

MATTHIAS MÜNCH SUSANNE KILIAN

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

**Part II:
Verification and Validation**

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

Part II: Verification and Validation

Matthias Münch^a, Susanne Kilian^b

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

^b*hhpberlin, Ingenieure für Brandschutz GmbH
Rotherstr. 19, 10245 Berlin*

Abstract

Because CFD programs, like FDS, generally consist of a large number of different components representing the variety of participating numerical algorithms and chemical / physical processes, it is nearly impossible to verify such codes in their entirety, for example with comparisons of fire tests. Instead, a careful verification and validation with respect to the underlying mathematical conditions and applied numerical schemes is indispensable. In particular, error cancellations between single program components can only be detected by such detailed component-level tests.

In part I [7] of this article series a conceptual deficiency of the FDS program package with regard to multi-mesh computations was illustrated and an alternative domain decomposition strategy FDS-SCARC was introduced. In this second part we will present the structure of a comprehensive test concept and the needs for a more mathematically and numerically orientated test procedure that is much more suited for a reliable evaluation than only a simple visual comparison of the numerical results with experimental fire tests.

After a general introduction of our test concept we will demonstrate the high potential of the new FDS-SCARC technique compared to the FDS-FFT technique which is used in the FDS program package as yet. Based on this concept, we will present a comprehensive set of analytical and numerical test results.

Key words: CFD, domain decomposition, Fire Dynamics Simulator (FDS), FFT, SCARC, Validation, Verification

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

1. Introduction

In part I of this series [7] we explained a conceptual lack of the current FDS program package concerning the geometric decomposition of the computed domain into smaller subdomains or meshes. We presented a new generalized domain decomposition strategy for the efficient parallel solution of the FDS-pressure equation that guarantees the necessary accuracy. In this second part, we describe a comprehensive test methodology and first tests to prove the correctness of this new strategy.

Due to a positive experience with certification processes for tools and components for civil engineering, it seems appropriate to develop analogous quality assessment procedures for fire safety-related CFD programs as described in [16]. In fact, in analogy with fire tests for structural elements, the comparison of CFD-based simulation data with measurements from fire experiments has become one standard approach for testing the applicability of such codes.

However, on the basis of such global comparisons it is not possible to decide whether a CFD code produces good results because it is really correct, or simply because of internal error cancelation. Unfortunately, the required details of the flow fields are often plainly inaccessible due to a lack of appropriate measurement techniques, e.g., in the presence of intense smoke. Thus, quality assessments of CFD codes for fire safety should not rely exclusively on comparisons with experimental results. This conclusion is further supported by the fact that only a limited range of flow regimes can be realized in the laboratory. As a consequence, even if a CFD code has positively passed scrutinizing tests based on comparisons with a large experimental data base, there is no guarantee whatsoever that it will work equally well in flow regimes which the experiments have not covered. For example, it is an open issue whether fire events in very large open-space buildings can be downscaled to laboratory sizes while maintaining all the rules of similarity.

Another disadvantage concerning a detailed justification of a CFD programs is the fact that data from fire experiments always consider net effects of all physical processes of a fire. Therefore cancelation of errors inside the computational results may remain undetected.

Especially the simulation of fire and smoke spreading requires the modelling of complicated physical and chemical processes, which are partially not really well understood. For this reason, the developers of such programs use empirical models as well as many approximations to limit the computational costs in an appropriate range. Furthermore, there is a strong non-linear coupling between these processes, for example between turbulence, combustion and radiation.

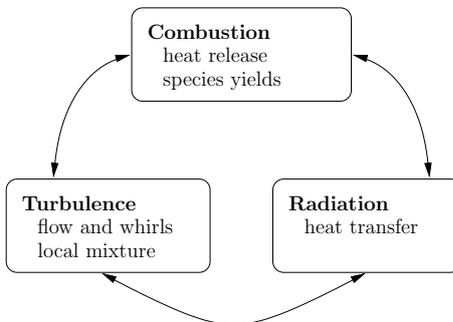


Figure 1: Process interaction

And last but not least it is possible that because of the limited range of experimental facilities errors outside this range can not be detected.

In summary, comparisons with fire experimental data are necessary and useful. But, if adopted as the only means of testing they are insufficient to document the performance of CFD programs. Therefore, a more comprehensive testing strategy is indispensable.

