Robert Cimrman
Domain decomposition methods for solving the Burgers equation

DOMAIN DECOMPOSITION METHODS FOR SOLVING
THE BURGERS EQUATION

Robert Cimrman, Plzeň

Abstract. This article presents some results of numerical tests of solving the two-dimensional non-linear unsteady viscous Burgers equation. We have compared the known convergence and parallel performance properties of the additive Schwarz domain decomposition method with or without a coarse grid for the model Poisson problem with those obtained by experiments for the Burgers problem.

Keywords: domain decomposition, multilevel methods, fluid mechanics, Burgers equation

MSC 2000: 65M55, 76D99

1. Introduction

The numerical resolution of most of the problems is equivalent to solving large systems of linear algebraic equations with sparse matrices with high condition numbers. Their size prevents them from being solved by direct solvers and their condition numbers complicate a usage of iterative solvers. The *domain decomposition methods* can solve this dilemma—they are a hybrid class of methods staying between the direct and the iterative approach. Moreover, they have a built-in parallelism and thus they meet perfectly the demands on the parallel performance.

A great scientific effort in this field has led to many performance and convergence estimates for linear elliptic PDEs (stationary problems)—see, for example, [5] or [3]. However, in many areas of the research, for example in the fluid dynamics, the mathematical description is evolutionary (involves time) and non-linear.

The goal of this work was to perform various numerical tests for a two-dimensional non-linear unsteady viscous system of two equations called the *Burgers system*, which appears to be a decent model system for the fluid dynamics description. The same tests were run for the model Poisson equation too, to "calibrate" the solver used. Then the corresponding results were compared and a conclusion was made about

possible convergence properties of the particular domain decomposition method which was used for the Burgers problem.

We must emphasize here that there are *no theoretical results* yet, describing the convergence of the domain decomposition methods in connection with the Burgers problem. Thus this work could be used as a starting point of some deeper research which might lead to convergence estimates.

2. Test Problems

Let us denote by $\boldsymbol{x} = (x, y)$ the space coordinates and by t the time. Then we describe a fluid flow by functions $u = u(\boldsymbol{x}, t)$, $v = v(\boldsymbol{x}, t)$, $p = p(\boldsymbol{x}, t)$, $f_u = f_u(\boldsymbol{x}, t)$, $f_v = f_v(\boldsymbol{x}, t)$, where u, v are velocity components in x and y directions respectively, p is a pressure and f_u, f_v are body forces. Let ϱ be a mass density, η a kinematic viscosity. A two-dimensional incompressible flow with constant properties in rectangular coordinates in the Eulerian frame of reference can be described by the set of equations (see e.g. [4])

$$
(1) \qquad \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} - \eta \Delta u = -\frac{1}{\varrho} \frac{\partial p}{\partial x} + f_u,
$$

$$
(2) \qquad \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} - \eta \Delta v = -\frac{1}{\varrho} \frac{\partial p}{\partial y} + f_v,
$$

$$
(3) \qquad \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0.
$$

(1) and (2) are the Navier-Stokes equations (momentum equations), (3) is the continuity equation.

The Burgers equation is a decent model for studying the properties of the Navier-Stokes equations, like shock-waves and turbulences. In the subsequent sections we will deal with the two-dimensional unsteady viscous Burgers equation in the Cartesian coordinates in the bounded simple domain Ω for the time interval $[0, T]$. The Burgers system resembles the Navier-Stokes system (1,2)—it is non-linear and has both the advection and the diffusion terms. However, the pressure terms are missing. It means, in fact, that we assume the pressure to be kept constant. Therefore the continuity equation (3) is omitted as well—the pressure changes are eliminated by a non-zero divergence (i.e. the appearance or disappearance of new matter particles). Another difference lies in more general coefficients of the diffusion terms—ν_x and ν_y are in our case possibly non-continuous functions of $(\boldsymbol{x}, t)$.

