions. The flow geometry is taken as identical to that of [5] with expansion ration h:H = 1:2. The length of the

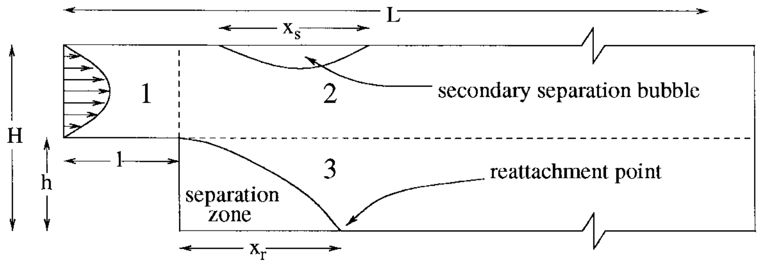


Figure 8. The backward facing step geometry and decomposition.

computational domain is $L = 30h$, and the length of the entrance section is $l = 6h$. A uniform grid of 130×66 is used for the part of the geometry excluding the entrance. The grid is extended uniformly to the entrance. On inflow, a parabolic inlet velocity profile is prescribed, and on outflow the tangential velocity and normal stress are zero. On the remaining boundaries, no-slip conditions are prescribed.

The following decompositions are used. The single-block problem is obtained by fitting regions 2 and 3 in Figure 8 with a single uniform grid. The multi-block problem without inlet consists of the two blocks defined by regions 2 and 3. The multi-block problem with inlet section modeled consists of all three blocks.

The Reynolds number is defined as in [53]:

$$Re = \frac{2\bar{v}h}{\nu}, \tag{40}$$

with $\bar{v}$ the average inlet velocity and $\nu = \mu/\rho$ the kinematic viscosity. Computations are carried out for $Re = 200$, 400, 600 and 800.

For all these Reynolds numbers, recirculation occurs directly after the inlet. The reattachment length, normalized by the step height $x'_r = x_r/h$ (see Figure 8) is the main result of the computation. It has been observed that at Reynolds numbers larger than 400 a secondary separation bubble appears at the top of the domain. We will also compute the normalized length of the secondary separation bubble $x'_s = x_s/h$. The locations were the flow separates or reattaches to the wall are obtained by looking at those positions on the boundary where the tangential stress $\mu\, \partial v_1/\partial x_2$ changes sign. This produces a result within an accuracy of the grid size $\mathcal{O}(\Delta x)$, so that for a reasonable grid this error should be small. The central discretization scheme we use is essential to obtain good results since it contains no artificial numerical smoothing. Especially at the higher Reynolds numbers this is important. Figure 9 shows the result of a multi-block computation at $Re = 600$ and clearly shows the two recirculation areas.

Computations will be done with and without the inlet channel modeled. Without the inlet channel present, we perform single- and multi-block computations to compare accuracy and efficiency. Multi-block solution with the inlet present is used to check the influence of more accurate geometrical modeling on the results.

As explained in the Section 1, we use a Euler time stepping scheme to compute the stationary solution. Even though this scheme is implicit, it is still not possible to take arbitrarily large time steps. This is because of the iterative solution methods used for the momentum equations. Therefore, we gradually increase the Reynolds number during computation as

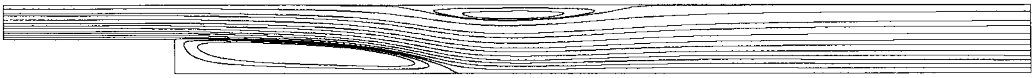


Figure 9. Streamlines of multi-block backward-facing step at $Re = 600$. Two recirculation areas are visible.

$$Re(t) = \max\{Re_{\text{final}}, 2 \cdot t\}. \tag{41}$$

In combination with a time step of 0.25, this proved to be sufficient to reach steady state. A relative stationary accuracy of 10^{-3} in (38) proved to be sufficient to reach steady state with sufficient accuracy.[5]

Table III shows the single- and multi-block results for the problem without inlet, and the numerical results of [5,51,52] (also obtained without inlet channel) and the experimental results of [53]. Our results are in good agreement. Furthermore, the single- and multi-block results also agree well. There are small differences between the single- and multi-block solutions that are probably caused by the grid stretching at the block interface. For Reynolds numbers larger than 400, the experimental results of [53] deviate somewhat from the numerical results. As noted in [53], the reason for this is probably that the actual (experimental) flow is no longer two-dimensional at these Reynolds numbers.

Table IV shows results for the secondary separation bubble. Again, good agreement is obtained between our results and those found in the literature. For both the secondary separation bubble and the reattachment length, small differences in the results occur between single- and multi-block solution. The reason can be that for determining the recirculation length one has to look for sign changes of the tangential stress along the boundary. With this procedure, even small numerical errors can cause a relatively large change in the measured reattachment lengths and secondary separation bubbles. Direct comparison of the single- and multi-block results showed that the multi-block velocity fields were always within less than 7% of the single-block solution.[6]

Table III. Reattachment lengths for the backward-facing step problem without the inlet

Re	Single	Multi	Figure 11, [5]	Figure 9, [52]	Figure 14, [53]	[51], p. 118
200	5.2	5.2	5.3	5.1	5.3	Not computed
400	8.5	8.4	8.3	8.5	8.7	8.7
600	10.6	10.3	10.2	10.6	11.9	10.7
800	12.0	11.8	11.7	11.8	14.3	11.9

Single- and multi-block results, and results by others are listed.

Table IV. Length of the secondary separation bubble without inlet

Re	Single	Multi	Figure 11, [5]	Figure 9, [52]	Figure 14, [53]
400	1.6	2.1	2.5	Not given	Not present
600	7.1	7.5	7.8	7.8	7.4
800	10.8	11.0	11.4	11.5	5.5

[5] We carried out several experiments by varying the stationary accuracy from 10^{-2} to 10^{-6} to verify this.

[6] The maximum difference occurred of course in those regions where the solution was very small in magnitude, where even small absolute differences produce large relative differences.

Table V. Reattachment lengths and secondary separation bubbles with and without inlet

Re	Single	Multi without inlet	Multi with inlet
Reattachment lengths			
200	5.2	5.2	4.9
400	8.5	8.4	7.9
600	10.6	10.3	9.8
800	12.0	11.8	11.1
Secondary separation bubbles			
400	1.6	2.1	3.4
600	7.1	7.5	7.9
800	10.8	11.0	11.6

To show the effect of more accurate geometric modeling on the results, Table V compares the results of multi-block solution with inlet with the results without inlet. Note that the reattachment lengths predicted with inlet taken into account are somewhat smaller than those computed without inlet. This was, as remarked in [5], first noted in [54]. The discrepancies between the present result and [54] can be attributed to the upstream influence of the inlet section. This causes the velocity profile at the expansion to deviate from a parabola, so that effectively the problem without inlet section modeled uses a slightly different inlet velocity field.

