SUSANNE KILIAN   MATTHIAS MÜNCH

A new generalized domain decomposition strategy for the efficient parallel solution of the FDS-pressure equation

Part I: Theory, Concept and Implementation

A new generalized domain decomposition strategy for the efficient parallel solution of the FDS-pressure equation
Part I: Theory, Concept and Implementation

Susanne Kilian[a], Matthias Münch[b]

[a]*hhpberlin, Ingenieure für Brandschutz GmbH*
Rotherstr. 19, 10245 Berlin

[b]*Interessengruppe Numerische Risikoanalyse (INURI)*
c/o Freie Universität Berlin, FB Mathematik und Informatik, Arnimallee 6, 14195 Berlin

Abstract

Due to steadily increasing problem sizes and accuracy requirements as well as storage restrictions on single-processor systems, the efficient numerical simulation of realistic fire scenarios can only be obtained on modern high-performance computers based on multi-processor architectures. The transition to those systems requires the elaborate parallelization of the underlying numerical concepts which must guarantee the same result as a potentially corresponding serial execution and preserve the convergence order of the original serial method. Because of its low degree of inherent parallelizm, especially the efficient parallelization of the elliptic pressure equation is still a big challenge in many simulation programs for fire-induced flows such as the Fire Dynamics Simulator (FDS). In order to avoid losses of accuracy or numerical instabilities, the parallelization process must definitely take into account the strong global character of the physical pressure. The current parallel FDS solver is based on a relatively coarse-grained parallellization concept which can't guarantee these requirements in all cases. Therefore, an alternative parallel pressure solver, SCARC, is proposed which ensures a high degree of global coupling and a good computational performance at the same time. Part I explains the theory, concept and implementation of this new strategy, whereas Part II describes a series of validation and verification tests to proof its correctness.

Key words: CFD, Zero-Mach number scheme, domain decomposition, Fire Dynamics Simulator (FDS), pressure equation, FFT, SCARC, fire safety

Email addresses: `S.Kilian@hhpberlin.de` (Susanne Kilian), `Matthias.Muench@inuri.de` (Matthias Münch)

1. Introduction

The Fire Dynamics Simulator (FDS) provided by the National Institute of Standards and Technology [23] is a world wide used tool for numerical simulation issues of fire and smoke spreading. The FDS program is focused on fire-induced flows. It is used as a prediction tool especially for thermal stresses and smoke movement as well as the usefullness of fire-extinguishing systems, or smoke funnels in buildings for practical fire engineering and scientific investigations.

One important feature of the program package FDS is the possibility to decompose the computed domain geometrically into smaller subdomains or meshes. This technique is a prerequisite for parallel computing and a time efficient numerical computation of practical problems. But, the application of the parallel FDS-version may cause inaccuracies or instabilities, as demonstrated by different authors e.g. [12, 19, 26]. The figure below presents an example that shows the possibility of big differences in the computed temperature which are inacceptable for fire safety engineering issues.
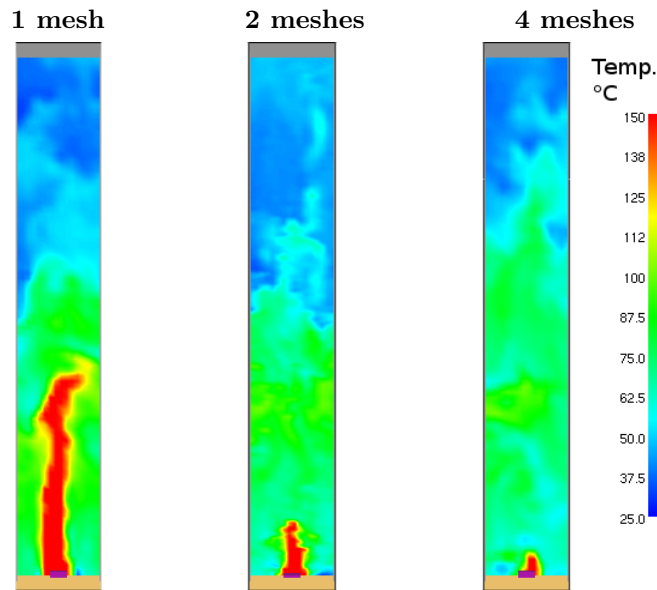

Figure 1: Comparision of the computed temperature of a burner in a shaft (FDS version 5.2). The shaft includes left: 1, middle: 2 and right: 4 equal (sub-)domains

To be precise: These errors result from deficiencies in the domain decomposition/-parallelization strategy in conjunction with the FFT-solver used to solve the pressure equation in FDS, not from the implemented physical model. Therefore computations without domain decomposition on single meshes are not affected by this problem. But for the most practical problems domain decomposition and parallelization are compelling features and unfortunately, FDS can not guarantee the necessary accuracy in this case.

When designing a parallelization strategy for FDS, it is extremely important to particularly account for one fundamental part, namely the pressure solver, which causes the major difficulties within a reliable parallelization process. The related pressure equation is an *elliptic partial differential equation of Poisson type* which has a very specific intrinsic character: It possesses an infinite rate of propagation for information, local information are spread extremely fast over the whole subdomain. In order to guarantee robust and accurate solutions, its parallelization MUST reproduce this global data flow as best as possible which requires sophisticated strategies for global data exchange.

The current parallelization concept of FDS is based on a mostly local philosophy which obviously isn't able to reflect this gobal dependency in all cases. But this is only a structural problem of the parallel solver and doesn't impair the quality of the serial FDS-version at all. In order to better extend the serial reliability into the parallel case, this article presents a new generalized domain decomposition strategy for the efficient parallel solution of the FDS-pressure equation which guarantees the necessary global coupling and accuracy in the parallel case.

The article consists of two parts: Part I explains the theory, the concept and the implementation of this new strategy. It illustrates the background theory as well as the including mathematical details. Part II [20] describes a series of validation and verification tests to proof the correctness of our new strategy. We demonstrate its numerical quality by illustrating its convergence properties and comparing the results with corresponding analytical solutions and the current FDS-FFT scheme.

Section 2 of this part describes the theoretical background of the asymptotically motivated Zero-Mach model used in the hydrodynamic solver of FDS. After a short introduction of the asymptotic theory for small Mach numbers we explain the consequences of the Zero-Mach number equations and derive the pressure equation used in FDS. In section 3 we discuss the lack of the current FFT-solver for the current parallel pressure equation in more detail. Section 4 presents the new numerical scheme (FDS-SCARC) and shows the enhancements compared to the current scheme (FDS-FFT). The definition of the corresponding boundary conditions is explained in more detail in section 5.

2. Theoretical background

In this section we present the theoretical background of the hydrodynamic solver in more general terms. Specifically, we employ a non-dimensional notation whereas the FDS program uses a dimensional formulation. Furthermore we omit all details which are not necessary to describe the basics of the theory. The fundamentals are the same and can be compared with [18]. Consider the dimensionless governing equations of mass, momentum, and energy for fire-induced

fluid flows

$$
\begin{aligned}
\frac{d}{dt} \int_V \rho \, dV & + \oint_{\partial V} \rho \mathbf{u} \cdot \mathbf{n} \, dA & &= \int_V S_\rho \, dV \\
\frac{d}{dt} \int_V \rho \mathbf{u} \, dV & + \oint_{\partial V} (\rho \mathbf{u} \circ \mathbf{u}) \cdot \mathbf{n} \, dA + \frac{1}{M^2} \oint_{\partial V} p \, \mathbf{n} \, dA & &= \int_V S_{\rho \mathbf{u}} \, dV \\
\frac{d}{dt} \int_V \rho e \, dV & + \oint_{\partial V} [\rho e + p] \, \mathbf{u} \cdot \mathbf{n} \, dA & &= \int_V S_{\rho e} \, dV \, .
\end{aligned}
\tag{1}
$$

To simplify the discussion all physical processes like radiation, combustion, heat conduction, etc. are subsumed in the source terms $S_\rho, S_{\rho \mathbf{u}}$, and $S_{\rho e}$. The integral conservation laws for mass, momentum, and energy then provide governing equations for the density ρ, velocity vector $\mathbf{u}$, pressure p, and energy density ρe as functions of time and space coordinates $(t, \mathbf{x})$ for arbitrary time independent control volumes V with boundary ∂V.

Assuming a perfect gas with a constant isentropic coefficient γ the energy density is defined in the equation of state

$$
\rho e = \frac{p}{\gamma - 1} + \frac{M^2}{2} \rho \mathbf{u} \cdot \mathbf{u} \, .
\tag{2}
$$

Non-dimensionalization changes neither the mathematical nor the physical content of the equations. However, one advantage of the dimensionless form is the occurrence of dimensionless reference numbers which weigh the various terms in the equations and explicitly characterize their relative order of magnitude. Here we consider the reference Mach number M, which is a scale for the importance of compressibility in a fluid flow computed by the relation between a reference flow velocity u_{ref} and a reference speed of sound c_{ref}.

$$
M = \frac{u_{ref}}{c_{ref}} \, .
\tag{3}
$$

Fire-induced flows can be in the range of 1 to 10 m/s or more. To make an order-of magnitude estimate, consider a cold inflow of $u_{ref} = 1 \, \text{m/s}$ with a speed of $c_{ref} \approx 330 \, \text{m/s}$. The resulting reference Mach number is small, $M \approx 0.003$, and this range is even valid for the hot smoke area.

As a consequence, the pre-factor of the pressure term in the momentum equation in (1) is very large. In the limit of a vanishing Mach number, a (mathematical) singularity arises, which signals the physical transition from a compressible to an incompressible flow. Such singularities are very hard to capture accurately in numerical computations. In the present example, the truncation errors associated with the approximation of the pressure gradient get amplified by $1/M^2$ and will destroy the accuracy of the velocity computation as M becomes very small. Numerous solutions to this problem have been proposed, but it is still a matter of active research.