Let Ω be a unit square domain $]0, 1[\times]0, 1[$ and $\overline{\Omega}$ its enclosure. On this domain we want to solve the following Burgers system with Dirichlet boundary conditions

on the boundary $\partial\Omega$ in the time interval $[0, T]$:

$$(4) \qquad \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} - \nabla \cdot (\nu_x \nabla u) = f_u,$$

$$(5) \qquad \frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} - \nabla \cdot (\nu_y \nabla v) = f_v$$

$$\text{for } \boldsymbol{x} \in \Omega, \; t \in [0, T],$$

$$(6) \qquad u(\boldsymbol{x}, t) = g_u(\boldsymbol{x}, t), \quad v(\boldsymbol{x}, t) = g_v(\boldsymbol{x}, t)$$

$$\text{for } \boldsymbol{x} \in \partial\Omega, \; t \in [0, T],$$

$$(7) \qquad u(\boldsymbol{x}, 0) = u_0(\boldsymbol{x}), \quad v(\boldsymbol{x}, 0) = v_0(\boldsymbol{x}) \; \text{ for } \boldsymbol{x} \in \Omega,$$

where u_0, v_0, f_u, f_v, g_u, g_v are given sufficiently regular functions of space and time variables, ν_x and ν_y are given possibly non-smooth functions of space and time variables. We suppose u and v to be smooth enough for the above relations to have sense.

The Poisson problem

$$(8) \qquad -\nabla \cdot (\nu \nabla u) = f \quad \text{for } \boldsymbol{x} \in \Omega,$$

$$u = g \quad \text{for } \boldsymbol{x} \in \partial\Omega$$

has been used for comparing the results, since its convergence properties are well known, unlike those of the Burgers problem. The coefficient ν is a given possibly non-smooth function of space and time variables chosen so that its properties correspond to those of ν_x, ν_y of the Burgers problem.

A numerical resolution of the systems mentioned above consists usually of two steps. The first step is a discretization of the original system. Its result is a corresponding system of algebraic equations. The second step involves solving the resulting system numerically. Basic methods accomplishing the first step are the finite difference method (FDM), the finite element method (FEM) and the finite volume method (FVM). We have chosen the FVM for its conservative properties.

Our discretization procedure for the Burgers problem was as follows: First we divided the computational domain into regular rectangular disjoint elements called *finite volumes*. Then we integrated the equations (4,5) over all finite volumes. After that we used the Rothe method to discretize the integral equations with respect to time and linearized the advection terms. In the end we replaced the integrals and derivatives by numerical formulas. As a result we obtained a system of linear algebraic equations for every discrete time step.

Let x_i, y_i for $i = 0, \ldots, N-1$ be the coordinates of the centers of the finite volumes (*nodes*). After denoting $u_i := u(x_i, y_i)$, $v_i := v(x_i, y_i)$ we can form vectors

$\hat{u} = [u_0, \ldots, u_{N-1}]^T$ and $\hat{v} = [v_0, \ldots, v_{N-1}]^T$. The corresponding nodal values of the right-hand side functions are stored in vectors $\hat{f}_u$ and $\hat{f}_v$. Then the discretized Burgers system has the following form in all time steps:

$$(9) \qquad \underbrace{\begin{bmatrix} A_u & 0 \\ 0 & A_v \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} \hat{u} \\ \hat{v} \end{bmatrix}}_{x} = \underbrace{\begin{bmatrix} \hat{f}_u \\ \hat{f}_v \end{bmatrix}}_{b},$$

where A_u and A_v are pentadiagonal ($\Leftarrow$ space discretization scheme) $N \times N$ matrices. The equations for $\hat{u}$ and $\hat{v}$ are mutually independent in our case because of the advection terms' linearization—we considered the non-differentiated variables of the advection terms to be known from the previous time step. Nevertheless, we could obtain also an interconnected system for a different treatment of the advection terms. The matrix of the system (9) is non-symmetric in the general case.

