European Conference on Computational Fluid Dynamics
ECCOMAS CFD 2006
P. Wesseling, E. Oñate and J. Périaux (Eds)

A BLOCK STRUCTURED LOW MACH NUMBER MPV SCHEME FOR HETEROGENEOUS DOMAIN DECOMPOSITION

Sabine P. Roller*, Harald Klimach, Claus-Dieter Munz†

*High Performance Computing Center Stuttgart (HLRS)
Nobelstr. 19, D-71569 Stuttgart
e-mail: roller@hlrs.de, klimach@hlrs.de
web page: http://www.hlrs.de/people/roller
†Institute for Aerodynamics and Gasdynamics (IAG), University of Stuttgart
Pfaffenwaldring 21, D-71550 Stuttgart
e-mail: munz@iag.uni-stuttgart.de

Key words: Low Mach, Multi-Scales, Semi-Implicit schemes, Heterogeneous Domain Decomposition, Coupling

Abstract. *This paper is concerned with heterogeneous domain decomposition for semi-implicit Low Mach number schemes. The actual scheme is the Multiple Pressure Variables (MPV) scheme, which can be applied for a large range of Low Mach Number flows, ranging from natural convection flows with large temperature and density changes to aero-acoustics with recoupling of the acoustic to the flow. For aero-acoustic calculations, the numerical costs often are high due to the large computational domains. To reduce the computational effort, the domain is divided into non-overlapping sub-domains with non-matching cells. The future aim is to couple the semi-implicit code to domains with explicit time marching schemes. Therefore methods are considered, that do not require information from the neighbouring domains during one time step. This leads to a block-structured implementation spanning the solution space for the CG–solver over every single block instead of the hole computational domain. Information is exchanged at the beginning or end of the time step, not during the iteration steps.*

1 INTRODUCTION

Low Mach number flows are typical multi-scale applications where phenomena on different length scales interact with each other. In the recent years, different approaches for the calculation of Low Mach Number fluid flow have been developed. Among them is the Multiple Pressure Variables (MPV) scheme, which accounts for the different roles played by "the pressure" by using different variables and different numerical determinations of the thermodynamic, acoustic and hydrodynamic pressure parts. Based on an asymptotic

multi-scale analysis, the equations and the interactions of these pressure parts are determined. Thereby, the different scales can be accounted for in a very efficient way. The influence of the small scale to the large one is included as well as vice versa. The scheme can be applied for a large range of Low Mach Number flows, ranging from natural convection flows with large temperature and density changes to aero-acoustics with recoupling of the acoustic to the flow [1, 2, 3].

Nevertheless, especially aero-acoustic calculations are still expensive. On the one hand, the discretization has to be fine enough to resolve the small vortical structures. These are responsible for the generation of sound waves. The waves themselves have a much longer wave length, but travel over long distances within the same time interval. Therefore, the computational domain is usually very large. On the other hand, interaction of fluid flow and sound waves is essential only in a rather small domain around the sound sources. Away from this sound origin, resolution of small vortices is not necessary, therefore it is desired to use a much coarser grid. This allows for larger time steps also. Moreover, non-linear and viscous effects can be neglected in the so-called far field.

Utzmann et. al [4] therefore developed a heterogeneous domain decomposition scheme, where the equations, methods, grids and time steps are adapted to meet the local requirements. Heterogeneity in this context means that in every domain only the necessary effort is payed. Near the origin of the sound waves, a fine, maybe unstructured grid is applied on which the non-linear Euler- or Navier–Stokes equations are solved. This domain is the most expensive and therefore kept as small as possible. Attached to this domain is a coarser and structured grid where viscosity is neglected, but non-linear effects still play a role. In the (near) far field then, linearized equations might be used on a usually even coarser grid.

The current implementation of the coupling scheme uses explicit time marching schemes in all domains. The future aim is to couple the MPV-code to these domains. The numerical scheme implemented in the MPV-scheme is a semi-implicit SIMPLE-type scheme which requires the solution of a non-linear pressure Poisson equation. Different preconditioned Krylov subspace methods are implemented [5, 6], which are embedded in an outer iteration due to the necessary linearization of the pressure Poisson equation.

Parallelization of Krylov subspace methods is usually done by domain decomposition, where the Krylov spaces are spanned over the whole domain, but solved line by line on several processors. This requires communication in each iteration step. Moreover, all domains have to provide the same information, i.e. have to solve the same equations with the same numerical scheme.