Because for fire-induced flows we are not interested in resolving sound waves, one may completely filter out their influence by considering the asymptotic

limit equations that arise as $M \to 0$. The remaining set of equations describes the advection of entropy and vorticity and the influence of the various source terms on these quantities. A fundamental mathematical investigation for the limit $M \mapsto 0$ is published by *Klainerman et al.* [14] and *Schochet* [28]. The consequence is a pressure decompostion similar to the results found in *Rehm & Baum* [25] for thermally driven buoyant flows and the necessity for a system which does not resolve the propagation of sound waves. Based on these pressure decomposition idea *Klein* [15] introduces an asymptotic multiple scale ansatz

$$\mathbf{U}(\mathbf{x}, t; M) = \sum_i M^i \, \mathbf{U}^i(\mathbf{x}, M\mathbf{x}, t) \tag{4}$$

where $\mathbf{U}$ denotes the vector of the unknowns, which will be expanded with regard to the power of the Mach number. By inserting this asymptotic multiple scale ansatz into (1) *Klein* identifies three different physical relevant parts of the pressure:

$$p(\mathbf{x}, \xi, t) = \underbrace{p_0(\mathbf{x}, \xi, t)}_{thermodynamic} + \underbrace{M p_1(\mathbf{x}, \xi, t)}_{acoustic} + \underbrace{M^2 p_2(\mathbf{x}, \xi, t)}_{hydrodynamic\,pressure\,part}. \tag{5}$$

Here, $\mathbf{x}$ denotes the short hydrodynamic length scale and $\xi = M\mathbf{x}$ represents the acoustic scale. Furthermore, the results of the multiple scale asymptotic advise the differentiation of three flow regimes:

- $M = 1$: the full compressible regime with the total pressure p,

- $0 < M \ll 1$: the weak compressible 'Low-Mach' regime with the pressure decomposition $p = p_0 + M p_1 + M^2 p_2$,

- $M \mapsto 0$: the incompressible 'Zero-Mach' regime with the pressure decomposition $p = p_0 + M^2 p_2$.

The main difference between these regimes is the lack of an acoustic pressure part in the 'Zero-Mach' regime similar to the theory of *Rehm & Baum*. We use the definition of a 'Zero-Mach' number flow for the following discussion.

2.1. Zero-Mach number regime

One important result of the related asymptotic analysis is that the pressure may be decomposed into two contributions with very different physical meanings (details see [15, 16])

$$p(\mathbf{x}, t) = p_0(t) + M^2 p_2(\mathbf{x}, t). \tag{6}$$

Furthermore, the asymptotics provide a constraint for the pressure p_0

$$\oint_{\partial V} p_0 \, \mathbf{n} \, dA = 0 \tag{7}$$

which implies that this pressure is only a function of time and constant in space.

5

Physically the pressure p_0 represents the thermodynamic part of the pressure p, whereas the pressure p_2 represents the hydrodynamic part which is responsible for flow acceleration and for maintaining incompressibility.

Inserting the pressure ansatz (6) into the equations (1) and (2) leads to

$$
\begin{aligned}
\frac{d}{dt}\int_V \rho\, dV &+ \oint_{\partial V} \rho\mathbf{u}\cdot\mathbf{n}\, dA & &= \int_V S_\rho\, dV \\
\frac{d}{dt}\int_V \rho\mathbf{u}\, dV &+ \oint_{\partial V} (\rho\mathbf{u}\circ\mathbf{u})\cdot\mathbf{n}\, dA &+ \oint_{\partial V} p_2\,\mathbf{n}\, dA &= \int_V S_{\rho\mathbf{u}}\, dV \\
\frac{d}{dt}\int_V \rho e\, dV &+ \oint_{\partial V} [\rho e + p_0]\,\mathbf{u}\cdot\mathbf{n}\, dA & &= \int_V S_{\rho e}\, dV
\end{aligned}
\tag{8}
$$

and a corresponding equation of state

$$
\rho e = \frac{p_0}{\gamma - 1}.
\tag{9}
$$

Inserting (9) into the energy equation in (8) leads to a divergence constraint for the velocity field $\mathbf{u}$

$$
\oint_{\partial V} \mathbf{u}\cdot\mathbf{n}\, dA = \frac{\gamma - 1}{\gamma\, p_0}\int_V S_{\rho e}\, dV - \frac{|V|}{\gamma\, p_0}\frac{dp_0}{dt}.
\tag{10}
$$

Therefore using this pressure decomposition reduces the energy equation in (1) to a divergence constraint for the velocity field $\mathbf{u}$ for variable density flows with vanishing Mach numbers. What does this divergence constraint mean in practice and physics for numerical simulations of fire induced flows as described in *Münch et al.* [21]

The consequence of the pressure decompostion (6) is a system of equations (8, 9) representing the governing equations for Zero Mach number flows. Sound waves are not resolved and the pressure term in the momentum equation will be determined by the hydrodynamic pressure p_2. Furthermore, the momentum and energy equations are decoupled, because the advection term in the energy equation will be determined by the thermodynamic pressure p_0. As a consequence the velocity field must fullfill the divergence constraint (10). Although the documentation of FDS never uses non-dimensional equations, the algorithm of FDS is based on this asymptotic theory.

2.2. Hydrodynamic Model used in FDS

The Fire Dynamics Simulator is based on this asymptotic motivated model of the previous section. But the developers decided to use a non-dimensionless non-conservation formulation, therefore the influence of the dimensionless mach numbers is hidden. Because the energy equation is reduced to a divergence

constraint the system (8) can be written as a set of partial differential equations of mass and momentum

$$
\begin{aligned}
\frac{\partial \rho}{\partial t} &+ \nabla \cdot (\rho \mathbf{u}) &= \mathcal{S}_\rho \\
\rho \left(\frac{\partial \mathbf{u}}{\partial t} \right. &\left. + (\mathbf{u} \cdot \nabla)\mathbf{u} \right) + \nabla \tilde{p} &= \mathcal{S}_{\rho \mathbf{u}}.
\end{aligned}
\tag{11}
$$

Here $\tilde{p}$ is analogous to the dimensionless hydrodynamic pressure $M^2 p_2$. Including the vector identity with the vorticity ω

$$
(\mathbf{u} \cdot \nabla)\mathbf{u} = \nabla \frac{|\mathbf{u}|^2}{2} - \mathbf{u} \times \omega
$$

into the momentum equation and some transpositions lead to a new formulation of the momentum equation

$$
\frac{\nabla \tilde{p}}{\rho} = -\frac{\partial \mathbf{u}}{\partial t} - \frac{\nabla |\mathbf{u}|^2}{2} + \mathbf{u} \times \omega + \frac{\mathcal{S}_{\rho \mathbf{u}}}{\rho}.
\tag{12}
$$

The divergence of this momentum equation (12) results in a Poisson-type equation with variable coefficients to compute $\tilde{p}$ with iterative solvers

$$
\nabla \cdot \left(\frac{\nabla \tilde{p}}{\rho} \right) = -\frac{\partial (\nabla \cdot \mathbf{u})}{\partial t} + \nabla \cdot \left(\mathbf{u} \times \omega - \frac{\nabla |\mathbf{u}|^2}{2} + \frac{\mathcal{S}_{\rho \mathbf{u}}}{\rho} \right).
\tag{13}
$$

Because of computational performance reasons the developers of FDS decided to use a fast direct method that utilizes Fast Fourier Transforms (FFT). To be able to apply this method, they use the ansatz

$$
\frac{\nabla \tilde{p}}{\rho} = \frac{\nabla \tilde{p}}{\rho_\infty} + \left(\frac{1}{\rho} - \frac{1}{\rho_\infty} \right) \nabla \tilde{p}
\tag{14}
$$

in (13) and separate the baroclinic torque in a way, that constant coefficients are possible[1]. Furthermore instead of the hydrodynamic pressure $\tilde{p}$ a new variable $\mathcal{H}$ is used as unknown pressure term

$$
\nabla \cdot \left(\nabla \underbrace{\left[\frac{|\mathbf{u}|^2}{2} + \frac{\tilde{p}}{\rho_\infty} \right]}_{\mathcal{H}} \right) = -\frac{\partial (\nabla \cdot \mathbf{u})}{\partial t} - \nabla \cdot \mathbf{F},
\tag{15}
$$

$$
\text{with} \qquad \mathbf{F} = -\mathbf{u} \times \omega + \left(\frac{1}{\rho} - \frac{1}{\rho_\infty} \right) \nabla \tilde{p} - \frac{\mathcal{S}_{\rho \mathbf{u}}}{\rho}.
$$

The variable $\mathbf{F}$ combines all the convective and diffusive influences and forces of the momentum equation.

This pressure equation plays a decisive role within the numerical scheme of FDS. In a first predictor step FDS computes a rough approximation of the thermodynamic quantities for the next time step. Furthermore the velocity

---

[1]A first estimation about the errors neglecting the baroclinic torque for fire induced flows can be found in *Münch et al.* [22]

is estimated at the next timestep using the new pressure term $\mathcal{H}$. Based on this estimated velocity a corrector step 'corrects' the thermodynamic quantities and computes the corrected velocity using a recomputed pressure $\mathcal{H}$ (for details see the Technical FDS Guide [18]). The numerical scheme in FDS requires the solution of the Poisson equation for the computation of the pressure $\mathcal{H}$ twice within a time iteration. Because of the interaction with the calculation of the thermodynamic quantities, the solution of the pressure equation affects the computation of all quantities. Therefore the correct computation of (15)

$$\nabla^2 \mathcal{H} = -\frac{\partial(\nabla \cdot \mathbf{u})}{\partial t} - \nabla \cdot \mathbf{F} \tag{16}$$

is a central issue for the accuracy of the numerical scheme implemented in FDS.

3. Current Parallelization Concept for the Pressure Equation in FDS

From a mathematical point of view the pressure equation (15) is an elliptic partial differential equation of Poisson type. When designing solvers for the pressure equation, it is very important to take into account the following intrinsic character of elliptic problems: Local influences or perturbations change the solution in the whole computational domain. In other words, there is an *infinite rate of propagation for information* which makes up the greatest hurdle on the way to an efficient parallelization: The fast and robust solving of the equation of Poisson type is essentially based on the fact of how good the numerical method reproduces this global dependency.