3. Domain Decomposition Methods

The domain decomposition methods (DDM) were originally meant for solving elliptic boundary value problems on complicated domains by solving the same problem restricted to simpler subdomains of the original domain (Schwarz, 1870). In their discrete versions they can be used to efficiently solve large sparse systems of linear equations arising from a discretization of elliptic problems. Their convergence properties depend on the mutual overlap δ and the size H of the subdomains. They are independent on the discretization parameter h, i.e. on the size of the discrete system being solved. See Figure 1.

A detailed description of multiplicative and additive Schwarz domain decomposition methods and of their usage as preconditioners can be found in [5]. Other useful information can be found in [3] or [1].

It is easy to accelerate the DDM by the so-called *Krylov subspace methods* (for example the conjugate gradient method for symmetric positive definite matrices, GMRES, BiCGSTAB for non-symmetric ones). The DDM are then used as preconditioners of the Krylov methods. We have used the BiCGSTAB method (cf. [6]) preconditioned by the additive Schwarz method with or without a *coarse grid*.

This combination has the following properties, as concerns the model Poisson problem:

- The number of iterations is proportional to the square root of the mutual condition number $\text{cond}(BA)$, where B is the additive or multiplicative Schwarz preconditioner.

- The additive Schwarz method satisfies

$$(10) \qquad \mathrm{cond}(BA) \leqslant C_1(1 + \max_i k_i)\Big(1 + C_2 \max_i \frac{1}{H_i^2}\frac{H_i}{\delta_i}\Big),$$

where k_i is the number of neighbours of the subdomain Ω_i, $H_i = O(H)$ its diameter and $\delta_i = O(\delta)$ the minimum overlap between Ω_i and $\bigcup_{j \neq i} \Omega_j$. Since $1/H$ is proportional to the number of subdomains, the convergence gets worse when we increase the number of subdomains and keep the relative overlap δ/H constant. As the number of subdomains is equal to the number of parallel tasks in our case, this convergence property is very unpleasant, because we want to use as many processors as possible.

- The additive Schwarz method with a coarse grid satisfies

$$(11) \qquad \mathrm{cond}(BA) \leqslant C_1'(1 + \max_i k_i)\Big(1 + C_2 \max_i \frac{H_i}{\delta_i}\Big).$$

We can see here that the unwanted dependence on the number of subdomains has disappeared. This is the reason of adding the coarse grids to the Schwarz domain decomposition methods. The methods obtained are then called *multi-level domain decomposition methods*.
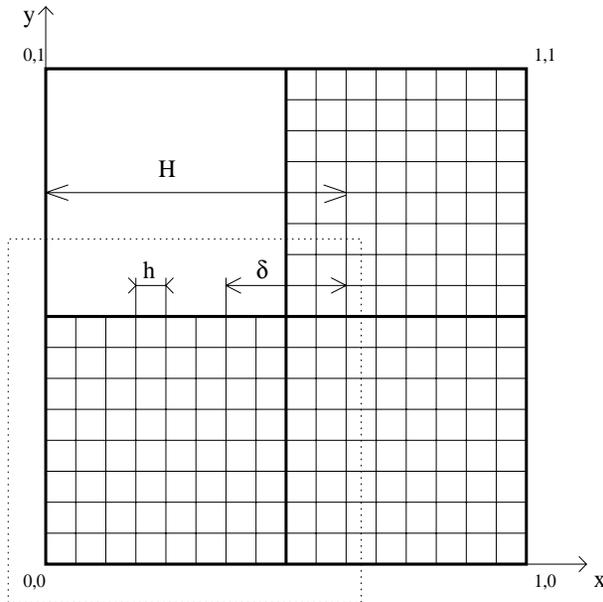
Figure 1. Our domain and its parameters.

4. Software Equipment

All computations were made with help of our own software called D^3S (Distributed Domain Decomposition Solver). It is tailored for distributed memory machines—it is built over the PVM message passing interface. It is based on the philosophy of UN-COL (United Communication Library) proposed by Kortas and Angot in [2], which is an efficient and portable library for parallel programming. The current version of our C program contains a parallel implementation of the BiCGSTAB method, preconditioned by the additive Schwarz method combined either additively or multiplicatively with a coarse grid solver. The restriction and prolongation operators used are described below in Section 5.