For the coupling with explicit schemes, these information are not available, and the Krylov spaces cannot be spanned over the whole domain. Therefore, we investigate in this paper construction methods, which do not require information from the neighboring domains during the Krylov-subspace iteration. This leads to a block-structured implementation spanning the solution space for the CG-solver over every single block instead of the whole computational domain. Information is exchanged at the beginning or end of

the outer (the linearization) iteration, not during the iteration steps.

In this paper, we shortly review three different types of domain decomposition techniques for the desired scheme. Numerical results as well as convergence analysis will be presented. Special emphasize will be put on the coarsening factor between two neighboring grids, and peculiarities at the corners explained. The introduction of these techniques into the non-linear iteration process of the MPV-scheme will be outlined.

2 DOMAIN DECOMPOSITION FOR THE POISSON EQUATION

Our aim is a block-structured implementation of the MPV scheme where in each domain the Poisson equation is solved individually. Information from the neighboring domains is included as boundary condition only. The naive way of doing is to prescribe the neighboring values as Dirichlet boundary conditions. In that case, the domains have to overlap, otherwise the scheme does not converge [7]. Convergence is better, the larger the overlap is. Prescribing the values on the boundary or in one row of ghost cells can be interpreted as overlapping domains where the overlap is exactly this row of ghost cells. Thus, the overlap is very small and hence the convergence rate poor.

For our coupling purpose, we apply a non-overlapping domain decomposition. In this case, a second condition has to be fulfilled on the borders of the domains, that is the equality of the normal derivatives. Formulating the two conditions as functions of the pressure p (since this is the variable, the Poisson equation is solved for), the requirement on the artificial boundary $\Gamma = \Omega_i \cap \Omega_j$ is

$$\Phi(p_i) = \Phi(p_j) \quad \text{on} \quad \Gamma \tag{1}$$

$$\Psi(p_i) = -\Psi(p_j) \quad \text{on} \quad \Gamma. \tag{2}$$

where $\Phi(p) = p$, $\Psi(p) = \frac{\partial p}{\partial n}$, and n being the outward unit normal, i.e. $n_i = -n_j$.

Thus, the original Poisson problem

$$\begin{aligned} \Delta p &= f && \text{in} && \Omega \\ p &= p_{Dirichlet} && \text{on} && \partial\Omega_{Dirichlet} \\ \tfrac{\partial p}{\partial n} &= \left(\tfrac{\partial p}{\partial n}\right)_{Neumann} && \text{on} && \partial\Omega_{Neumann} \end{aligned} \tag{3}$$

is decomposed into multiple subproblems

$$\begin{aligned} \Delta p_i &= f_i && \text{in} && \Omega_i \\ p_i &= p_{Dirichlet} && \text{on} && \partial\Omega_i \cap \partial\Omega_{Dirichlet} \\ \tfrac{\partial p_i}{\partial n} &= \left(\tfrac{\partial p}{\partial n}\right)_{Neumann} && \text{on} && \partial\Omega_i \cap \partial\Omega_{Neumann} \end{aligned} \tag{4}$$

with the additional interface conditions (1), (2).

The solution of the interface problem is done iteratively. The solution methods can be summarized into three classes.

2.1 Dirichlet–Neumann scheme

The domains are red-black colored and separated into domains fulfilling the Dirichlet-condition (1) on the artificial boundary, and domains fulfilling the Neumann-condition (2) on the interface.

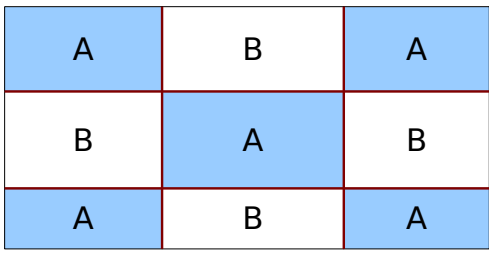

Figure 1: checkerboard coloring of the domains

First, the Dirichlet-domains have to be solved, afterwards the Neumann-domains can be solved. This is iterated up to convergence. With a negligible loss in the convergence rate, it is possible to perform both steps in parallel by starting the Neumann-domains from values of the Dirichlet-domain of the previous iteration level, and to correct afterwards.

A more serious problem is that if domains A are the Neumann-domains, the domain in the middle in figure 1 has Neumann-boundaries only. In this domain, the solution of the Poisson equation is not unique and iterative solvers for the linear equation system will not converge. For more than 9 domains, this situation is unavoidable.