3.1. Comparision of FDS4 and FDS5

The current solving strategy within FDS is based on the application of a highly optimized FFT-method (*Fast Fourier Transformation*), see e.g. *Schwarz* [29]. For single-mesh problems this methodology has proven to be very robust, reliable and extremely fast. For multi-mesh problems, a local FFT-method is performed on each single subdomain with a corresponding data exchange for the definition of local boundary conditions. But during the development of FDS4 it became apparent, that a purely local coupling of the local FFT-solutions may be insufficient to guarantee the global convergence for all cases. This can be illustrated by a simple FDS4-example for an angled 2D-domain in figure 2 which was subdivided into 3 and 5 subdomains, respectively.

Obviously the domain decomposition sustainably broke up the physical connectivity, which couldn't be compensated by the local data exchange. There are many other examples where the volume and mass flow couldn't consistently be computed along the interior boundaries.

Due to the mathematical theory, a purely local approach isn't sufficient to reproduce the global data dependencies for elliptic problems, see *Rannacher* [24]. Increasing the number of subdomains will worsen the convergence rate, possibly
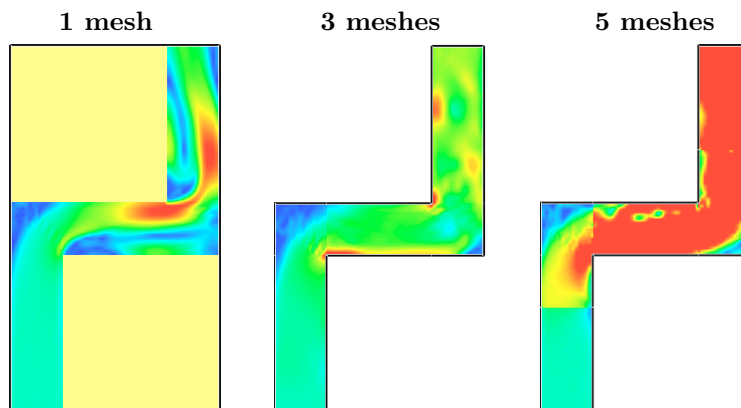
Figure 2: Example of a subdivision in 3 and 5 subdomains with FDS4

right up to divergence at a (problem-dependent) critical number of subdomains. To find a remedy, a domain-spanning correction process must take place which is able to spread global information all over the domain.

With the release of FDS5 a new *pressure-correction* process was introduced which is based on an averaging process on the coarse grid level (the grid of the subdivision itself) ensuring the consistency of the volume flow along interior boundaries. For a detailed derivation of the pressure correction see the Technical FDS Guide [18]. In the course of the single subreleases of FDS5, many other substantial improvements have been done. Except for very small variations, no visual differences can be detected in our pipe example any more, see figure 3.
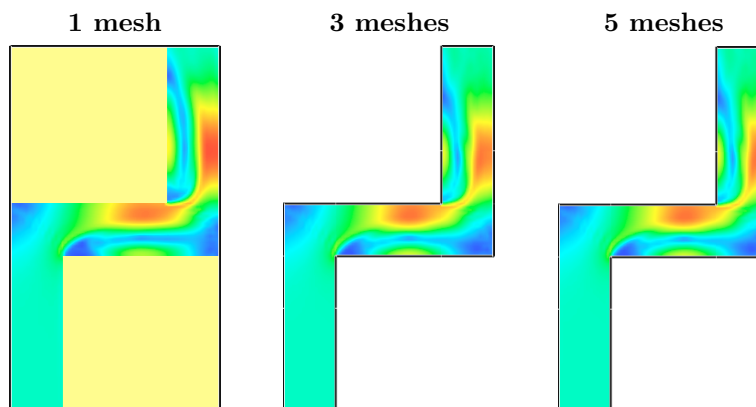
Figure 3: Example of a subdivision in 3 and 5 subdomains with FDS5

Nevertheless, remember figure 1 in the introductory section which was already computed with FDS5. Obviously, it demonstrates strong differences in the computed temperature. Therefore a purely visual comparison of the computational results is not an adequat method to proof the qualitiy of a numerical scheme. For a reliable evaluation more mathematical-numerical criteria are necessary.

Figures 2 and 3 illustrate another important argument for the need of a new, strong parallel solver: Because of the restriction to rectangular meshes in FDS, the simple flow-field geometry in the one-mesh case can only be realized by inserting obstacles (yellow areas in the left pictures) and taking out complete areas in the computational domain off. Although these areas don't belong to the real flow-field, they consume a considerable part of the storage space and computational time. To avoid 'dead ranges' like this, it's highly desirable to have the possibility to subdivide complex geometries into regular parts and solve the whole problem reliably with an efficient parallel solver.

3.2. Numerical quality criterias

As demonstrated above, numerical discretization schemes have a crucial influence on a CFD code's quality and decisively affect the stability and accuracy behaviour of the overall method.

From a physical point of view the solution of the underlying set of equations must be independent of the underlying domain decomposition. Simplified: the solution of a single- and multi-mesh-calculation should be the same. But what does that explicitly mean? Domain decomposition methods for the solution of boundary value problems always lead to more or less additional numerical errors and increase the inaccuracy of a numerical scheme. Nevertheless, the numerical error of a domain decomposition method or parallelization strategy must be limited by the numerical error defined by the order of the underlying numerical scheme. In case of FDS the scheme should be of second order accuracy in time and space (see *FDS Verification Guide* [17]).

The common methods to proof the accuracy of a numerical scheme are convergence tests. With convergence tests, the correctness of a numerical scheme can be probed empirically. As the grid size Δx vanishes, the truncation error should vanish as well at a rate determined by the order of the scheme,

$$\lim_{\Delta x \to 0} \left(\frac{\partial \varphi}{\partial x} - \frac{\Delta \varphi}{\Delta x} \right) = 0. \tag{17}$$

Convergence studies involving calculations of the same problem on grids with varying mesh sizes are necessary to check this basic aspect. Only a series of convergence tests on well-selected non-trivial test problems can establish with reasonable certainty that a code correctly implements the discretization schemes that it has been built upon. Unless a code has passed such tests, one cannot expect that it produces reliable results for realistic application problems.

It is beyond the scope of the present paper to provide more than an rough overview of the related investigations. In part II of this article we give an introduction to test strategies which are much more suited for a reliable evaluation than simple visual comparisions of numerical results. With these tests we demonstrate the quality of our new numerical scheme.

4. A New Parallelization Concepts for the Pressure Equation

The upper deficiencies suggest to develop completely new strategies for the solution of the pressure equation. Subsequently, a new parallelization concept, the generalized domain decomposition/multigrid method SCARC, is presented which is no longer based on a direct methodology as the current FFT-solver but on an iterative one. For a clearer understanding the differences between direct and iterative methods are shortly illustrated in the beginning. After explaining the most important core component of SCARC, the so called *basic iteration*, and some of its most important representatives, its algorithmic description is given. For more information about the underlying concepts see *Kilian, Turek* [11, 13]

4.1. Discretization Strategy

In FDS the computational domain is divided into rectangular boxes with rectangular equidistant grid cells inside. The spatial derivatives of the governing quantities are approximated by second-order accurate finite difference methods. Scalar quantities such as the pressure are assigned in the center of each grid cell. The corresponding discretized pressure equation looks like

$$
\begin{aligned}
\frac{\mathcal{H}_{i+1,jk} - 2\mathcal{H}_{ijk} + \mathcal{H}_{i-1,jk}}{\delta x^2} & \\
+ \frac{\mathcal{H}_{i,j+1,k} - 2\mathcal{H}_{ijk} + \mathcal{H}_{i,j-1,k}}{\delta y^2} & \\
+ \frac{\mathcal{H}_{ij,k+1} - 2\mathcal{H}_{ijk} + \mathcal{H}_{ij,k-1}}{\delta z^2} \quad = \quad & -\frac{F_{x,ijk} - F_{x,i-1,jk}}{\delta x} \\
& -\frac{F_{y,ijk} - F_{y,i,j-1,k}}{\delta y} \\
& -\frac{F_{z,ijk} - F_{z,ij,k-1}}{\delta z} - \frac{\partial}{\partial t}(\nabla \cdot \mathbf{u})_{ijk}
\end{aligned} \tag{18}
$$

where different discretizations for the time derivative of the divergence are used in the predictor and corrector step, see the Technical FDS Guide [18]. The definition of the corresponding boundary conditions is explained in more detail in section 5.

For a single-mesh computation with n grid cells in total this leads to the following system of equations

$$
Ax = b
$$

where A is a matrix in $\mathbf{R}^{n \times n}$ and x, b are vectors in $\mathbf{R}^n$. The multi-mesh case is discussed below. Due to its bad conditioning, this system of equations must be solved with very robust and efficient solvers.

Direct Methods:

At a first glance, the ongoing improvements in the current computer technology motivate the use of direct methods such as the Gaussian elimination method (and its variants for symmetric, positive definite matrices) for the solution of the resulting systems of equations. The decision for the local FFT-methods in FDS follows this trend: FFT-methods are known to be highly efficient direct methods which are successfully used in many different branches of science.

Direct methods compute the solution of a system of equations within one single (possibly very complex) computational cycle without any approximations in-between. They may be performed with enormous speed and are often used for the demonstration of potential computer power, see the LINPACK-tests by *Dongorra et al.* [6, 7]. They distinguish themselves to be very robust even in the non-symmetric and ill-conditioned case and are nearly independent of the degree of grid distortion.

In contrast to iterative methods, they don't need a good initial solution, but they don't take advantage of the fact if such an initial guess is already available. They achieve high computational accuracies, but they don't take advantage of the fact if only a moderate accuracy is needed.

Because of their highly recursive character direct methods are not very suited for an efficient parallel solution of sparse systems of equations as they occur in FDS. Usually the underlying parallelization strategies follow algebraic considerations, see *Frommer* [8], and are most often not conformal with geometrically motivated domain decompositions as they are used in FDS.

Iterative Methods:

Iterative methods produce a sequence of iterates during multiple computational cycles (with lower computational complexity) approximating the exact solution more and more. Most often they are easier to implement than direct ones, because they can be reduced to a series of core components such as matrix-vector multiplications, linear-combinations of vectors, scalar-products and inversion of tridiagonal or lower triangular systems, which may be suitably optimized.

However, iterative methods may depend on special properties of the underlying problem such as symmetry or positive-definitness and may convergence very slow for ill-conditioned problems. They often require the optimal choice of different method parameters such as relaxation parameters which can be very difficult.