Finally, we compare the efficiency of the multi-block with the single-block solution. Table VI compares computational efficiency of the single-block with that of the multi-block solution. As can be seen, the multi-block solution requires two to three times as much computing time as the single-block solution.

There are several reasons for this. One reason is that the subdomain problems are solved accurately in each domain decomposition iteration. This was required for the reduction of the total system of equations to a system restricted to the unknowns at or near the block interfaces, see Section 4. Alternatively, one could omit the reduction of the system size to the interface equations and permit inaccurate solution of subdomains. This is the subject of our paper [55] (also discussed in [17] Chapters 6 and 7).

Table VI. Comparison of efficiency of single- and multi-block solution for Reynolds 600

	Time (min)	Momentum (%)	Pressure (%)
Single-block	199.2	24	22
Multi-block (no inlet)	504.5	34	56
Multi-block (inlet)	539.6	34	57

The total computing time is shown, together with the percentages of computing time solving the momentum and pressure equations respectively.

Another reason is that in our implementation, the subdomain Krylov solver is used as black box, which means that for each repeated solution of the same subdomain, the subdomain Krylov space must be rebuilt from the start. Significant optimizations can probably be made by saving the subdomain Krylov spaces between consecutive domain decomposition iterations.

5.2. Skewed cavity

In [15,56], the skewed cavity problem was defined for testing discretizations on non-orthogonal grids. The problem geometry is depicted in Figure 10. It is a parallelogram with boundary length $L = 1$. The angle β can be varied. We examine only test case 1 from [56] and take $\beta = 45°$. The top wall is moving with velocity $v_1 = 1$ to the right. On all other boundaries no-slip conditions are prescribed. The Reynolds number is defined as

$$Re = \frac{v_1 L}{v}. \tag{42}$$

The skewed cavity problem is used here to validate the extrapolation procedure at corners discussed in Section 3.2. For this purpose we will compare the single- and multi-block solutions at the centerlines CL_1 and CL_2. The multi-block problem uses four blocks with the interfaces CL_1 and CL_2. The problem is also used to compare efficiency of single- with multi-block solution again.

The solution to the skewed cavity problem is computed at Reynolds numbers 100 and 1000, with global grids of respectively 64 × 64 and 128 × 128 grids. A stationary accuracy of 10^{-2} is used, which proved to be sufficient.[7] All problems are computed with time step $\Delta t = 0.15$. As opposed to the backward-facing step problem, we do not increase the Reynolds number during computation but start immediately at $Re = 1000$.

To compare results with [15,56], we compute the minimum and maximum values of the streamfunction Ψ and the respective locations. As a measure of the accuracy of the solution on the finest grid, we compute the quantity

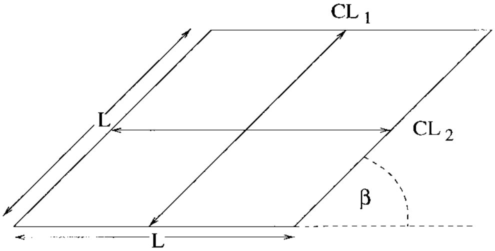


Figure 10. Geometry of the skewed cavity problem.

[7] We also carried out tests at larger stationary accuracies to verify this.

$$\delta = \left| \frac{\Psi_{128} - \Psi_{64}}{\Psi_{128}} \right| \cdot 100. \tag{43}$$

The quantity δ indicates the relative error percentage between solutions obtained on the finest grid of 128 × 128 cells and that of 64 × 64 cells.

Figure 11 shows the streamlines obtained on the 128 × 128 grid, which agree closely with those in [15,56]. The values of the streamlines are identical to those in [15,56].

A more detailed comparison is made by comparing the minimum and maximum values of the streamfunctions. These results are shown in Table VII. As can be seen from Table VII the single- and multi-block results agree closely. Furthermore, we have good agreement with the results of [15,56]. The accuracy measure δ is somewhat larger for our solution than for the solutions of [15,56]. This difference is probably caused by the fact that much finer grids were employed in these references: 320 × 320 and 256 × 256 grids respectively in [15,56].

To investigate the effect of the extrapolation procedure at corners introduced in Section 3.2, we plot the Cartesian velocity component v^1 at CL_1, and v^2 at CL_2 for multi-block and single-block solution, see Figure 12. As can be seen from this figure, single- and multi-block solutions agree excellently at the centerline. This verifies again that the extrapolation procedure can be applied safely.

Finally, we compare the efficiencies of multi-block and single-block solution again. Table VIII compares computing times for the single- and multi-block problems using 128 × 128 grid cells. The multi-block solution costs approximately four times as much time as single-block solution. Also, we see that, similar to the backward facing step problem, the pressure equations requires the most time with multi-block solution. Possible reasons for this are as explained in the previous section.

5.3. *Flow around a cylinder in a wall-bounded shear flow*

We compute the flow around a cylinder in a wall-bounded shear flow. The cylinder is situated in the viscous sublayer of a turbulent boundary layer, so the flow is laminar and the inlet velocity profile is linear.

Table VII. Minima and maxima of the streamfunction for the skewed cavity problem with $Re = 1000$ on different grids

	Single-block	Multi-block	Reference [56]	Reference [15]
$\Psi_{\min}$	$-5.3591 \cdot 10^{-2}$	$-5.3617 \cdot 10^{-2}$	$-5.3507 \cdot 10^{-2}$	$-5.3523 \cdot 10^{-2}$
x_1	1.3089	1.3089	1.3130	1.3128
x_2	0.5745	0.5745	0.5740	0.5745
δ	0.02	0.01	0.1	0.12
$\Psi_{\max}$	$9.9486 \cdot 10^{-3}$	$9.9887 \cdot 10^{-3}$	$1.0039 \cdot 10^{-2}$	$1.0039 \cdot 10^{-2}$
x_1	0.8173	0.8251	0.7766	0.7775
x_2	0.4033	0.4033	0.3985	0.4005
δ	1.3	0.8	0.33	0.15