2.2 Neumann–Neumann scheme

Here, no coloring is necessary and all domains are treated in parallel. In every domain, the Poisson equation is solved twice to fulfill both requirements: in a first step, Dirichlet conditions are used, requiring equality with the values of the neighboring domain in the previous iteration step. Afterwards, the Poisson problem is solved a second time with Neumann requirement of equal normal derivatives. Again, it is iterated up to convergence. The scheme is therefore twice as expensive as the Dirichlet–Neumann scheme.

The problem of purely Neumann boundary conditions and thus the non-uniqueness of the solution is present here for all domains with only inner boundaries, making this method unsuited for our purposes, too.

2.3 Agoshkov–Lebedev schemes

The third possibility is the method of Agoshkov–Lebedev which prescribes mixed boundary conditions. The domains are again colorized in a checker-board manner and solved alternatingly. The boundary conditions on the interface are defined as a linear combination of Dirichlet- and Neumann conditions. Additionally, a relaxation of the solutions can be used to accelerate convergence.

The iteration procedure for the Agoshkov–Lebedev scheme thus looks like:

1. Solving domains of type A (Ω_A):

$$\begin{aligned} \Delta p_A^{k+1/2} &= f_A && \text{in} \quad \Omega_A \\ p_A^{k+1/2} &= p_{Dirichlet} && \text{on} \quad \partial\Omega_A \cap \partial\Omega_{Dirichlet} \\ \frac{\partial p_A^{k+1/2}}{\partial n} &= \left(\frac{\partial p}{\partial n}\right)_{Neumann} && \text{on} \quad \partial\Omega_A \cap \partial\Omega_{Neumann} \\ a \cdot p_A^{k+1/2} + \frac{\partial p_A^{k+1/2}}{\partial n_A} &= a \cdot p_B^k - \frac{\partial p_B^k}{\partial n_B} && \text{on} \quad \Gamma \end{aligned} \tag{5}$$

2. Relaxation in Ω_A:

$$p_A^{k+1} = p_A^k + \alpha \cdot \left(p_A^{k+1/2} - p_A^k\right) \tag{6}$$

3. Solving domains of type B (Ω_B):

$$\begin{aligned} \Delta p_B^{k+1/2} &= f_B && \text{in} \quad \Omega_B \\ p_B^{k+1/2} &= p_{Dirichlet} && \text{on} \quad \partial\Omega_B \cap \partial\Omega_{Dirichlet} \\ \frac{\partial p_B^{k+1/2}}{\partial n} &= \left(\frac{\partial p}{\partial n}\right)_{Neumann} && \text{on} \quad \partial\Omega_B \cap \partial\Omega_{Neumann} \\ p_B^{k+1/2} + b \cdot \frac{\partial p_B^{k+1/2}}{\partial n_B} &= p_A^{k+1} - b \cdot \frac{\partial p_A^{k+1}}{\partial n_A} && \text{on} \quad \Gamma \end{aligned} \tag{7}$$

4. Relaxation in Ω_B:

$$p_B^{k+1} = p_B^k + \beta \cdot \left(p_B^{k+1/2} - p_B^k\right) \tag{8}$$

Equations (1), (2) are replace by (5d), (7d). The parameters $a \geq 0$ and $b \geq 0$ as well as the relaxation parameters α and β can be chosen individually in every iteration. We consider here only constant parameters for all iterations. The different sign in the Neumann-contributions to the boundary conditions is due to the opposite orientation of the normal vectors n_A and n_B. By setting a, b to zero, the Dirichlet–Neumann scheme is regained. Thus, the Dirichlet–Neumann scheme forms a subclass of the Agoshkov–Lebedev schemes.

In every iteration, all sub-domains have to be solved once. Since the domains Ω_B need the solutions from Ω_A, the sub-domains Ω_B have to be solved afterwards. By using p_A^k instead of p_A^{k+1} in the boundary conditions for domains of type B, this constraint can be omitted for the benefit of parallelization and at the (negligible) cost of convergence.

2.4 Prescribing boundary conditions using ghost cells

The interface conditions are prescribed on the (artificial) boundary of the subdomains. These boundary values are not used directly in the numerical scheme. The current implementation uses ghost cells which are determined in such a way that the interface conditions hold. That's the way it is done for the outer boundaries of the complete domain.