The convergence rate of iterative methods usually depends on the grid resolution, but can be considerably improved by a suitable *preconditioning* which will be of great importance subsequently: The use of a *preconditioning matrix* $B \in \mathbf{R}^{n \times n}$ transforms the original system $Ax = b$ into an equivalent system $B^{-1}Ax = B^{-1}b$ which may be solved much faster.

The *preconditioning matrix B* has to fulfill two contradictory conditions: On the one hand, B^{-1} should be a good approximative inverse of A, i.e. $B^{-1}A \sim I$, or in other words, $B \sim A$. On the other hand, B should be easily applicable, i.e. $B \sim I$. The more special properties of the problem can be incorporated in B, the better the convergence is, but the higher the computational costs are. So, a careful compromise has to be found between those conflicting requirements.

Nevertheless, iterative methods have shown very satisfactory convergence results for a wide variety of applications which will be discussed in more detail below. Especially with regard to an efficient parallelization, iterative methods seem to be easier and more universally applicable than direct ones. Therefore they are the favorite candidates for our new pressure solving strategy.

4.3. The Basic Iteration

Following the above considerations the core of the new concept is the so called *basic iteration*

$$x^k = x^{k-1} - \omega B^{-1}(Ax^{k-1} - b) \tag{19}$$

Again, B is a matrix in $\mathbf{R}^{n \times n}$, x^k, x^{k-1} are vectors in $\mathbf{R}^n$ and ω is a *relaxation parameter* which must be chosen very carefully. This basic iteration represents a simple *defect-correction* scheme for the solution of $Ax = b$ with preconditioning matrix B and initial solution x^0.

Please keep in mind, that this basic iteration has to be performed in the predictor and corrector step of each encompassing time iteration in FDS, where the index 'k' doesn't belong to the time iteration, but only to the basic iteration. To be precise, the vector 'x^k' ought to be indexed with respect to the time as well, but to simplify the notation this is omitted.

The term $d^{k-1} := Ax^{k-1} - b$ is denoted as *defect* and serves as name giver for this class of methods. Measured in a suitable norm, it indicates how good the equation $Ax = b$ is fulfilled by the current iterate x^k. For the error $x - x^k$ in the k-th iteration step, there holds

$$x - x^k = (I - \omega B^{-1}A)^k(x - x^0)$$

with the *error propagation operator* $F = (I - \omega B^{-1}A)$. The sequence of iterations x^k converges to the solution of $Ax = b$ if and only if its spectral radius (the maximum of the absolute values of the eigenvalues) is smaller than 1. To put it briefly, the sense of a preconditioning consists in finding a better distribution of the eigenvalues for the transformed system $B^{-1}Ax = B^{-1}b$ to reach a faster convergence.

The general form (19) of the basic iteration is very well suited for an efficient implementation: As already mentioned it allows the splitting into matrix-vector multiplication, preconditioning and linear combinations where each can be separately performed with high performance tools if available. Besides, the explicit use of the complete defect $Ax^{k-1} - b$ is advantageous for certain techniques for implementing boundary conditions (see *Turek* [30]).

13

Typical candidates for the *preconditioner B* are:

- $B = \text{diag}(A)$ corresponds to Jacobi iteration

- $B = \text{lower part}(A)$ corresponds to Gauß–Seidel schemes

- $B = \text{tridiagonal}(A)$ corresponds to linewise variants of the above schemes

- $B = \tilde{L}\tilde{U}(A)$ corresponds to incomplete LU decomposition

A detailed overview on different preconditioners is given in *Hackbusch* [10] and *Deuflhard* [5].

4.4. Efficient Generalizations of the Basic Iteration

Especially in case of very complex geometries the convergence properties of the pure basic iteration are not very satisfactory. Therefore more efficient generalizations such as the global *conjugate gradient method* (CG) or a *multigrid method* (MG) may be used, which are probably the most efficient techniques for the solution of huge systems of equations arising from the discretization of partial differential equations.

Both classes are closely related, because they are based on simple defect correction iterations, which approximate the error by using a sequence of smaller subproblems. The main difference between both classes is found in the choice of the underlying subspaces. This issue is discussed in more detail in *Hackbusch* [10]. Subsequently a short comparison of both classes is given. For an algorithmic description of both methods see the subsection 5.1.1.

4.4.1. CG-Methods:

The CG-method is an effective descent method for symmetric, positiv-definite problems which only needs less storage space for several auxiliary vectors. In the course of one iteration a matrix-vector multiplication is needed which requires local data exchange in case of a parallel execution. Furthermore global scalar products are computed which contribute to a strong global coupling and a high stability. The parallel computation of these global scalar products is based on a global data exchange which only has a low parallel efficiency.

The convergence rate of CG-methods also depends on the discretization parameters, but can be considerably improved by corresponding preconditioning techniques, especially on the base of domain decomposition, see *Bramble, Pasciak, Xu* [4]. There also exist variants for the non-symmetric case, so called *bicg-methods* see e.g. *Saad* [27] or *Axelsson, Barker* [2].

4.4.2. MG-Methods:

Excellent convergence rates independent of the grid size with moderate computational complexity may be reached with MG-methods. The name 'multigrid' must not be confused with the term 'subdomain': It doesn't mean a sequence of different subgrids arising from a domain decomposition but rather a hierarchy of grids with different resolutions for one and the same domain/subdomain.

The basic idea behind MG-methods is to improve the convergence speed of the basic iteration by correcting the defects on successively coarser grids. The process explicitly exploits an important property of the single representatives of the basic iteration, the so called *smoothing property*. This special property is based on the fact that the new iteration value in one single grid point is computed as more or less simple mean value of the surrounding grid values. The complete procedure looks like this:

Starting from a given initial solution on the finest grid level, several steps of a simple basic iteration with a suitable matrix B are performed. After only a few iteration steps the mentioned smoothing property usually effects a considerable reduction of the high-frequent error components of the defect while the low-frequent components are nearly unchanged and still may be very large. This suggests to restrict this *smoothed* defect on the next coarser grid (e.g. with the double grid size) by using a suitable restriction Operator I_h^{2h}, where it can be approximated at much lower costs, see figure 4.

Restriction: fine $\longrightarrow$ coarse

$$I_h^{2h} = \frac{1}{4} \begin{bmatrix} 1 & & 1 \\ & \cdot & \\ 1 & & 1 \end{bmatrix}_h^{2h}$$
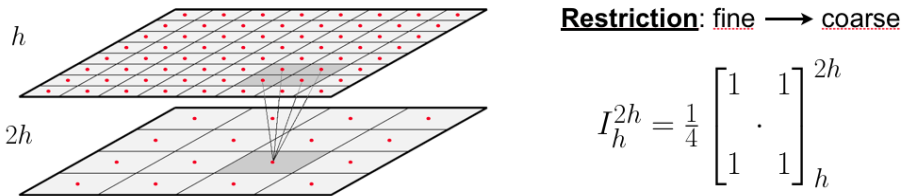
Figure 4: Restriction to the next coarser grid

This *presmoothing* process may be continued until the coarsest grid level has been reached where the remaining coarse grid problem is solved exactly. At this stage the low-frequent components are resolved by the maximum possible global coupling. Then, the resulting coarse grid solution is successively prolonged on the next finer grids whereby several steps of the basic iteration can be performed on each level for *postsmoothing*, depending on the type of the MG-method, see *Hackbusch* [10]

All in all, each grid level is responsible for the reduction of a special range of the error frequencies. The low-frequent components on a finer grid appear as high-frequent components on the next coarser grid. The efficiency of the complete method substantially depends on the fact of how good the ranges, which are smoothed on the single grid levels, are adjusted among each other.

There is not only ONE single MG-method but a big class of MG-methods consisting of very different components (smoothers, transfer operators between the single grids, coarse grid solvers) which can be adjusted to the underlying problem to the highest possible extend. The close-meshed connectivity of the different grid stages leads to an extremely strong coupling which will be very convenient for the solution of the pressure problem.

Unfortunately, not all ingredients of a MG-methods are equally suited for an efficient parallelization. The coarser the grid resolutions are, the worse the ratio of computational work to communication overhead gets. Besides, the inherent *recursive character* of many serial MG-methods (especially the underlying smoothing procedures) must be split off in order to achieve a better parallel efficiency. But this splitting is usually associated with a considerable loss of numerical efficiency (worse convergence behavior, dependencies on the number of subdomains or the refinement parameters), so a proper compromise has to be found. Especially, the exact solution of the coarse grid problem represents a serious bottle-neck: It is only a small problem with low computational complexity which additionally requires a global data exchange. Its computation implies a logarithmical growth of the communication overhead if the number of subdomains is increased, which however seems to be an unavoidable disadvantage, especially for increasing complexity of the underlying problem. Nevertheless, by using adequate domain decomposition strategies high numerical efficiencies may be reached, such that usually only a few MG-iterations must be performed.

4.5. Preconditioning by Domain Decomposition Techniques

A finite-difference discretization of the Poisson equation (16) in the multi-mesh case on the base of a subdivision into N subdomains leads to N local systems of equations
$$A_i x_i = b_i \qquad i = 1, \ldots N\,,$$
where each subgrid has n_i local grid cells. Here, $A_i \in \mathbf{R}^{n_i \times n_i}$ is the local system matrix on subdomain i and with corresponding local solution vector x_i and right hand side vector b_i in $\mathbf{R}^{n_i}$. Informally, A_i is the restriction of the global matrix A to the subdomain Ω_i, i.e. $A_i \sim \text{``}A\big|_{\Omega_i}\text{''}$.

For the choice of the preconditioning matrix B domain decomposition strategies can be used in a very natural way. The so called *Additive Schwarz preconditioner* looks like this
$$B_{as}^{-1} = \widetilde{\sum}_{i=1}^{N} A_i^{-1}\,. \tag{20}$$

The marked sigma sign indicates that a suitable averaging at internal boundaries has to take place. Obviously, the preconditioning is based on the local solutions of the single subdomain problems. In this way, many specific properties of the whole problem can be exploited. Depending on the underlying problem, it may be sufficient to solve the local problems only approximately up to a certain accuracy. The local solutions can be calculated by any efficient method, for example with local FFT-methods.