Also scripts for the *gnuplot* (a free Unix drawing program) were created to visualize the results. To allow a pseudo-dynamical visualization of the 2D fluid flow, a simple visualization tool called PFV (Particle Flow Visualization) was developed. This program animates a particle flow on a rectangular domain described by a velocity vector field. It displays both the particles flowing from outer regions over the border and the particles born in inner divergence sources. The latter ones are devoured by inner divergence holes. It is written in C++ and requires the Qt GUI toolkit, version 1.30 or higher.

5. Restriction and Prolongation Operators in D^3S

To remove the unwanted dependence on the number of subdomains we used a global coarse grid correction. The coarse grid was nested into the fine grid (i.e. all coarse grid points were fine grid points in some subdomain as well) to simplify the coarse—fine grid mappings. However, it was global and independent on subdomains. We have used two simple restriction operators—the trivial (T) restriction and the *residual conservation* (RC) restriction operator. The T operator simply takes the value of the fine grid node and maps it to the coarse grid node with the same coordinates. The RC operator is based upon the relation

$$(12) \qquad \int\limits_{\mathcal{V}_H(M)} r_H \, \mathrm{d}\mathcal{V} = \int\limits_{\mathcal{V}_H(M)} r_h \, \mathrm{d}\mathcal{V},$$

where $\mathcal{V}_H(M)$ is a coarse grid finite volume (a rectangle) around the node M, r_h is the fine grid residual and r_H is the coarse grid residual (the coarse volume contains several fine volumes). We denote the sizes of $\mathcal{V}_H$ by H_x, H_y and the sizes of $\mathcal{V}_h$ by h_x, h_y. See Figure 2. After the approximation of the integrals in (12) we get the

relation defining the RC operator:

$$(13) \qquad r_H(M) = \frac{1}{H_x H_y} \sum_{\mathcal{V}_{hi} \in \mathcal{V}_H} r_h(i) h_x h_y.$$

Here i means the fine node in the fine volume $\mathcal{V}_{hi}$. Our prolongation operator from the coarse grid back to the fine grid was the bilinear interpolation.
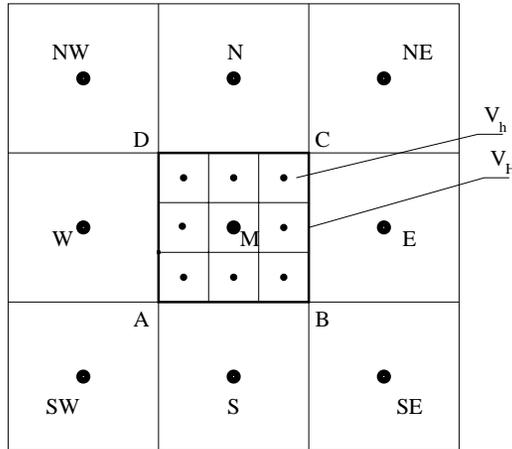
Figure 2. Coarse and fine volumes.

6. Performed Tests

All tests were run under the following conditions:

- Our domain was the unit square $(0 < x,\, y < 1)$ (see Figure 1).
- The convergence was declared when the initial residual was reduced by the factor 10^{-8} in the classical Euclidean (l^2) norm.
- The subdomain solutions were pseudo-exact. We used for them again the BiCGSTAB method. We required the reduction of the initial residual by 10^{-10} in this case.
- The RC operator was used for the restriction to the coarse grid.
- The computer used was *AlphaServer 8400 5/300 TurboLaser* with eight CPUs Alpha 21164 on 300 MHz. Since it had eight processors, the tests for more than 8 subdomains were run, in fact, in pseudo-parallel. That is why we did not include the elapsed times into our results.
- The programs were compiled using the *gcc 2.7.2.3* compiler with -O2 flag.

We will not write the actual right-hand sides here. They were chosen so that we obtained the test solutions presented below.