For the artificial boundaries, there are two possibilities of prescribing the values: The first one is to interpolate the values from the inner cells of domain Ω_A to the interface $\Gamma = \Omega_A \cap \Omega_B$ and then to interpolate them to the ghost cells of domain Ω_B. It is also possible to interpolate the values in the center of the ghost cells of Domain Ω_B directly in the neighboring domain Ω_A, without explicitly defining the values on the interface itself. If the two domains use the same discretization, i.e. the cells in both domains have equal size, then the ghost cells of domain Ω_B are identical to the first or last inner cells of domain Ω_A. Therefore, the second possibility is to prescribe directly the inner values of domain Ω_A to the ghost cells of domain Ω_B and vice versa. This is still possible if the cells are not of the same size. In that case some interpolation is necessary to get the value at the center of the ghost cell from the inner cells in the neighbor domain. The effect of these two possibilities will be investigated in the numerical examples.

3 NUMERICAL INVESTIGATIONS AND RESULTS

The testcase for the following analysis is the two-dimensional boundary value problem on the unit domain $[0,1] \times [0,1]$:

$$\begin{aligned} \Delta p(x,y) &= 32 \cdot ((y-1)y + (x-1)x) \\ p(0,y) = p(1,y) &= p(x,0) = p(x,1) = 0 \end{aligned} \tag{9}$$

The exact solution is

$$p(x,y) = (1-x)x \cdot (1-y)y. \tag{10}$$

For this testcase, convergence of the Dirichlet–Neumann, Neumann–Neumann, and Agoshkov–Lebedev scheme is investigated. Since the goal of the implementation is the coupling of the MPV code within a heterogeneous domain decomposition, the dependence of the convergence rate on the mesh ratio between two neighboring domains is especially considered. The results are summerized in a rather condensed manner only.

3.1 Convergence of the Dirichlet–Neumann scheme

The convergence of the Dirichlet–Neumann (D–N) scheme depends on several criteria: the way, the boundary conditions are set (interpolation to the boundary or injection to the ghost cells directly), on the location of the domain intersection (at the symmetry plane of the solution or asymmetrically), and on the mesh ratio.

Figure 2 shows the convergence of the D–N scheme for a cutting at the symmetry plane $x = 0.5$ (left), and asymmetrically at $x = 0.25$ (right). For the symmetric cutting, the D–N scheme converges exactly in two steps. After the second step, the discretization accuracy is reached. This 2-step convergence is lost for the asymmetric cut. At least, the convergence rate is independent of the grid size.

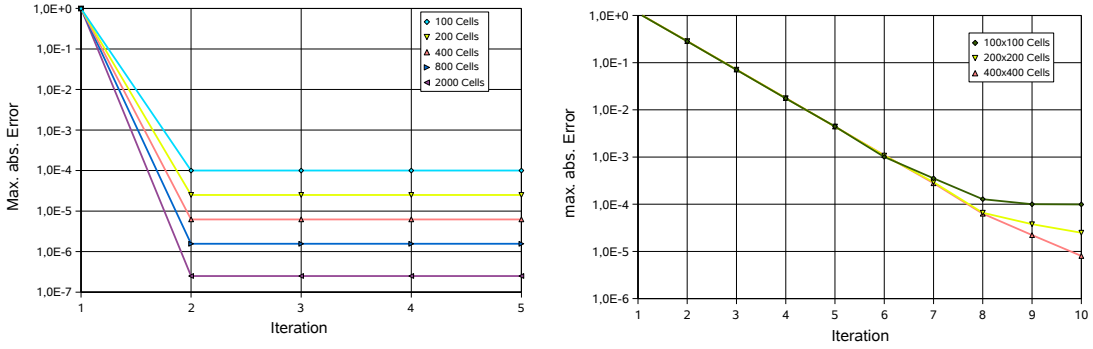

Figure 2: Convergence of the D–N scheme for a cutting at the symmetry plane $x = 0.5$ (left) and for an asymmetric cutting at $x = 0.25$

If the domains are discretized with unlike coarseness of the grid, the solution depends on whether the coarser domain is the Dirichlet- or the Neumann-domain. If the Dirichlet-domain is discretized finer than the domain with Neumann boundary condition, the discretization error of the coarser domain is carried fully into the domain with the higher resolution. The reason therefore is that the order of accuracy of the boundary condition doesn't match the discretization order. For a second order scheme, the reachable accuracy on the finer domain is reduced for any mesh ratio higher than 2. When the Dirichlet-domain is discretized finer than the Neumann-domain, the error is not only larger. The maximum of the error also lies in the higher resolved domain instead of the coarser one.