16

But this ansatz suffers from the same problems than the current FFT-strategy: Even for simple Poisson problems with moderate geometric irregularities this locally based approach is not satisfactory, especially for growing number of subdomains. According to the mathematical theory, its convergence rate depends of the number of subdomains. By the use of an additional coarse grid matrix $A_c \in \mathbf{R}^{N \times N}$, defined as Poisson matrix only on the coarse grid cells itself, this dependency may be considerably mellowed or even abolished. The corresponding *Additive Schwarz preconditioner with coarse grid correction* is defined as:

$$B_{asc}^{-1} = A_c^{-1} + \widetilde{\sum}_{i=1}^{N} A_i^{-1} \,. \tag{21}$$

Now, the complete preconditioning process not only uses the local subdomain solutions but also the solution of the coarse grid problem, which explicitly incorporates the global transfer of data.

Based on these requirements, the *generalized domain decomposition/multigrid approach* ScARC is defined as symbiosis of efficient global and local iterative techniques. In its simplest form, ScARC consists of the nested combination of an outer (global) defect-correction with N inner (local) defect-corrections.

Basic ScARC preconditioner:

- Solve the **global** problem
$$Ax = b$$
  by an overall defect-correction with additive Schwarz preconditioning $B^{-1} := B_{as}^{-1}$ or $B^{-1} := B_{asc}^{-1}$ corresponding to (20) or (21)

$$x^k = x^{k-1} - \omega B^{-1}(Ax^{k-1} - b) \,.$$

- In each step k of the global defect-correction solve N **local** problems for the restricted defects $d_i^{k-1} := \text{``}(Ax^{k-1} - b)\big|_{\Omega_i}\text{''}$

$$A_i y_i = d_i^{k-1} \,, \qquad i = 1, \ldots, N \,,$$

  with direct methods or with local defect-corrections based on suitable preconditioners C_i for A_i

$$y_i^m = y_i^{m-1} - \omega_i C_i^{-1}(A_i y_i^{m-1} - d_i^{k-1}) \,, \qquad i = 1, \ldots, N \,.$$

The global defect-correction method corresponds to a block Jacobi scheme on subdomain level. Its preconditioning consists in the solution of local subdomain problems which involve a high computational complexity and can be done in a processor-optimized way. A detailed description of the algorithmic concept and all related topics can be found in *Kilian, Turek* [11, 13].

The notation SCARC stands for:

- **Scalable**, w.r.t. the number of global ('k') and local solution steps ('m'),

- **Recursive**, since it may be applied to more than 2 global/local levels,

- **Clustering**, since adaptive blocking of subdomains is possible.

As already mentioned, the convergence properties of the pure basic iteration are not satisfactory for complex situations. Further optimizations can be reached, if the simple defect-corrections are accelerated by corresponding CG- or MG-methods. So, in the standard version of SCARC the outer iteration is replaced by a data-parallel global MG-method which is mainly based on the blockwise smoothing with local Schwarz problems. The local iterations on the m subdomains may also be replaced by local MG-methods with specially optimized local smoothers. Equally, it is possible to use a data-parallel global CG-method for the outer defect-correction and/or local CG-methods for the inner ones. Even direct local solvers with local FFT-methods are possible. By a sophisticated combination of global and local MG- and CG-strategies the advantages of both classes are combined to the greatest possible extend and a high numerical efficiency is reached, see *Kilian* [11], *Becker* [3].

The combination of a data-parallel global MG-method with optimized local MG-methods of Schwarz type corresponds to the execution of a complete Schwarz domain decomposition method on each level of the subgrids. In spite of the usual MG-features with low degree of parallelizm (e.g. the coarse grid problem), this double MG-structure effects a very strong global coupling such that the resulting high numerical efficiency dominates the losses in parallel efficiency. Especially in case of irregular geometries, the Schwarz smoothing has proven to be very robust.

All in all, SCARC represents a large class of methods which contains a wide spectrum of the known multigrid and domain decomposition approaches for the solution of discretized PDE's. The most important representatives are:

- set $k = 1$ (globally), solve exactly (locally)
  $\longrightarrow$ **Parallel CG-method with Additive Schwarz preconditioning**

- set $m = 1$ (locally) and $C_i = part(A_i)$
  $\longrightarrow$ **Standard multigrid with blockwise smoothing**

- set $m > 1$ (locally) and $C_i = part(A_i)$ and solve approximately (locally) via MG, CG, FFT
  $\longrightarrow$ **full** SCARC

Figure 5 illustrates the CG-variant and the full MG-variant for a 2×2-subdivision of a square domain. Here, the red arrows indicate a local data exchange between neighboring subdomains whereas the blue arrows indicate a global global data exchange. For the MG-variant the staggered sub-squares symbolize the use of a subgrid hierarchy.
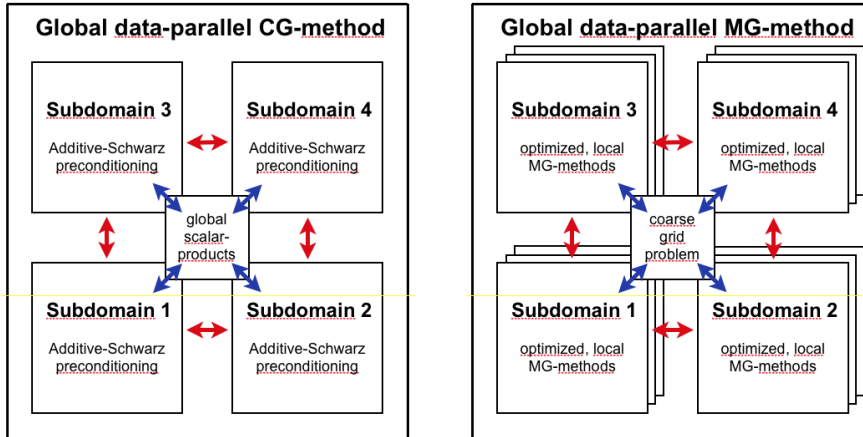
18

Figure 5: CG- and full MG-variant of SCARC

Another fundamental philosophy of this approach is the preservation of a maximum of *data locality* by the best possible discovery of locally structured blocks. Local irregularities should be hidden within the single subdomains and compensated by the optimized local smoothing. In this context, the good choice of suitable smoothing procedures for the local MG-methods plays a very prominent role. Experience has shown, that linewise Jacobi and Gauss-Seidel methods, especially ADI-TRIGS-methods, do a very good job. The application of local FFT-methods instead of the local MG-methods will be analyzed carefully in the medium term.

This strategy involves a high degree of arithmetical work which can be performed purely locally. The subgrids are mainly based on local *tensor product meshes* (with linewise numbering) which optimally fits to the rectangular grids of FDS. Additionally the single grid cells can be shifted towards local irregularities which allows very fine local grid resolutions and constitutes a form of fine grid adaptivity, see 4.6.2. These linewise grid structures enable the local application of *optimized libraries* in combination with *highly regular data structures*, which permit an optimized exploitation of local processor features like e.g. the cache, see *Turek et. al.* [1, 9]. Altogether, high computational efficiency as well as very fine local grid resolutions can be achieved.

In contrast to the current FFT-strategy this concept rests upon a global discretization where the grid points are grouped within the single subdomains. The corresponding matrix A of the global basic iteration is only formally defined and will never be assembled on the whole, but is distributed over the single processors. The complete iteration is globally defined and uses the solution of the subdomain problems only for the approximate solution of the global defect. All matrix-vector multiplications produce the same result as they would do if one performed the whole computation on one huge serial processor which is called *data-parallel execution*. During a global matrix-vector multiplication only local data exchange between neighboring macros are neccessary.

19

A very important advantage of this strategy is the fact that there is no need to impose artificial conditions at interior subdomain boundaries. Nodes along interior boundaries are inner nodes related to the virtual global matrix A and are treated as normal inner nodes. The computation ends with a global solution which is completely consistent along interior boundaries (especially with respect to the volume flow) without any further averaging processes. This stands in contrast to the more loose coupling of the local solution in the FFT-method.

4.6. Special Features

4.6.1. **Adaptivity on Coarse Grid Level**

A very important initial step is the careful choice of a suitable coarse grid which is already adjusted to the special characteristics of the underlying problem. In this context, two important requirements should be fulfilled: The coarse grid should consist of as much as possible 'orthogonal' subdomains. This fact decisively improves the convergence properties of our favourized local smoothing procedure (above all in situations with strong local anisotropies), namely the linewise Gauss-Seidel method. Additionally, due to the block-Jacobi character of the global smoothing procedure (blockwise composition of local solvers), the differences in size from one subdomain to its neighbors should be of moderate size. Instead of big jumps the use of some more subdomains should be preferred.

4.6.2. **Adaptivity on Fine Grid Level**

As mentioned before one important strategy of our SCARC-concept is based on *generalized tensor product meshes* which should be used on the single subdomains as far as possible. This allows for the application of highly regular data structures and an optimized exploitation of the local processor power. In order to achieve extremely fine grid resolutions these special meshes can simply be shifted in the direction of an anisotropic detail without loosing the logical shape of a tensor product mesh. A detailed formal definition of this shifting procedure is given in *Kilian* [11].

4.6.3. **Blind Nodes**

In order to resolve complex details adaptively and to minimize the number of global unknowns at the same time, finer grid resolutions on some special subdomains with geometric irregularity is possible. This leads to blind nodes along inner boundaries which have to be treated carefully, especially in the context of the local communication of inner boundary nodes.