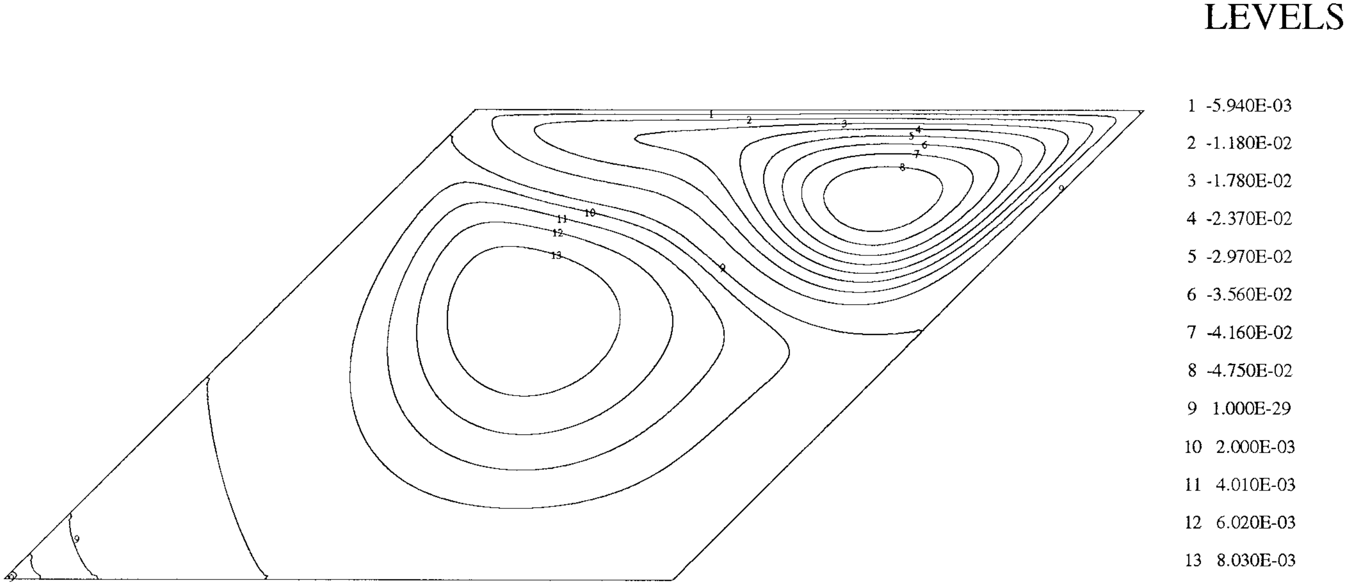


Figure 11. Streamlines for the skewed cavity problem at $Re = 1000$.

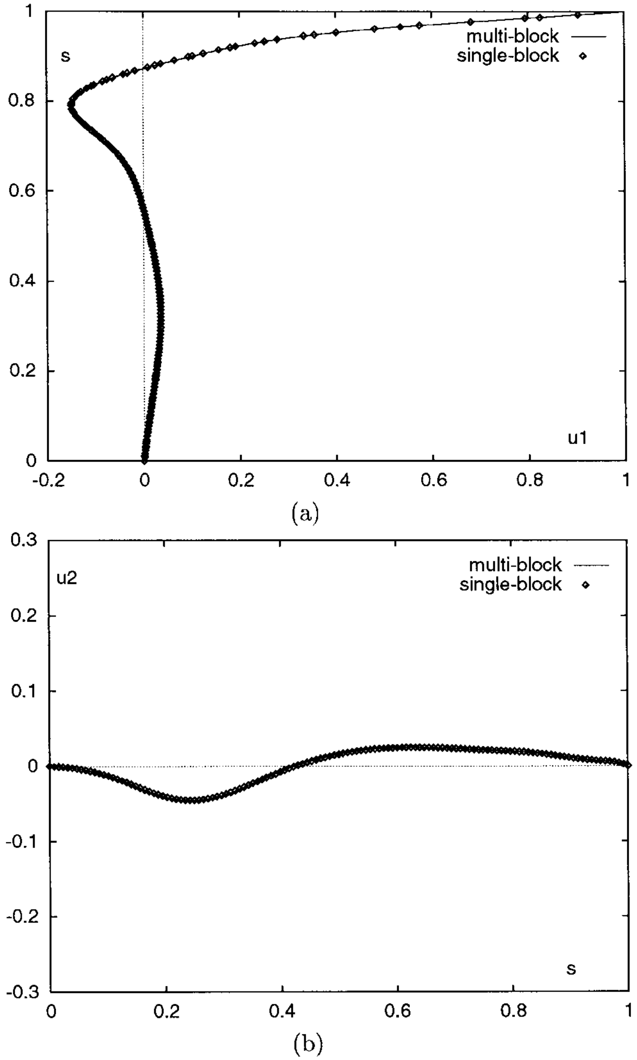


Figure 12. Cartesian velocity components for the single- and multi-block solution on the centerlines vs. arc-length s along the centerline: (a) u_1 on CL$_1$, (b) u_2 on CL$_2$.

This problem models the removal of particles from a surface. Examples of where this phenomenon occurs are the cleaning of surfaces by water jets, vacuum cleaners, and contamination of surfaces (for instance, in the disposal route of irradiated fuel of nuclear reactors). Therefore, this problem is of considerable practical interest.

From a numerical point of view, it is also interesting because both single-block and multi-block solutions can be computed and the grid is non-orthogonal. The emphasis lies in

Table VIII. Comparison of efficiency of single- and multi-block solution for the skewed cavity problem

	Time (min)	Momentum (%)	Pressure (%)
Single-block	73.8	26	56
Multi-block	290.2	18	79

The total computing time is shown, together with the percentages of computing time solving the momentum and pressure equations respectively.

comparing single- with multi-block solutions and verification of the multi-block algorithm (especially the extrapolation procedure described above). Figure 13 depicts the flow geometry. The cylinder has diameter a. The Reynolds number is defined as

$$Re = \frac{av^*}{v}, \tag{44}$$

with

$$v^* = \sqrt{\frac{\tau_0}{\rho}}, \tag{45}$$

where $\tau_0 = \mu\, \partial v_1/\partial x_2$ is the shear stress associated with the linear inlet velocity profile. Typical Reynolds numbers for this problem are $Re = 1-5$. Our results are given for $Re = 2$. In the computation we have used $L_1 = L_2 = H = 10$. The boundary conditions are given in Figure 13.