If the boundary values are set by direct injection on the ghost cells instead of interpolation to the interface, the scheme uses overlapping domains. Even for this very small overlap of only one row, the characteristics of the scheme change. On the one hand, the algorithm is accelerated, on the other an influence of the cell width is observed now. Still, the Neumann-domain should be the one with the higher resolution, but this effect is weakened. In summary, it can be stated that the definition of the boundary values in the ghost cells instead of the interface makes the scheme less depending on the special choice of the subdomains.

3.2 Convergence of the Neumann–Neumann scheme

The behavior of the Neumann–Neumann scheme is very similar to that of the Dirichlet–Neumann scheme. The only positive effect which can be gained from choosing the Neumann–Neumann scheme instead of the Dirichlet–Neumann scheme is a greater stability at crosspoints, as all Domains exchange Dirichlet boundary values.

As in the Dirichlet–Neumann scheme the Neumann–Neumann scheme fails, if there is a to big difference in the cell size of the interfacing grids. Since with this scheme all domains are treated equally it doesn't matter which of the domains is coarse.

The Neumann–Neumann boundary values can not be defined properly in ghost cells, as there has to be a shared value on the interface, which is used by both domains. So to use this scheme boundary values on the interface have to be used.

The Neumann–Neumann scheme converges in many cases at the same rate, as the Dirichlet-Neumann scheme. However in each step there have to be two boundary value problems to be solved. So this scheme needs twice the calculation time and is quite expensive without any real gain for our purpose.

3.3 Convergence of the Agoshkov–Lebedev scheme

The Agoshkov–Lebedev schemes form a generalization of the Dirichlet–Neumann scheme. The use of mixed boundary conditions on the interfaces overcomes the limits of the Dirichlet–Neumann as well as the Neumann–Neumann schemes, since no domains with purely Neumann boundary conditions can appear.

The scheme uses four parameters, thus representing a wide range of domain decomposition methods. The parameters a and b are weighting factors for the Neumann- and Dirichlet contribution to the boundary conditions respectively. The α and β are relaxation parameters. Numerical experiments have shown the best convergence for methods without relaxation, i.e. $\alpha = \beta = 1$. We therefore concentrate here on the analysis of the two linear factors defining the boundary conditions.

Choice of the weighting factors for Dirichlet- and Neumann-contribution. Particular points in the parameter space are those fulfilling the relation $b = \frac{1}{a}$. For such a combination, the ratio of Dirichlet- and Neumann-contribution to the boundary condition is the same in all domains. Especially for $a = b = 1$, all domains are treated equally. In 3 and 4, the convergance rate is plotted in dependence on a and b for a decomposition in two (left) and four (right) equal domains. For two domains, the map shows a clear maximum for the choice $a = 2$, and $b = 0.5$. I.e., in both domains the Dirichlet-contribution to the boundary condition is weighted twice as much as the Neumann-contribution. Here, a convergence in two steps is obtained, independent from the grid resolution. For four domains, the maximum convergence rate is at about $a = 3.8$, $b = 0.25$. The optimal parameters here do not coincide with the equal weight of the Dirichlet- to the Neumann-contribution on the boundary conditions. Nevertheless, the point $a = 4$, $b = 0.25$ is close to the maximum.

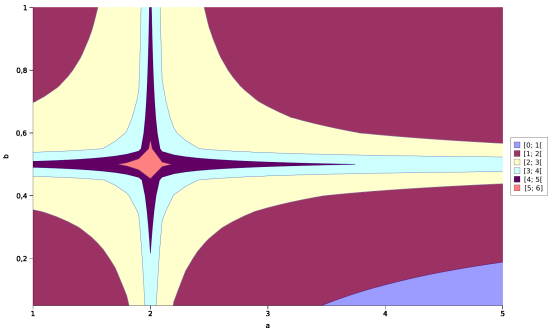

Figure 3: Convergence rate of the Agoshkov–Lebedev scheme in dependence on the parameters a and b for a symmetrically cutted boundary value problem