Only one level of difference in the local resolution modes is allowed between two neighboring subdomains, see figure 6. If a very fine mesh width is needed, the refinement has to be staggered over a series of neighboring subdomains successively.
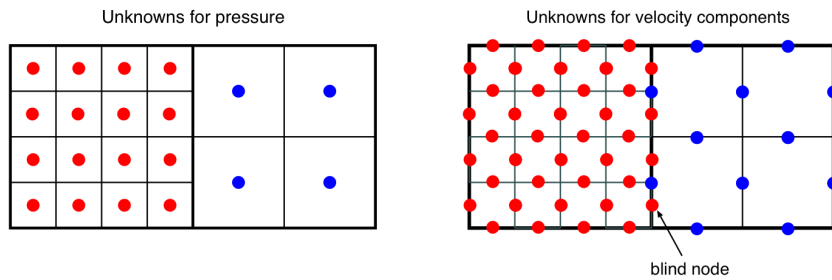
Figure 6: Different stages of refinement for the pressure and the velocity

4.6.4. **Introduction of a Master Solver**

Usually, it is more efficient to solve the small coarse grid problem not in parallel. Each subdomain is only responsible for a few nodes, so that the computational complexity on a single process is extremly low. In contrast to that, it requires the frequent exchange of small data over the whole network when, e.g., solved with the Gaussian elimination. This process is far from from exploiting the computing power of modern processors.

Therefore, it may be better to solve the coarse grid problem only on one single process. This process may be an already existing subgrid process or an additional *master process*. This separate master process would have to gather the coarse grid data from the subgrid processes using a global data exchange, then to compute efficiently the exact solution of it and, at last, to spread the solution back to the subgrids.

Experience has shown that most often the use of a master process is computationally faster than a distributed solution. In addition to it, the application of sophisticated distribution strategies could allow to use this master process not only for the solution of the coarse grid problem, but also for some other computations which might be done independently from the single subgrid processes.

Naturally, the use of a master process always induces a loss of parallel efficiency. It may happen, that all subgrid processes have to wait until the master process has finished its global calculation. But in order to reach a strong global coupling this kind of efficiency loss seems to be unavoidable.

Fortunately, the biggest part of the computational work can be done locally on the single subgrid processes. The current FDS strategy for starting a parallel computation of a given problem is to request as many processors as there are subdomains. The use of an additional master process would require one processor more.

5. Computational Details

5.1. Definition of Boundary Conditions

For the proper definition of external boundaries, different types of boundary conditions are used, see the Technical FDS Guide [18].

- At vents or solid boundaries *no-flux* or *forced-flow* boundary conditions

$$\frac{\partial \mathcal{H}}{\partial n} = -F_n - \frac{\partial u_n}{\partial t}$$

  are applied with the normal component F_n of $\mathbf{F}$ and the prescribed rate of change $\partial u_n / \partial t$ in the normal component of velocity at a forced vent.

- At open external boundaries the setting

$$\begin{aligned} \mathcal{H} &= |\mathbf{u}|^2/2 \quad \text{outgoing}, \\ \mathcal{H} &= 0 \qquad \quad \text{incoming} \end{aligned}$$

  is used, depending on whether the flow is outgoing or incoming.

Let's have a closer look at the definition of the external boundary conditions. As an example we regard a simple unit square geometry in 2D with equidistant grid in x- and z-direction, $h := \partial x = \partial z$. In this case, the no-flux condition at the floor $z = 0$ is given by

$$\frac{\mathcal{H}_{i,1} - \mathcal{H}_{i,0}}{h} = B_{i,0}$$

or equivalently

$$\mathcal{H}_{i,0} = \mathcal{H}_{i,1} - h\, B_{i,0} \tag{22}$$

where $B_{i,0} := -F_{z,i,0}$. Following the FDS notation, only the indices i and k are used, whereas the index j (which is always 1 for 2D-cases) is omitted for the sake of simplicity.

The resulting matrix line corresponding to node (i,k) looks like

$$\frac{1}{h^2}\left(\mathcal{H}_{i,k-1} + \mathcal{H}_{i-1,k} - 4\mathcal{H}_{i,k} + \mathcal{H}_{i,k+1} + \mathcal{H}_{i+1,k}\right) = R_{i,k} \tag{23}$$

with some value $R_{i,k}$ for the right hand side corresponding to the definition in (19). Setting (22) into (23) for all ghost cells with $k = 0$, there holds

$$\frac{1}{h^2}\left(\mathcal{H}_{i-1,1} - 3\mathcal{H}_{i,1} + \mathcal{H}_{i+1,1} + \mathcal{H}_{i,2}\right) = R_{i,1} + \frac{1}{h}\, B_{i,0}$$

At an open boundary (say $i = nx$) $\mathcal{H}$ is defined as

$$\begin{aligned} \mathcal{H}_{nx+\frac{1}{2},k} &= (u_{nx,k}^2 + \bar{v}_{nx,k}^2 + \bar{w}_{nx,k}^2)/2 \quad &, \quad u_{nx,k} > 0\,, \\ \mathcal{H}_{nx+\frac{1}{2},k} &= 0 &, \quad u_{nx,k} < 0\,, \end{aligned}$$

with corresponding interpolations to the boundary locations (indicated by the overbars) and the ghost cell value defined by linear extrapolation

$$\mathcal{H}_{nx+1,k} = 2\mathcal{H}_{nx+\frac{1}{2},k} - \mathcal{H}_{nx,k} \, .$$

And again this is used in (23) for all cells with $i = nx + 1$ to get

$$\frac{1}{h^2}\left(\mathcal{H}_{nx,k-1} + \mathcal{H}_{nx-1,k} - 5\mathcal{H}_{nx,k} + \mathcal{H}_{nx,k+1}\right) = R_{nx,k} - \frac{2}{h^2}\,\mathcal{H}_{nx+\frac{1}{2},k} \, .$$

For the case that a Neumann-boundary meets a Dirichlet-boundary, e.g. for (nx,1), we get

$$\frac{1}{h^2}\left(\mathcal{H}_{nx-1,1} - 4\mathcal{H}_{nx,1} + \mathcal{H}_{nx,2}\right) = R_{nx,1} + \frac{1}{h}\,B_{nx,0} - 2\frac{1}{h^2}\,\mathcal{H}_{nx+\frac{1}{2},1} \, .$$

The resulting matrix A is irreducibly diagonally dominant which guarantees the solvability by many iterative methods.

5.1.1. Communication Effort for the Global CG-Method and MG-Method

Subsequently a short overview of the communication effort for the global CG- and MG-method is given in terms of an algorithmic description in combination with a specification of the communication type. The notation $\boldsymbol{C_{local}}$ on the right hand side of a single instruction means that a local communication between neighboring subgrids takes place whereas $\boldsymbol{C_{global}}$ stands for a global communication concerning all subgrids. If no specification is used, the related instruction may be performed completely locally without any communication overhead.

The communication effort strongly depends on the choice of the global preconditioning matrix B. As far as there is no coarse grid problem involved as e.g. for a CG-method with additive Schwarz preconditioning without coarse grid correction, the parallel efficiency is rather high because the preconditioning only needs a local data exchange whereas a global exchange only takes place for the computation of the global scalar products. Using an additional coarse grid correction requires a global exchange of the coarse grid data in each preconditioning step as well.

The execution of the smoothing steps in the global multigrid only requires local data exchange because only the local additive Schwarz techniques are used whereas the solution of the coarse grid problem needs a global communication as described above.

Parallel CG-Iteration

Initialization:

$$r^0 = A\,x^0 - b \qquad \longleftarrow \quad \boldsymbol{C_{local}}$$

$$v^0 = \mathbf{B^{-1}\,r^0} \qquad \longleftarrow \quad \left\{ \begin{array}{l} B^{-1} = B_{as}^{-1} : \boldsymbol{C_{local}} \\ B^{-1} = B_{asc}^{-1} : \boldsymbol{C_{local}} + \boldsymbol{C_{global}} \end{array} \right.$$

$$d^0 = -v^0$$

$\boldsymbol{k \geq 0 :}$

$$\alpha_k = (r^k, v^k)/(d^k, A\,d^k) \qquad \longleftarrow \quad \left\{ \begin{array}{l} A\,d^k \qquad\;\; : \boldsymbol{C_{local}} \\ (d^k, A\,d^k) : \boldsymbol{C_{global}} \end{array} \right.$$

$$x^{k+1} = x^k + \alpha_k d^k$$

$$r^{k+1} = r^k + \alpha_k A\,d^k$$

$$v^{k+1} = \mathbf{B^{-1}\,r^{k+1}} \qquad \longleftarrow \quad \left\{ \begin{array}{l} B^{-1} = B_{as}^{-1} : \boldsymbol{C_{local}} \\ B^{-1} = B_{asc}^{-1} : \boldsymbol{C_{local}} + \boldsymbol{C_{global}} \end{array} \right.$$

$$\beta_k = (r^{k+1}, v^{k+1})/(r^k, v^k) \qquad \longleftarrow \quad \left\{ (r^{k+1}, v^{k+1}) : \boldsymbol{C_{global}} \right.$$