AC: $u = 0,\ v = 0$

AE: $u = \frac{\tau_0}{\mu} \cdot y,\ v = 0$

DC: $\sigma_{xx} = 0,\ v = 0$

ED: $\sigma_{xy} = \tau_0,\ v = 0$

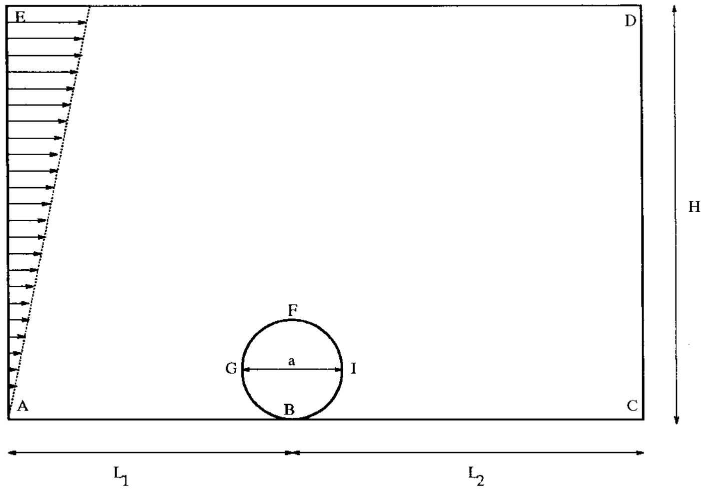


Figure 13. Geometry and boundary conditions of the cylinder problem.

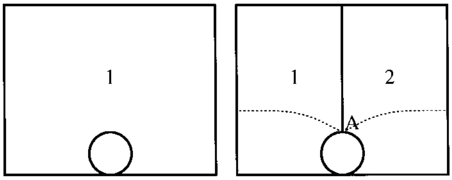


Figure 14. Two different decompositions of the domain.

Two different grids were used as shown in Figure 14: a single- and a multi-block grid. The two dashed lines in the second decomposition indicate additional block interfaces that were introduced during grid generation to obtain more grid control. So the grid of decomposition two was generated by using four blocks and after that joining the two blocks on the left and the two blocks on the right to a single block. This also means that the grid may be non-smooth across these dashed lines, because the grid is generated in the subdomains independently. Figure 15 shows the corresponding single- and multi-block grids that were used at grid refinement factor 1/2 (one-fourth of the number of grid cells at refinement factor 1).

The advantage of grid 1 is that it is a single-block grid that is well-adjusted to the cylinder. The disadvantage is that it is badly adjusted to the other external boundaries (there is a 90° kink in the grid lines at the top-left and top-right corners). The advantage of grid 2 is that it is well-adjusted to the external boundaries and is nearly Cartesian in large parts of the domain. The disadvantage is that there is a singularity in the grid at point A.

The stationary solution is computed using the implicit Euler time integration scheme with start time $t = 0$ and end time $t = 10$, with an appropriate time step. These time steps are given in Table IX. Figure 16 shows the computed velocity profile for the two grids at grid refinement factor 3. With grid 1 (single-block) we see that the velocity field is inaccurate at the top-right corner. This is probably due to the 90° kink in the grid lines at the top-right corner. The multi-block solutions do not have this problem because they are well-adjusted to the external boundaries.

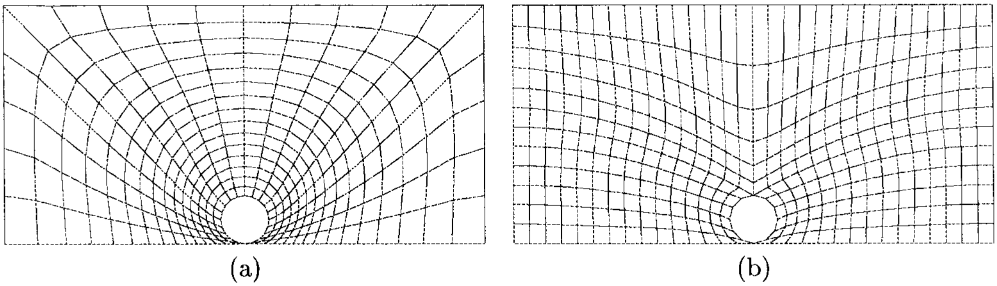


Figure 15. Single-block grid (a) and the multi-block grid (b) of two blocks.

Table IX. Results for the cylinder problem

Grid refinement	Min. ψ	Max. ψ	No. grid cells	Δt
Grid 1				
0.5	-0.72088 at (8.69, 1.81)	50.000	300	0.1
1	-0.03286 at (1.95, 0.87)	50.000	1200	0.1
2	-0.03159 at (1.97, 0.84)	50.000	4800	0.05
3	-0.03144 at (2.02, 0.86)	50.000	10 800	0.05
Grid 2				
0.5	-0.03288 at (1.78, 0.84)	50.000	360	0.1
1	-0.03062 at (2.24, 0.79)	50.000	1380	0.1
2	-0.02949 at (2.02, 0.86)	50.000	5400	0.05
3	-0.02932 at (1.98, 0.82)	50.000	12 240	0.02

Minimum and maximum streamfunction at different grid refinement factors.

A comparison of the maxima and minima of the streamfunction in Table IX shows that grids 1 and 2 agree reasonably well. The streamlines are depicted in Figure 17.