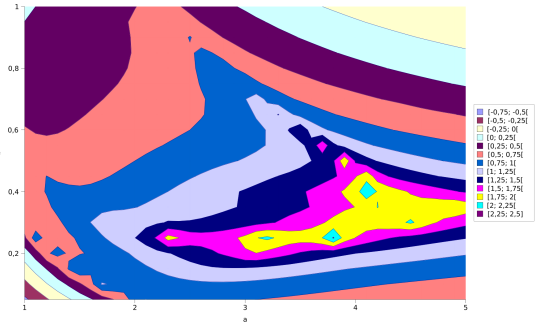

Figure 4: Convergence rate of the Agoshkov–Lebedev scheme in dependence on the parameters a and b for the decomposition into four subdomains of a 1D boundary value problem

Influence of the cutting position. Similar to the Dirichlet–Neumann scheme, the maximum convergence rate is shifted if the domain is decomposed asymmetrically. Other than for the D–N scheme, the scheme remains convergent. Detailed investigations have shown, that for parameter pair $a = 4$, $b = 0.25$, the convergence rate shows the lowest dependency on the positions of the decomposition.

Effects of the ghost cell value definition. The choice of the boundary values on the interface shows the same effects as described for the Dirichlet–Neumann scheme. For coarse grid resolution, the convergence is accelerated when the ghost values are prescribed directly by injection of the inner cells of the neighbouring domain. With higher resolution of the grid, the convergence rate decreases and tends asymptotically to the convergence rate obtained when using the interpolated values on the interface.

Unlike discretized domains The Dirichlet–Neumann scheme has shown an influence of the ratio of mesh resolution in the subdomains on the solution and the solvability of the problem. A survey over the behavior of the algorithm at different discretizations of the domain is given in table 1.

	max. Error in domain	
Cells in fine domain	fine	coarse (15×30)
150×300	$6.18 \cdot 10^{-4}$	$1.06 \cdot 10^{-3}$
75×150	$5.69 \cdot 10^{-4}$	$1.06 \cdot 10^{-3}$
60×120	$5.46 \cdot 10^{-4}$	$1.06 \cdot 10^{-3}$
45×90	$4.48 \cdot 10^{-4}$	$1.06 \cdot 10^{-3}$
30×60	$3.70 \cdot 10^{-4}$	$1.06 \cdot 10^{-3}$

Table 1: Reached accuracy of the Agoshkov–Lebedev scheme at unlike discretization of two domains

The maximum error in the finer discretized domain is increasing for higher resolutions

due to the smaller cells and therefore smaller distance from the incorrect values of the coarse domain. The limit is the discretization error of the coarser domain on the interface. Contrairy to the Dirichlet–Neumann or Neumann–Neumann scheme, the maximum error remains always in the coarser domain, even for high mesh ratios. The error distribution for a discretization ratio of 10 is depicted in fig. 5.

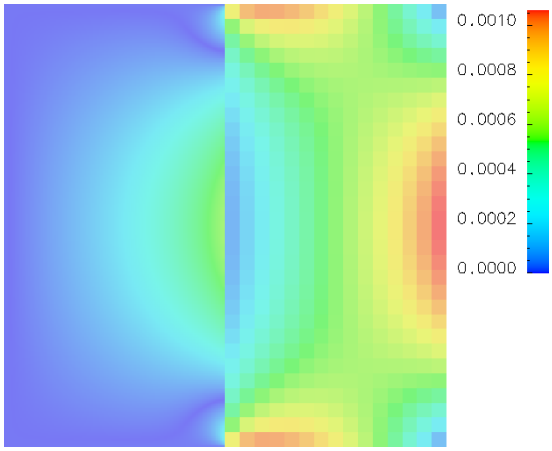

Figure 5: Error distribution for a discretization ratio of 10

By defining the Dirichlet-conditions on the cell interfaces, the influence of the discretization error in the coarse domain on the finer discretized domain is reduced. Since for the parameter family $b = \frac{1}{a}$, the same weighting of Dirichlet- and Neumann-contributions is chosen in all subdomains, it is unimportant whether domains of type A or of type B are discretized finer or coarser.

This advantagous behavior is due to the mixed type of boundary conditions. In none of the sub-problems, pure Dirichlet boundary conditions are used. The Neumann-part of the boundary condition thus leads to a reduction of the error induced by the unequal discretization.