$$d^{k+1} = -v^{k+1} + \beta_k d^k$$
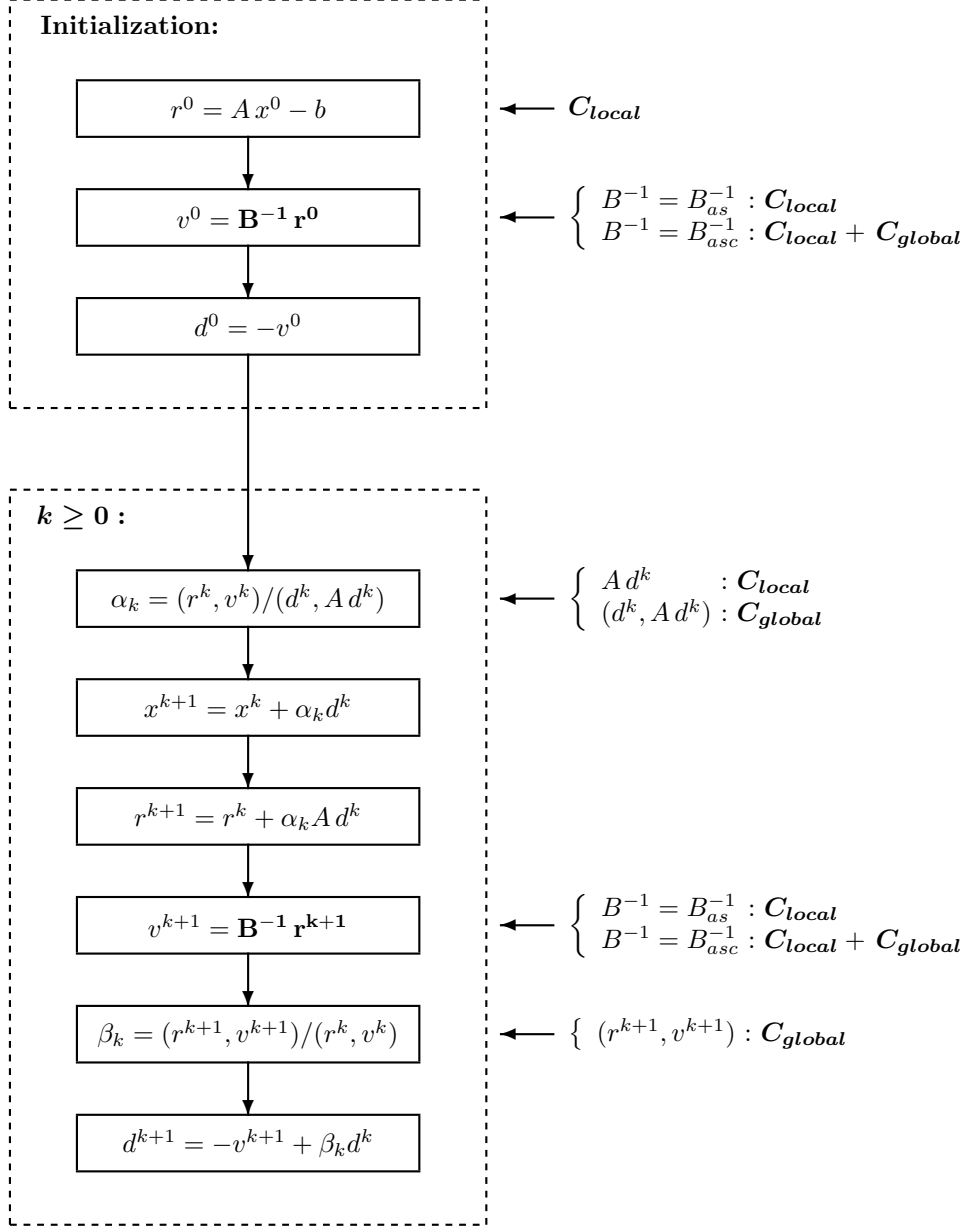
Let there be given a hierarchical sequence of L meshes for the domain Ω with corresponding mesh parameters $0 < h^{(L)} < \ldots < h^{(0)}$, starting from a coarse grid associated with the index '0'. On every mesh level l, $l = 0, \ldots, L$, the corresponding matrices, solution vectors and right hand side vectors are denoted with $A^{(l)}$, $x^{(l)}$ and $b^{(l)}$, resp. Further, let $R^{(l)^T}$ be the prolongation from level l to level $l+1$ and $R^{(l)}$ the corresponding restriction.

The parallelization is based on a non-overlapping domain decomposition in N subdomains Ω_i with $\overline{\Omega} = \cup_{i=1\ldots N} \overline{\Omega}_i$. . Corresponding to the single levels we define the local matrices $A_i^{(l)}$ and vectors $x_i^{(l)}$, etc., with the index 'i' denoting the affiliation to subdomain Ω_i. Let $R_i^{(l)}$ be a restriction operator which maps a global vector $x^{(l)}$ of level l on its local counterpart $x_i^{(l)}$ on subdomain $\Omega_i^{(l)}$, namely $R_i^{(l)} x^{(l)} = x_i^{(l)}$, and let $R_i^{(l)^T}$ be the appropriate prolongation operator. Further, let $A_i^{(l)}$ be the local portion of the matrix $A^{(l)}$ corresponding to subdomain $\Omega_i^{(l)}$ on level l. By construction there should hold $A_i^{(l)} = R_i^{(l)} A^{(l)} R_i^{(l)^T}$.

As already mentioned, the global matrices $A^{(l)}$ are never assembled on the whole, but only partially on the single submeshes. So, a global matrix-vector product on MG-level l is computed as

$$A^{(l)} x^{(l)} = \sum_{i=1}^{N} R_i^{(l)^T} A_i^{(l)} R_i^{(l)} x^{(l)} = \sum_{i=1}^{N} R_i^{(l)^T} A_i^{(l)} x_i^{(l)} \, .$$

The same holds true for the right hand sides and defects on the single levels.
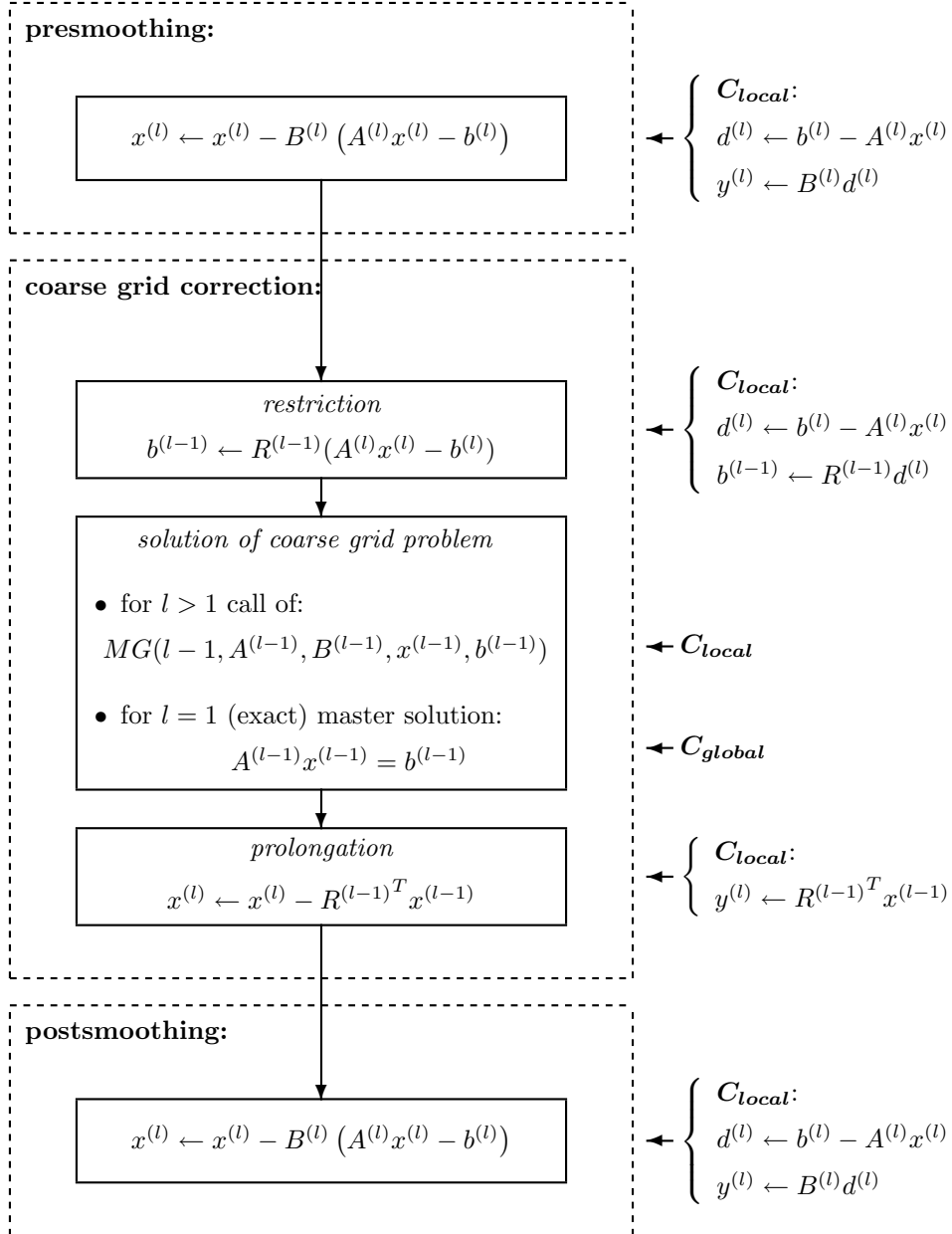
To simplify the notation, the global defect-correction index 'k' is omitted. Instead of the entire indexing in (19), the notation $x^{(l)} \leftarrow x^{(l)} - B^{(l)} \left(A^{(l)} x^{(l)} - b^{(l)} \right)$ is used to indicate that the left hand side is assigned to the value of the right hand side on level 'l'.

We restrict ourselves to the presentation of the most inner kernel which is associated with the smoothing procedure of a *global* MG-cycle. As indicated above, the global smoothing is based on the blockwise composition of optimized local solvers associated with the single subdomains. These may be local MG-methods which use quite different smoothing procedures specially adapted to the local (anisotropic) situations within the single subdomains.

The notation $MG(l, A^{(l)}, B^{(l)}, x^{(l)}, b^{(l)})$ denotes the application of a global multigrid cycle on level l using the corresponding global matrix $A^{(l)}$, the Schwarz preconditioner $B^{(l)} = B_{as}^{(l)}$, the iteration vector $x^{(l)}$ and the right hand side vector $b^{(l)}$ on MG-level l.