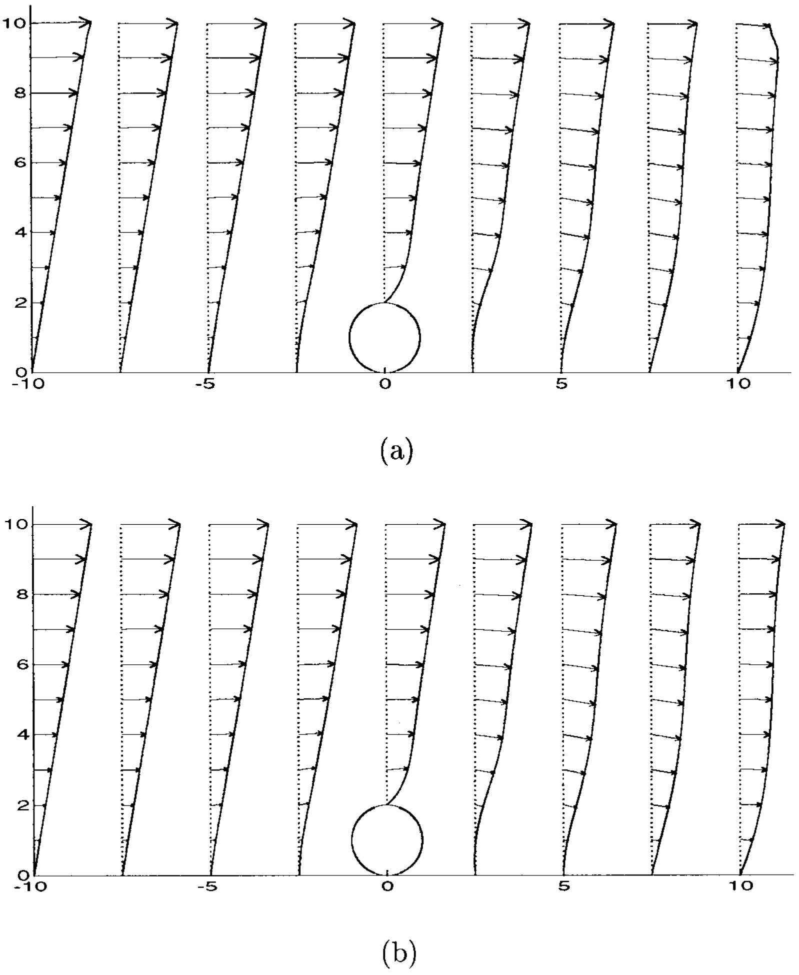
The isobars are shown in Figure 18. These plots show that the single-block solution of grid 1 is inaccurate near the top-left and top-right corners. This is probably due to the strong non-orthogonality at these corners combined with a change in type of boundary condition. This inaccuracy does not seem to propagate into the domain. The multi-block grid 2 does not have these problems at the corners because it is well-adapted to the external boundaries of the grid. However, there is an inaccuracy emanating from the top of the cylinder, following the grid discontinuity to the left and right.

To verify that the problem with the pressure on grid 2 is not caused by the extrapolation procedure that is performed at the top of the cylinder with the multi-block solution, we compute the flow only in block 1 using $\sigma_{x_1x_2}=0$ and $v_2=0$ on the former block interface. Figure 19 shows that there still is an inaccuracy of the pressure that follows the grid line emanating from the top. This confirms that the pressure inaccuracies observed with the multi-block solutions are not due to the extrapolation procedure but to non-smoothness of the grid.

The conclusion from the computation on these two grids is that to improve accuracy of the multi-block solution either grid generation or discretization should be improved. In the grid generation phase, the grids should not be generated separately in the subdomains because this gives non-smooth grids across block interfaces. Instead, the grid generation procedure should be extended to the entire domain, for instance, using domain decomposition. A general introduction to grid generation is [57], a more specialized work on multi-block grid generation is [58]. Another solution is to improve accuracy of the discretization by using a more advanced discretization scheme, see [9,10].

6. DISCUSSION/CONCLUSIONS

We have described a domain decomposition method for solving the incompressible Navier–Stokes equations. The discretization is based on a staggered grid. The pressure-correction



(a)

(b)

Figure 16. Velocity field: (a) grid 1, (b) grid 2.

scheme is used in combination with the θ method for time integration. The domain decomposition method extends the pressure-correction scheme to the composite domain. The momentum and pressure equations are solved separately but over the composite domain using a Schwarz-type domain decomposition method with minimal overlap.

A Krylov subspace (GMRES) method is used to accelerate convergence of the basic Schwarz domain decomposition method. This also makes the method more robust with respect to numbers of subdomains, number of grid points, and boundary conditions. The basic assumption that subdomain problems are solved accurately enables a reduction of the total system of equations to be solved (composite domain) to a system involving only unknowns at or near the block boundaries.

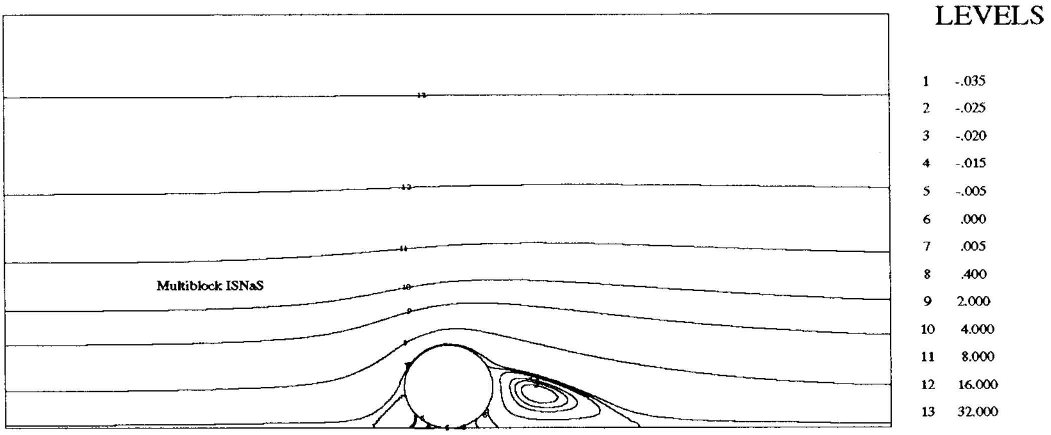


Figure 17. Streamlines using the multi-block grid 2.

The method is applied to three test problems: the backward-facing step problem, skewed cavity problem, and the problem of a cylinder in a wall-bounded shear flow. These problems show that the multi-block methods produces accurate results, also at higher Reynolds numbers, and for highly non-orthogonal grids. The results for the cylinder problem show that multi-block solution makes it much easier to obtain grids that are well-adapted to the external boundaries. As comparison with the single-block solution showed, this also improves accuracy of the resulting solution.

A problem with the current multi-block method is that it requires two to four times as much computing time as the single-block solution.[8] Several factors can be responsible for this. The first is the requirement of the current multi-block method that subdomain problems must be solved accurately. Another reason is that the iterative Krylov subdomain solvers throw away their Krylov spaces after the solution of the equations. Possibly, some improvements can be made by storing the subdomain Krylov subspaces between consecutive domain decomposition iterations, and re-using them.

The experiments show that, with multi-block, the solution of the pressure equation becomes the most expensive part of our Navier–Stokes solver. The momentum equations are relatively easy to solve, which is because of the $1/\Delta t$ term on the main diagonal of the momentum matrix, which improves convergence of the momentum equations.