Behavior at crosspoints The decomposition of a two-dimensional domain in a way that a crosspoint of intersections occurs, led for the Dirichlet–Neumann scheme to instabilities at the crosspoint for unequally discretized domains. This problem is not observed here, since all domains are connected to each other by a Dirichlet-contribution. However, for the parameter pair $a = 4$, $b = 0.25$, the maximum error occurs in the finer domain. The error distribution for a decomposition into 4 domains is depicted in fig. 6.

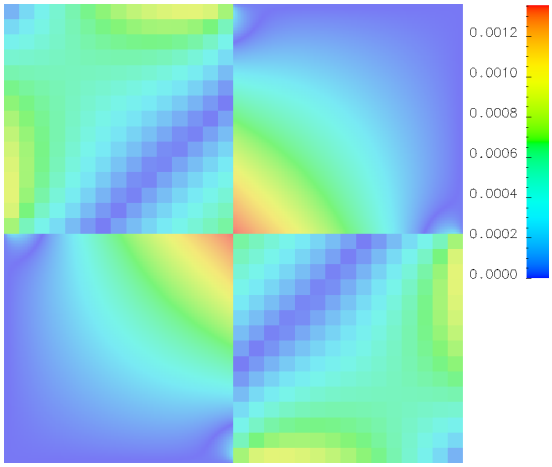

Figure 6: Error distribution on four domains (two with 150×150, two with 15×15) for parameters $a = 4$, $b = 0.25$

The algorithm remains stable, but the maximum error in the finer domain is not desirable. For this decomposition, the parameter pair $a = b = 1$, treating all domains equally as well as weighting Dirichlet- and Neumann-part in the boundary conditions equally, is advantageous. Fig. 7 shows the corresponding error distribution. This parameter choice leads to the required error distribution with higher accurary on the finer resolved subdomains.

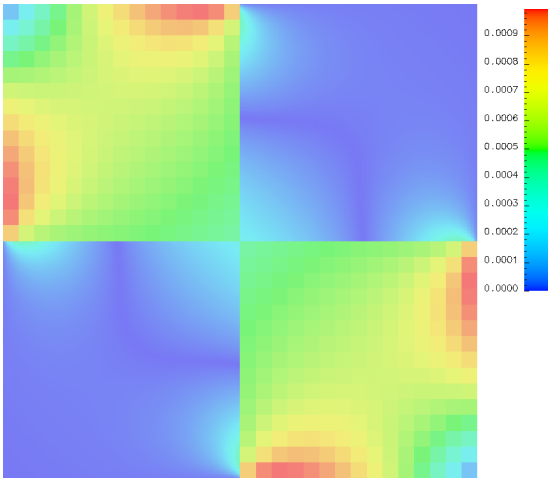

Figure 7: Error distribution on four domains (two with 150×150, tow with 15×15) for parameters $a = b = 1$

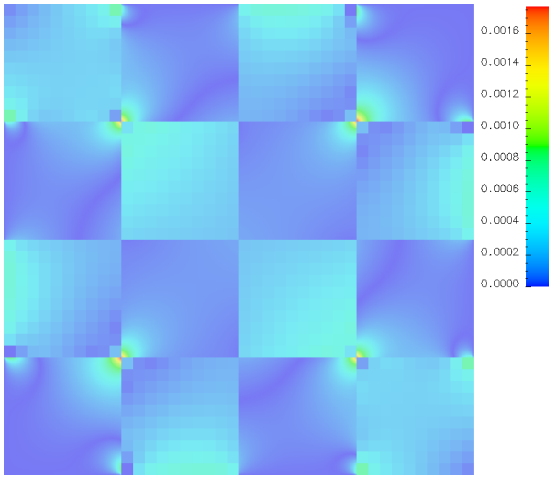

Figure 8: Error distribution on 16 domains (eight with 150×150 and the other eight with 15×15) for parameters $a = b = 1$

Negative for this parameter choice $a = b = 1$ is the observed slower convergence. A possible overcome could be to calculate the first steps with the faster parameter pair

$a = 4$, $b = 0.25$, and then to switch to the $a = b = 1$ pair to obtain the desired error distribution.

Is a subdomain completely surrounded by other subdomains, the error on the crossing points can no longer be avoided by the parameter pair $a = b = 1$. The error observed here in the corners of the finer domains due to the unequal discretization is shown in fig. 8. The obtained solution, however, remains stable.

4 CONCLUSIONS

With the aim of coupling domains with different meshes and different numerical schemes, mainly non-overlapping domain decomposition methods as the Agoshkov and Lebedev algorithm were used and analysed.