The test solution for the Burgers problem was a dipole diagonally crossing the domain. It can be described by the set of equations

$$u(x, y) = 10(t - y) \exp\left(-1/\nu_x((t - x)^2 + (t - y)^2)\right),$$
$$v(x, y) = 10(x - t) \exp\left(-1/\nu_y((t - x)^2 + (t - y)^2)\right).$$

The coefficients ν_x, ν_y from (4,5) were equal to 1 (homogeneous case), unless specified otherwise. The Dirichlet boundary conditions were created according to the test solution. We used zeros or the test solution at the initial time as the initial condition. The average convergence properties for all time steps did not differ for these two cases. The latter option is indicated explicitly whenever applied. The tests were run for the time interval $[0, 0.05]$ with the step 0.01. To use more time steps was not required—the number of iterations was always almost the same. The average results of all time steps are shown. The initial guess in every time step was a vector filled with ones.

For the Poisson problem we have used an exponential test solution

$$u(x, y) = \exp\left(2x + 2y\right).$$

The coefficient ν from (8) was equal to 1 (homogeneous case), unless specified otherwise. The Dirichlet boundary conditions were created according to the test solution.

6.1. Dependence on $1/H$

6.1.1. Description. Recall that h is the discretization parameter and H is the subdomain diameter. See Section 3 and Figure 1. According to the estimate of the condition number of the additive Schwarz method without the coarse grid (10), the number of iterations should not depend on h and should depend linearly on $1/H$, when the relative overlap δ/H is kept constant. On the other hand, when the coarse grid is used, the number of iterations should not depend on the number of subdomains ($\sim 1/H$) (see the estimate (11)). We measure the mesh size in $1/h$ units. For example, the mesh size 72 means that there are 72 nodes per side of the unit square domain.

This test was run with the following parameters: The relative overlap was $\delta/H = 44.44\%$. The coarse grid preconditioner (CGP) was combined with the additive Schwarz preconditioner multiplicatively. It is denoted "add-mul" in the tables. There were 81 fine nodes per one coarse node, i.e. one coarse finite volume contained a 9×9 cluster of fine finite volumes.

6.1.2. Results.

No. of subdomains	Mesh size $(1/h)$	No. of iterations without CGP	No. of iterations with CGP
2	72	6	7
4	72	9	8
8	72	11	9
16	72	17	8
2	144	5	7
4	144	9	9
8	144	10	10
16	144	15	9
2	216	5	6
4	216	9	9
8	216	10	9
16	216	15	9

Table 1. Poisson problem—results for the constant relative overlap $\delta/H = 44.44\,\%$.

No. of subdomains	Mesh size $(1/h)$	No. of iterations without CGP	No. of iterations with CGP
2	72	4	6
4	72	7	9
8	72	8	10
16	72	11	10
2	144	4 (4)	6 (6)
4	144	7 (7)	10 (9)
8	144	8 (8)	10 (10)
16	144	10 (10)	9 (9)
2	216	4	7
4	216	7	10
8	216	8	9
16	216	10	9

Table 2. Burgers problem—results for the constant relative overlap $\delta/H = 44.44\%$.

The results are presented in Table 1 for the Poisson problem and in Table 2 for the Burgers problem. The numbers in brackets concern the divergence free discretization of the Burgers system.

Table 1 clearly indicates the correctness of our software. The behaviour of the parallel additive Schwarz solver without the coarse grid is as it has been expected— there is no dependence of the number of iterations on the mesh size h and there is a linear dependence on the number of subdomains ($\sim 1/H$). When the coarse grid has been used, the dependence on the number of subdomains disappeared, which is again in accordance with the theory.

From Table 2 we can see that the Burgers problem shows a similar behaviour as the Poisson problem. However, there are some differences here. In the case without the coarse grid the dependence on $1/H$ is weaker than it was for the Poisson problem. Another difference is that the version with the coarse grid needs more iterations than the version without one, when less then 16 subdomains is used, while in the Poisson problem case this holds only for two subdomains. This behaviour could indicate that the restriction and prolongation operators used are not the very best for our problem. However, if we used a smaller relative overlap, the version with the coarse grid would be better even for a small number of subdomains, as is shown in Section 6.2.