Now, one pre- or postsmoothing step on level l of the global MG-method can be described as follows:

Parallel MG-Iteration for level $l \geq 1$: $x^{(l)} \leftarrow MG(l, A^{(l)}, B^{(l)}, x^{(l)}, b^{(l)})$

presmoothing:

$$x^{(l)} \leftarrow x^{(l)} - B^{(l)} \left(A^{(l)} x^{(l)} - b^{(l)} \right)$$

$\leftarrow \begin{cases} \boldsymbol{C_{local}}: \\ d^{(l)} \leftarrow b^{(l)} - A^{(l)} x^{(l)} \\ y^{(l)} \leftarrow B^{(l)} d^{(l)} \end{cases}$

coarse grid correction:

restriction
$$b^{(l-1)} \leftarrow R^{(l-1)} (A^{(l)} x^{(l)} - b^{(l)})$$

$\leftarrow \begin{cases} \boldsymbol{C_{local}}: \\ d^{(l)} \leftarrow b^{(l)} - A^{(l)} x^{(l)} \\ b^{(l-1)} \leftarrow R^{(l-1)} d^{(l)} \end{cases}$

solution of coarse grid problem

- for $l > 1$ call of:
  $MG(l - 1, A^{(l-1)}, B^{(l-1)}, x^{(l-1)}, b^{(l-1)})$

  $\leftarrow \boldsymbol{C_{local}}$

- for $l = 1$ (exact) master solution:
  $$A^{(l-1)} x^{(l-1)} = b^{(l-1)}$$

  $\leftarrow \boldsymbol{C_{global}}$

prolongation
$$x^{(l)} \leftarrow x^{(l)} - R^{(l-1)^T} x^{(l-1)}$$

$\leftarrow \begin{cases} \boldsymbol{C_{local}}: \\ y^{(l)} \leftarrow R^{(l-1)^T} x^{(l-1)} \end{cases}$

postsmoothing:

$$x^{(l)} \leftarrow x^{(l)} - B^{(l)} \left(A^{(l)} x^{(l)} - b^{(l)} \right)$$

$\leftarrow \begin{cases} \boldsymbol{C_{local}}: \\ d^{(l)} \leftarrow b^{(l)} - A^{(l)} x^{(l)} \\ y^{(l)} \leftarrow B^{(l)} d^{(l)} \end{cases}$

6. Current Stage of Implementation

The presented SCARC ansatz was first integrated into a private copy of the current version 5.3.0 of FDS. This so-called FDS-SCARC solver was extensively checked by different test series based on various geometric situations and parameter constellations. As far as possible the attained parallel solutions were compared to the according serial ones (based on single-mesh computations) and the corresponding FFT-solutions.

The results looked very promising: The parallel solutions were notedly conformable to the corresponding serial ones, the consistency of the global volume flow was automatically guaranteed. A detailed description of the considered test series can be found in part II of this article which is devoted to the verification and validation of FDS.

Up to now, FDS-SCARC is based on a global, data-parallel CG-iteration with different block-preconditioning techniques (block-Jacobi, block-SSOR, block-FFT). This ansatz already shows a very good numercial stability and accuracy as far as the domain decomposition is isotropic or moderately anisotropic. For more irregular decompositions it will be indispensable to implement the global, data-parallel MG-iteration, especially with ADI-TRIGS-smoothing techniques, as described in *Kilian* [11]. This very important step of the implementation is currently in work and will most probably lead to a massive stabilization of the whole method even for strong anisotropic cases.

Experientially, the MG-ansatz will result in a significantly better runtime performance compared to the current CG-ansatz which still shows a runtime drawback up to a factor of 2 compared to the current FDS-FFT-version. But it shouldn't be forgot that a fast runtime is only of little value if the results aren't accurate and reliable at the same time. Finally, this ansatz is very suited to an extension to adaptive grid refinement strategies which are under development as well.

During the complete implementation phase of FDS-SCARC and all the described tests series, there was an intense intercommunion with the main developers at the National Institute of Standards and Technology (NIST) in Gaithersburg/USA. Motivated by the first achievements with the new method, they decided to include FDS-SCARC into the official FDS-repository where it will be available for the whole FDS-community for beta-testing in the medium-term.

References

[1] **Altieri, M.; Becker, C.; Turek, S.**: *Proposal for SPARSE BANDED BLAS techniques*. Technical report, Universität Heidelberg, 1999. Preprints SFB 359, Nummer 99–11.

[2] **Axelsson, O.; Barker, V. A.**: *Finite element solution of boundary value problems*. Computer Science and Applied Mathematics. Academic Press Inc., Orlando, FL, 1984. ISBN 0-12-068780-1. Theory and computation.

[3] **Becker, C.**: *Strategien und Methoden zur Ausnutzung der High-Performance-Ressourcen moderner Rechnerarchitekturen für Finite-Element-Simulationen und ihre Realisierung in FEAST (Finite Element Analysis & Solution Tools)*. PhD thesis, Universität Dortmund, Logos Verlag, Berlin, may 2007. `http://www.logos-verlag.de/cgi-bin/buch?isbn=1637,` ISBN 978-3-8325-1637-6.

[4] **Bramble, J. H.; Pasciak, J. E.; Xu, J.**: *Parallel Multilevel Preconditioners*. *Math. Comp.*, volume 55:pages 1–22, 1990.

[5] **Deuflhard, P.; Hohmann, A.**: *Numerical Analysis in Modern Scientific Computing: An Introduction*, volume volume 43 of Texts in Applied Mathematics. Springer Verlag, New York, 2003. ISBN 0387954104.

[6] **Dongarra, J.**: *Linear Algebra Libraries for High-Performance Computers: A Personal Perspective*. *IEEE Parallel Distrib. Technol.*, volume 1(1):pages 17–24, 1993. ISSN 1063-6552.

[7] **Dongarra, J. J.; Duff, L. S.; Sorensen, D. C.; Vorst, H. A. V.**: *Numerical Linear Algebra for High Performance Computers*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1998. ISBN 0898714281.

[8] **Frommer, A.**: *Lösungen linearer Gleichungssysteme auf Parallelrechnern*. Teubner, 1990. ISBN 978-3528063979.

[9] **Göddeke, D.; Buijssen, S. H. M.; Wobker, H.; Turek, S.**: *GPU Acceleration of an Unmodified Parallel Finite Element Navier-Stokes Solver*. Technical report, Fakultät für Mathematik, TU Dortmund, april 2009. Ergebnisberichte des Instituts für Angewandte Mathematik, Nummer 392.

[10] **Hackbusch, W.**: *Iterative solution of large sparse systems of equations*, volume 95 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 1994. ISBN 0-387-94064-2. Translated and revised from the 1991 German original.

[11] **Kilian, S.**: SCARC: *Ein verallgemeinertes Gebietszerlegungs–/Mehrgitterkonzept auf Parallelrechnern*. PhD thesis, Universität Dortmund, Logos Verlag, Berlin, 2002. ISBN 3-8325-0092-8.

[12] **Kilian, S.**: *Iterative Gebietszerlegungskonzepte für den FDS-Drucklöser zur Stärkung der globalen Kopplung*. 2. FDS-Usergroup Workshop, hhpberlin, Berlin, 4.-5. Dezember 2008.

[13] **Kilian, S.; Turek, S.**: *An example for parallel* SCARC *and its application to the incompressible Navier-Stokes equations*, 1997.
URL `citeseer.ist.psu.edu/article/kilian97example.html`

[14] **Klainerman, S.; Majda, A.**: *Compressible and Incompressible Fluids.* *Comm. Pure Appl. Math.*, volume 35:pages 629–653, 1982.

[15] **Klein, R.**: *Semi-Implicit Extension of a Godunov-Type Scheme Based on Low Mach Number Asymptotics I: One-Dimensional Flow. Journal of Computational Physics*, volume 121:pages 213–237, 1995.

[16] **Klein, R.; Botta, N.; Schneider, T.; Munz, C.-D.; Roller, S.; Meister, A.; Hoffmann, L.; Sonar, T.**: *Asymptotic adaptive methods for multi-scale problems in fluid mechanics. Journal of Engineering Mathematics*, volume 39:pages 261–343, 2001.

[17] **McDermott, R.; McGrattan, K. B.; Hostikka, S.; Floyd, J.**: *Fire Dynamics Simulator (Version 5) – Technical Reference Guide, Volume 2: Verification*. Technical report Volume 2, National Institute of Standards and Technology, Building and Fire Research Laboratory, February 2009.

[18] **McGrattan, K. B.; Hostikka, S.; Floyd, J.; Baum, H.; Rehm, R.; Mell, W.; McDermott**: *Fire Dynamics Simulator (Version 5) Technical Reference Guide, Volume 1: Mathematical Model*. National Institute of Standards and Technology, Building and Fire Research Laboratory, 5.2 edition, February 2009.

[19] **Münch, M.**: *Ist eine Gebietszerlegung in mehrere Gitter bei FDS-Simulationen zulässig?*. 2. FDS-Usergroup Workshop, hhpberlin, Berlin, 4.-5. Dezember 2008.

[20] **Münch, M.; Kilian, S.**: *A new generalized domain decomposition strategy for the efficient parallel solution of the FDS-pressure equation, Part II: Verification and Valdiation*. Technical report in preparation, Konrad-Zuse-Zentrum für Informationstechnik Berlin, 2009.

[21] **Münch, M.; Klein, R.**: *Critical numerical aspects for field model applications. EUSAS-Journal*, volume 4:pages 41–54, July 2008.

[22] **Münch, M.; Schmidt, H.; Oevermann, M.; Klein, R.**: *Numerical Investigation of the Baroclinic Torque in the Context of Smoke Spread*. In *Abstracts of Work-In-Progress Poster Presentations, Thirty-First International Symposium on Combustion, University of Heidelberg, Germany*, page 174. The Combustion Institute, 5001 Baum Boulevard, Suite 635, Pittsburgh PA 15213 - 1851, August, 6 - 11 2006.

[23] **NIST**: *Fire Dynamics Simulator (FDS) and Smokeview.*
URL `http://fire.nist.gov/fds`

[24] **Rannacher, R.**: *Numerische Mathematik 2*, 2006.
URL `http://numerik.iwr.uni-heidelberg.de/\~lehre/SS06/numer%ik2/`

[25] **Rehm, R. G.; Baum, H. R.**: *The Equations of Motion for Thermally Driven Buoyant Flows. Journal of Research*, volume 83(3):pages 297–308, 1978.

[26] **Rogsch, C.**: *Parallelrechnung in der Praxis - Ist FDS5 wirklich besser als FDS4?*. 1. FDS-Usergroup Workshop, hhpberlin, Berlin, 6.-7. März 2008.

[27] **Saad, Y.**: *Iterative methods for sparse linear systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, second edition, 2003. ISBN 0-89871-534-2.

[28] **Schochet, S.**: *The mathematical theory of low Mach number flows. M2AN*, volume 39(3):pages 441–458, 2005. +DOI: 10.1051/m2an:2005017.

[29] **Schwarz, H. R.**: *Numerische Mathematik*. Teubner, Stuttgart, 4. edition, 1997. ISBN 3-519-32960-3.

[30] **Turek, S.**: *Efficient Solvers for Incompressible Flow Problems: An Algorithmic and Computational Approach*. Springer, Berlin, 1999.