The coupling accross domain corners is only weak, which can lead to instability of the decomposition algorithm in the Dirichlet–Neumann scheme. Another problem is the need of pure Neumann boundary conditions in both the Neumann-Neumann and the Dirichlet–Neumann algorithm. That prohibits us from using domains, completely surrounded by others.

To be able to use different solvers and discretizations in the neighboring domains, we want to decouple the solution in the different domains as far as possible. A main objective to achieve this goal was to achieve a decomposition algorithm independent from the grids in the coupled domains. The analysis of the decomposition methods with different grids in neighbouring domains showed, that the Neumann–Neumann algorithm and the Dirichlet–Neumann algorithm both get instable if the cell size in the neighbouring domains differ to much.

This leaves us with an algorithm in the scheme of the Agoshkov and Lebedev decomposition method with mixed boundary conditions. As we avoid pure Neumann boundary conditions in this case, it is no longer a problem, if a domain is completely surrounded by other domains, i.e. has no physical boundary. Further the problems because of weak couplings at the domain corners still exist in this algorithm but they don't lead to instability anymore. This holds true even for big differences of the cellsizes in neighbouring domains.

As the Agoshkov and Lebedev algorithm has 4 parameters to adjust the ratio of the boundary conditions and relaxations, there are a lot variations on this scheme. Analysis of the parameter space shows that it is desirable to weight the Dirichlet part of the boundary condition in relation to the Neumann part equally across all domains. Which means it should be the same in both types of domains, created by the checker-board coloring. Relaxation should not be used.

There are two possibilities to define the boundary conditions at the domain borders. They can be set in the ghost cells or on the cellboundaries themselves. These variants of boundary definition has only a minor influence on the decomposition. Mostly it is preferable to set the boundary values in the ghostcells, as in this case there has not to be taken special care of the boundary elements in the solver itself. The ghostcells are of

the same size as the inner cells and their values are computed by direct injection or (if necessary) interpolation in the neighbouring domain.

The analysis of the decompositon methods so long applies to the pure Poisson equation. Within the MPV scheme the decompositon is integrated into the iterations over the time and over the linearization of the equation. Ideally, the iteration over the domain decomposition is applied within each linearization step on to the linearized equation system. This would be quite expensive and it has to be determinied how the decomposition iterations can be intermixed with the linearization steps to save computing time. Additionally the influence of the other variables on the the domain decomposition of the pressure term is to be analyzed. There are interactions of the different iteration levels expected, which will propably stabilize the domain decomposition algorithm.

Further effort will be put on the weakest point of the Agoshkov and Lebedev decomposition, which is the weak coupling of domains interfacing only at a corner. A possible solution might be to determine the pressure in the ghostcell on the corner by interpolating it in the diagonal neighbour instead of planar extrapolation.

REFERENCES

[1] Sabine Roller. *Ein numerisches Verfahren zur Simulation schwach kompressibler Strömungen.* PhD thesis, University of Stuttgart, 2004.

[2] Claus-Dieter Munz, Sabine Roller, Rupert Klein, and Karl J. Geratz. The Extension of Incompressible Flow Solvers to the Weakly Compressible Regime. *Computers and Fluids*, 32(2):173–196, September 2003.

[3] S. Roller, Th. Schwartzkopff, R. Fortenbach, M. Dumbser, and C.D. Munz. Calculation of low mach number acoustics: a comparison of MPV, EIF and linearized euler equations. *Mathematical Modelling and Numerical Analysis (M2AN)*, 39(3):561–576, May-June 2005.

[4] J. Utzmann, T. Schwartzkopff, M. Dumbser, and C.-D. Munz. Heterogeneous domain decomposition for CAA. *AIAA Journal*, to appear in 2006.

[5] M. Ratzel, S. Roller, and C.-D. Munz. A semi-implicite pressure corretion scheme for low Mach number flows. In Koichi Oshima, editor, *Proc. of the ISCFD99 in Bremen*, Comput. Fluid Dynamics Journal, Special Number, 2001.

[6] Marc Ratzel. *Die Simulation von schwach kompressiblen Strömungen auf körperangepaßten, strukturierten Gittern.* PhD thesis, University of Stuttgart, 2004.

[7] Barry Smith, Petter Bjorstad, and William Gropp. *Domain Decomposition.* Cambridge University Press, 1996.