Different methods can be used to reduce the total computation time of the domain decomposed Navier–Stokes solver with respect to the single-block solver. A first method is to lower the accuracy with which the multi-block problem in each time step is solved. This approach can only be used to compute the stationary solution. Preliminary experiments have

[8] It should be noted that in general flow problems, geometrically complex regions cannot be solved anymore using a single topologically rectangular block.

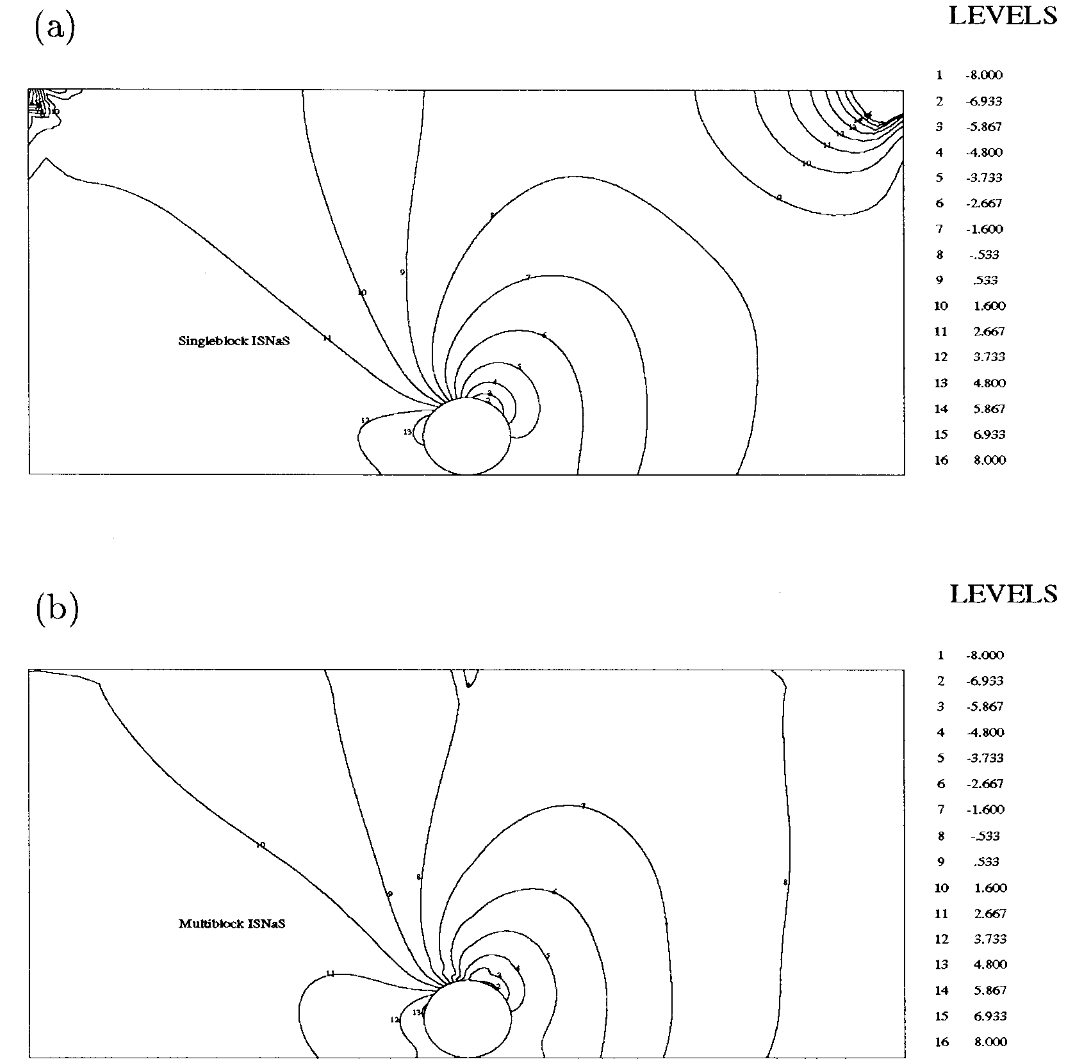


Figure 18. Isobars: (a) grid 1, (b) grid 2.

shown that this can approximately halve total computing time. The most promising method is to lower the accuracy with which subdomains are solved. This approach requires a different GCR [59] based acceleration technique. This method is the subject of our other paper [55].

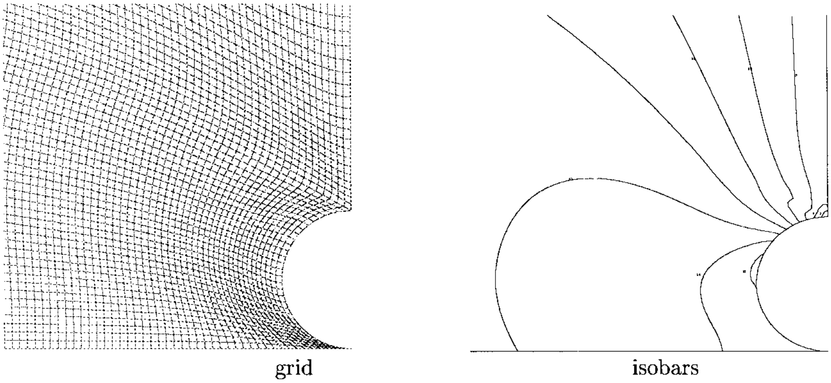


Figure 19. Flow across a half cylinder, single-block results.