6.2. Dependence on Relative Overlap

6.2.1. Description. This test shows the dependence of the convergence on changes of the relative overlap δ/H. It was run using the following parameters: The mesh size was $1/h = 168$. There were 441 fine nodes per one coarse node. The add-mul preconditioner was used. The test solution at the initial time was used as the initial condition for the Burgers problem.

6.2.2. Results. The results for both the Poisson and Burgers problems are shown in Table 3.

Relative overlap	Number of iterations			
	Poisson problem		Burgers problem	
	without CGP	with CGP	without CGP	with CGP
4.76 %	30	13	21	14
14.29 %	15	8	10	9
28.57 %	11	9	8	9
47.62 %	9	9	7	9
71.43 %	8	9	7	9

Table 3. Poisson and Burgers problems — dependence on the relative overlap.

We have obtained a classical result for the Poisson problem without the coarse grid (see e.g. [5])—the convergence improves rapidly when the relative overlap increases for small overlaps, while for bigger overlaps the improvement slows down. Another fact to point out is the influence of the coarse grid. When it was used, we obtained a good convergence even for very small overlaps. The behaviour of the Burgers problem was very similar to that of the Poisson problem. Since our test problems were not very large, it seemed that the bigger the overlap the better the results even in terms of the elapsed time. However, for bigger problems, the elapsed time increases again after crossing a certain border. We did not pursue this effect because of inexact subdomain solutions and a bad load balance on a generally used parallel computer.

6.3. Various Two-level Preconditionings

6.3.1. Description. Here we compare the convergence properties of three two-level Schwarz preconditioners, namely the pure additive (called add-add), and the hybrid add-mul and mul-add preconditioners (these two are distinguished only by the ordering) (see e.g. [5]). It was expected that the hybrid approach would give better results than the pure additive. Another task of this test was to reveal a possible dependence of the convergence on the coarseness of the coarse grid. The parameters were as follows: The mesh size was $1/h = 210$ for 4 subdomains and $1/h = 168$ for 16 subdomains. The relative overlap was $\delta/H = 44.44\%$ and $\delta/H = 42.80\%$ respectively.

6.3.2. Results. The results are presented in Table 4 for the Poisson problem and in Tables 5, 6 for the Burgers problem. The numbers in brackets are the convergence rates. We have included them to see better the differences among the methods. They were computed as

$$\text{convergence rate} \; = \; \sqrt[k]{\|r^k\|/\|r^0\|},$$

where k is the number of iterations.

No. of fine nodes	Number of iterations (convergence rate)		
per coarse node	add-add	mul-add	add-mul
9	11 (0.1611)	9 (0.1003)	9 (0.1206)
49	11 (0.1401)	8 (0.0936)	8 (0.0832)
225	10 (0.1548)	9 (0.1276)	9 (0.1105)
441	12 (0.1872)	10 (0.1135)	9 (0.1083)

Table 4. Poisson problem—results for various two-level preconditionings, 4 subdomains.

No. of fine nodes	Number of iterations (convergence rate)		
per coarse node	add-add	mul-add	add-mul
9	10 (0.1496)	9 (0.0963)	9 (0.1267)
49	11 (0.1555)	8 (0.0922)	9 (0.0934)
225	11 (0.1739)	9 (0.1233)	9 (0.1156)
441	11 (0.1629)	9 (0.0955)	9 (0.0988)

Table 5. Burgers problem—results for various two-level preconditionings, 4 subdomains.

No. of fine nodes	Number of iterations (convergence rate)		
per coarse node	add-add	mul-add	add-mul
9	12 (0.1811)	9 (0.1280)	9 (0.1203)
49	10 (0.1537)	8 (0.0916)	9 (0.1210)
441	9 (0.1256)	9 (0.1119)	9 (0.1138)

Table 6. Burgers problem—results for various two-level preconditionings,
16 subdomains.