REFERENCES

1. W. Rodi, S. Majumdar and B. Schönung, 'Finite volume methods for two-dimensional incompressible flows with complex boundaries', *Comput. Methods Appl. Mech. Eng.*, **75**, 369–392 (1989).
2. 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).
3. C.M. Rhie, 'A pressure-based Navier–Stokes solver using the multigrid method', *AIAA Paper 86-00207*, 1986.
4. M. Rosenfeld, D. Kwak and M. Vinokur, 'A fractional step solution method for the unsteady incompressible Navier–Stokes equations in generalized co-ordinate systems', *J. Comput. Phys.*, **94**, 102–137 (1991).
5. Y. Zang, R.L. Street and J.R. Koseff, 'A non-staggered grid, fractional step method for time-dependent incompressible Navier–Stokes equations in curvilinear co-ordinates', *J. Comput. Phys.*, **114**, 18–33 (1994).
6. C.W. Oosterlee, 'Robust multi-grid methods for the steady and unsteady incompressible Navier–Stokes equations', *Ph.D. Thesis*, Delft University of Technology, 1993.
7. P. Wesseling, C.G.M. Kassels, C.W. Oosterlee, A. Segal, C. Vuik, S. Zeng and M. Zijlema, 'Computing incompressible flows in general domains', in F.-K. Hebeker, R. Rannacher and G. Wittum (eds.), *Numerical Methods for the Navier–Stokes Equations*, Vieweg, Wiesbaden, 1994, pp. 298–314.
8. P. Wesseling, A. Segal, J. van Kan, C.W. Oosterlee and C.G.M. Kassels, 'Invariant discretization of the incompressible Navier–Stokes equations in general co-ordinates on staggered grids', *Comput. Fluids Dyn. J.*, **1**, 27–33 (1992).
9. P. van Beek, R.R.P. van Nooyen and P. Wesseling, 'Accurate discretization of gradients on non-uniform curvilinear staggered grids', *J. Comput. Phys.*, **117**, 364–367 (1995).
10. P. Wesseling, P. van Beek and R.R.P. van Nooyen, 'Aspects of non-smoothness in flow computations', in A. Peters, G. Wittum, B. Herrling, U. Meissner, C.A. Brebber, W.G. Gray and G.F. Pinder (eds.), *Computational Methods in Water Resources X*, Kluwer, Dordrecht, 1994, pp. 1263–1271.
11. M. Zijlema, A. Segal and P. Wesseling, 'Invariant discretization of the k–ε model in general co-ordinates for prediction of turbulent flows in complicated geometries', *Comput. Fluids*, **24**, 209–225 (1995).
12. M. Zijlema, A. Segal and P. Wesseling, 'Finite volume computation of incompressible turbulent flows in general co-ordinates on staggered grids', *Int. J. Numer. Methods Fluids*, **20**, 621–640 (1995).
13. M. Zijlema and P. Wesseling, 'On accurate discretization of turbulence transport equations in general co-ordinates', in C. Taylor and P. Durbetaki (eds.), *Numerical Methods in Laminar and Turbulent Flow; Proc. 9th Int. Conf. on Numerical Mathematical Laminar and Turbulent Flow*, Swansea, UK, vol. 9, part I, Pineridge Press, 1995, pp. 34–45.
14. A. Segal, P. Wesseling, J.J.I.M. 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. Numer. Methods Fluids*, **15**, 411–426 (1992).
15. C.W. Oosterlee, P. Wesseling, A. Segal and E. Brakkee, 'Benchmark solutions for the incompressible Navier–Stokes equations in general co-ordinates on staggered grids', *Int. J. Numer. Methods Fluids*, **17**, 301–321 (1993).

16. E. Brakkee and P. Wilders, 'The influence of interface conditions on convergence of Krylov–Schwarz domain decomposition for the advection–diffusion equation', *J. Sci. Comput.*, to appear (1997).
17. E. Brakkee, 'Domain decomposition for the incompressible Navier–Stokes equations', *Ph.D. Dissertation*, Delft University of Technology, Faculty of Applied Mathematics and Informatics, 1996. ftp://ftp.twi.tudelft.nl/TWI/publications/dissertations/E.Brakkee.ps.gz.
18. H.A. Schwarz, *Gesammelte Mathematische Abhandlungen*, vol. 2, Springer, Berlin, 1890, pp. 133–143. First published in *Viertelsjahrschrift der Naturforschenden Gesellschaft in Zürich*, **15**, 272–286 (1870).
19. P.L. Lions, 'On the Schwarz alternating method, I.', in R. Glowinski, G.H. Golub, G.A. Meurant and J. Périaux (eds.), *1st Int. Symp. Domain Decomposition Methods for Partial Differential Equations*, SIAM, Philadelphia, 1988, pp. 1–42.
20. R.F. van der Wijngaart, 'Composite-grid techniques and adaptive mesh refinement in computational fluid dynamics', *Ph.D. Thesis*, Stanford University, 1990.
21. Wei Pai Tang, 'Generalized Schwarz splittings', *SIAM J. Sci. Stat. Comput.*, **13**, 573–595 (1992).
22. K.H. Tan and M.J.A. Borsboom, 'Problem-dependant optimization of flexible couplings in domain decomposition methods, with an application to advection-dominated problems', *Preprint*, University of Utrecht and Delft Hydraulics, October 1993.
23. C.Y. Perng and R.L. Street, 'A coupled multigrid-domain splitting technique for simulating incompressible flows in geometrically complex domains', *Int. J. Numer. Methods Fluids*, **13**, 269–286 (1991).
24. W.D. Henshaw and G. Chesshire, 'Multigrid on composite meshes', *SIAM J. Sci. Stat. Comp.*, **8**, 914–923 (1987).
25. M. Hinatsu and J.H. Ferziger, 'Numerical computation of unsteady incompressible flow in complex geometry using a composite multigrid technique', *Int. J. Numer. Methods Fluids*, **13**, 971–997 (1991).
26. J.C. Strikwerda and C.D. Scarbnick, 'A domain decomposition method for incompressible flow', *SIAM J. Sci. Comput.*, **14**, 49–67 (1993).
27. J.A. Wright and W. Shyy, 'A pressure-based composite grid method for the Navier–Stokes equations', *J. Comput. Phys.*, **107**, 225–238 (1993).
28. Y. Zang, R.L. Street and J.R. Koseff, 'Three-dimensional composite-multigrid computations of incompressible flows in complex geometries', in Y.K. Cheung, J.H.W. Lee and A.Y.T. Leung (eds.), *Computational Mechanics; Proceedings of the Asian Pacific Conference on Computational Mechanics*, *Hong Kong*, *11–13 December 1991*, vol. 2, Balkema, Rotterdam, 1991, pp. 1379–1384.
29. Y. Zang, R.L. Street and J.R. Koseff, 'A composite-multigrid method for solving unsteady incompressible Navier–Stokes equations in complex geometries', in C. Taylor, P.M. Gresho and R.L. Sani (eds.), *Numerical Methods in Laminar and Turbulent Flow. Proceedings of the Seventh International Conference*, *15–19 July*, *1991*, vol. VII, part II, Pineridge, Swansea, 1991, pp. 1485–1495.
30. Y. Zang and R.L. Street, 'A composite multigrid method for calculating unstead incompressible flows in geometrically complex domains', *Int. J. Numer. Methods Fluids*, **20**, 341–361 (1995).
31. R.L. Meakin, 'Application of boundary conforming co-ordinate and domain decomposition principles to environmental flows', *Ph.D. Thesis*, Stanford University, 1986.
32. R.L. Meakin and R.L. Sreet, 'Simulation of environmental flow problems in geometrically complex domains. Part II: a domain splitting method', *Comput. Methods Appl. Mech. Eng.*, **68**, 311–331 (1988).
33. F. Durst and M. Schäfer, 'A parallel block-structured multi-grid method for the prediction of incompressible flows', *Int. J. Numer. Methods Fluids*, **22**, 549–565 (1996).
34. Y. Saad and M.H. Schultz, 'GMRES: a generalized minimal residual algorithm for solving non-symmetric linear systems', *SIAM J. Sci. Stat. Comput.*, **7**, 856–869 (1986).
35. C. Vuik, 'Solution of the discretized incompressible Navier–Stokes equations with the GMRES method', *Int. J. Numer. Methods Fluids*, **16**, 507–523 (1993).
36. W.D. Gropp and B.F. Smith, 'Experiences with domain decomposition in three dimensions: overlapping Schwarz methods', in A. Quarteroni, J. Périaux, Yu.A. Kuznetsov and O.B. Widlund (eds.), *Proc. 6th Int. Symp. on Domain Decomposition Methods in Science and Engineering*, AMS, Providence, 1992, pp. 323–333.
37. M. Dryja and O.B. Widlund, 'Some recent results on Schwarz-type domain decomposition algorithms', in A. Quarteroni, J. Périaux, Yu.A. Kuznetsov and O.B. Widlund (eds.), *Proc. 6th Int. Symp. on Domain Decomposition Methods in Science and Engineering*, AMS, Providence, 1992, pp. 53–61.
38. X.-C. Cai, W.D. Gropp and D.E. Keyes, 'A comparison of some domain decomposition and ILU preconditioned iterative methods for non-symmetric elliptic problems', *Numer. Linear Algebra Appl.*, **1**, 477–504 (1994).
39. A.E. Mynett, P. Wesseling, A. Segal and C.G.M. Kassels, 'The ISNaS incompressible Navier–Stokes solver: invariant discretization', *Appl. Sci. Res.*, **48**, 175–191 (1991).
40. F.H. Harlow and J.E. Welch, 'Numerical calculation of time-dependent viscous incompressible flow of fluid with a free surface', *Phys. Fluids*, **8**, 2182–2189 (1965).
41. A.J. Chorin, 'Numerical solution of the Navier–Stokes equations', *Math. Comput.*, **22**, 745–762 (1968).