We can clearly see that there are almost no differences between the two problems in this case. The hybrid add-mul and mul-add methods are equivalent and, as was expected, they perform better than the pure additive one. A bit surprising is the independence on the coarseness of the coarse grid, even for the Burgers problem.

6.4. Coefficient Jump Influence

6.4.1. Description. Here we modified the coefficient functions ν, ν_x and ν_y in the following manner:

$$\nu_* = \begin{cases} 0.01 & \text{for } x < 0.5 \text{ and } y < 0.5, \\ 1 & \text{otherwise,} \end{cases}$$

where $\nu_* \in \{\nu, \nu_x, \nu_y\}$. The other parameters were as follows: The mesh size was $1/h = 168$ and the relative overlap was $\delta/H = 42.80\%$. There were 441 fine nodes per one coarse node. The add-mul preconditioner was used. The test solution in the initial time was used as the initial condition for the Burgers problem.

6.4.2. Results. The results are presented in Table 7 for both the Poisson problem and the Burgers problem.

Problem	No. of subdomains	No. of iterations without CGP	No. of iterations with CGP
Poisson	4	9	9
Poisson	16	14	9
Burgers	4	7	10
Burgers	16	12	10

Table 7. Results for the solution with coefficient jumps.

This test was done only to see whether the algorithm converges in the non-homogeneous case. We can see that it may have a similar behaviour as in the homogeneous case for the jumps towards zero. However, for the jump from 1 to 100 the algorithm did not converge.

7. Conclusion

We have compared numerically the known convergence properties of two-level domain decomposition methods for the linear elliptic problems with those obtained by experiments for the non-linear parabolic Burgers system.

We can say in general that the multi-level domain decomposition methods can be used even for the Burgers problem, though they are designed primarily for linear elliptic problems. We have shown several interesting properties of our simple two-level Schwarz preconditioners. We have demonstrated the independence of their convergence on the number of subdomains and moreover, even for the Burgers problem, the independence of their convergence on the coarseness of the coarse grid. The latter fact was really surprising—we do not know its reason and we suspect that it might disappear for great numbers of subdomains (≈ 1000), complicated domain geometries or non-smooth coefficients.

Then we have shown the similar behaviour of both our problems when increasing the relative overlap of subdomains. The convergence improves rapidly when increasing small overlaps, while the improvement is not very fast for increasing already big overlaps.

The hybrid two-level Schwarz methods (add-mul or mul-add) converged better than the pure additive method.

A deeper research must be done concerning the influence of the size and smoothness of the coefficients ν_x, ν_y, since we have put them equal to 1 in the majority of our tests. These coefficients are in a close relation to the Reynolds number that characterizes the properties of a fluid described by the Navier-Stokes system. Therefore the results may be directly useful in the computational fluid dynamics.

References

[1] *P. Angot*: Parallel Multi-Level and Domain Decomposition Methods. Calculateurs Parallèles L.T.C.P. *6(4)* (1994).

[2] *S. Kortas and P. Angot*: A practical and portable model of programming for iterative solvers on distributed memory machines. Parallel Computing *22* (1996), 478–512.

[3] *P. Le Tallec*: Domain Decomposition Methods in Computational Mechanics. Comput. Mech. Adv. (1994).

[4] *P. J. Roache*: Computational Fluid Dynamics. Hermosa Publishers, 1972.

[5] *B. Smith, P. Bjærstad, and W. Gropp*: Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations. Cambridge Univ. Press, 1996.

[6] *H. A. Van der Vorst*: Bi-CGSTAB: A fast and smoothly converging variant of BiCG for the solution of non-symmetric linear systems. SIAM J. Sci. Statist. Comput. *13* (1992), 631–644.

Author's address (home): *Robert Cimrman*, V Bezovce 2, 320 16 Plzeň, Czech Republic; work: Department of Mechanics, Faculty of Applied Sciences, University of West Bohemia, Plzeň, Czech Republic, e-mail: `cimrman3@students.zcu.cz`.