42. 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).
43. C. Vuik, 'Fast iterative solvers for the discretized incompressible Navier–Stokes equations', *Reports of the Faculty of Technical Mathematics and Informatics 93–98*, Delft University of Technology, Delft, 1993. Available from anonymous ftp://ftp.twi.tudelft.nl/TWI/publications/tech-reports/1993/DUT-TWI-93-98.ps.gz. *Int. J. Numer. Methods Fluids*, to appear.
44. M.J. Berger, 'On conservation at grid interfaces', *SIAM J. Numer. Anal.*, **24**, 967–984 (1987).
45. P.E. Bjørstad and O.B. Widlund, 'Iterative methods for the solution of elliptic problems on regions partitioned into substructures', *SIAM J. Numer. Anal.*, **23**, 1097–1120 (1986).
46. R. Glowinski, G.H. Golub, G.A. Meurant and J. Périaux (eds.), *1st Int. Symp. on Domain Decomposition Methods for Partial Differential Equations*, SIAM, Philadelphia, 1988.
47. T.F. Chan, R. Glowinski, J. Périaux and O.B. Widlund (eds.), *Proc. 2nd Int. Symp. on Domain Decomposition methods*, SIAM, Philadelphia, 1989.
48. T.F. Chan, R. Glowinski, J. Périaux and O.B. Widlund (eds.), *Proc. 3rd Int. Symp. on Domain Decomposition Methods for Partial Differential Equations*, SIAM, Philadelphia, 1990.
49. R. Glowinski, Yu.A. Kuznetsov, G. Meurant, J. Périaux and O.B. Widlund (eds.), *Proc. of the 4th Int. Symp. on Domain Decomposition Methods for Partial Differential Equations*, SIAM, Philadelphia, 1991.
50. D.E. Keyes, T.F. Chan, G. Meurant, J.S. Scroggs and R.G. Voigt (eds.), *Proc. 5th Int. Symp. on Domain Decomposition Methods for Partial Differential Equations*, SIAM, Philadelphia, 1992.
51. M.C. Thompson and J.H. Ferziger, 'An adaptive multigrid technique for the incompressible Navier–Stokes equations', *J. Comp. Phys.*, **82**, 94–121 (1989).
52. J. Kim and P. Moin, 'Application of a fractional-step method to incompressible Navier–Stokes methods', *J. Comput. Phys.*, **59**, 308–323 (1985).
53. B.F. Armaly, F. Durst, J.C.F. Pereira and B. Schönung, 'Experimental and theoretical investigation of backward-facing step flow', *J. Fluid Mech.*, **127**, 473–396 (1983).
54. C.Y. Perng, *Ph.D. Dissertation*, Department of Mechanical Engineering, Stanford University, 1990 (unpublished).
55. E. Brakkee, C. Vuik and P. Wesseling, 'Domain decomposition for the incompressible Navier–Stokes equations: solving subdomain problems accurately and inaccurately', *Int. J. Numer. Methods Fluids*, **26**, 1217–1237 (1998).
56. I. Demirdžicand, Ž. Lilek and M. Perić, 'Fluid flow and heat transfer test problems for non-orthogonal grids: benchmark solutions', *Int. J. Numer. Methods Fluids*, **15**, 329–354 (1992).
57. J.F. Thompson and Z.U.A. Warsi, 'Boundary fitted co-ordinate systems for numerical solution of partial differential equations', *J. Comput. Phys.*, **47**, 1–108 (1982).
58. N.P. Weatherill, M.J. Marchant and D.A. King, 'Multiblock grid generation: results of the EC/BRITE-EURAM EUROMESH, 1990–1992', *Notes on Numerical Fluid Mechanics*, vol. 44, Vieweg, Wiesbaden, 1993.
59. S.C. Eisenstat, H.C. Elman and M.H. Schultz, 'Variational iterative methods for non-symmetric systems of linear equations', *SIAM J. Numer. Anal.*, **20**, 345–357 (1983).