1.1. Component-level and isolated process tests

Verification and validation (V&V) is widely discussed in the CFD community (e.g., [1, 5, 8, 17]). It is beyond the scope of this paper to summarize the discussion regarding different definitions of verification and validation. In contrast, we aim to provide some evidence why other types of tests are necessary to ensure reliable results from CFD programs. Figure 2, from *Schlesinger* [19], illustrates various facets of what we will expand upon in the present text.

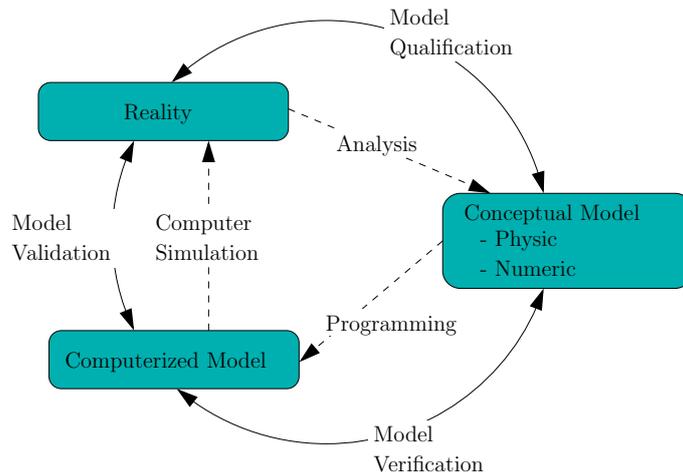


Figure 2: The issue of verification and validation [19]

1. **Model Qualification:** is the process of determining whether an adopted conceptual model accurately represents the real world as far as its intended use is concerned. To do so, the conceptual model should include descriptions of all physical system components and processes that are of interest for the intended use. Conceptual models for CFD consist of the equations of fluid dynamics extended by auxiliary model equations, e.g., for turbulence and chemical reactions, and of initial and boundary conditions, [17].
2. **Model Verification:** is the process of determining whether a computerized model accurately represents the developer's conceptual model and its solutions [1]. The fundamental goal of verification is the identification and quantification of errors in the computational model and its solution. In verification activities, the accuracy of a computational solution is primarily measured relative to two types of references: analytical solutions and highly accurate numerical solutions [17].

3. **Model Validation:** is the process of determining the degree to which the computerized model is an accurate representation of the real world from the perspective of the intended utilization [1]. The strategy of validation is to assess how accurately computational results match with experimental data, with quantified error and uncertainty estimates for both [17].

Because computerized models, namely the CFD programs, generally consist of very large numbers of different components representing the variety of participating system components and processes, it is close to impossible to verify a CFD program in its entirety. Instead, careful verification and validation of single components as well as groups of components of increasing complexity are imperative. In particular, error cancelations between a program's components can only be detected by such component-level tests.

To illustrate the scope of the issue, here is a sample of the components of a CFD program which will require individual assessment:

- **Physical submodels:** turbulence, radiation, boundary conditions, . . .
- **Numerical algorithms:** flux functions, time integrators, linear algebra solvers, . . .
- **Data handling components:** data structures, parallelization, load balancing, . . .
- **Grid handling components:** discretization techniques, domain decomposition, grid refinement, . . .

All these components interact in various ways, so that component-level tests must process the components themselves as well as the interactions between them.

Although all steps are important, the present paper focuses on the verification issues to test the implementation of the new FDS-SCARC scheme in comparison to the current scheme. Furthermore, we will give a rough introduction of some useful strategies to prove the quality of numerical schemes which are much more suited for a reliable evaluation than only a simple optical comparison of the numerical results.

2. Test of numerical qualities

Because the current von-Neumann computer can not handle partial differential equations, numerical discretization schemes play an important role for the CFD code's quality. Our comprehensive testing strategy considers the requirements needed to use these discretization techniques in the right way. In the next subsections we will present a short introduction into the background story of numerical qualities that are useful to test CFD programs.

2.1. Boundary conditions

Algebraic equations, e.g. the discretization algorithm to compute a gradient (3), are mostly easy to implement in computational areas far away from the boundary of the domain. However, more or less complicated situations arise in areas near the boundary. In the example below we have to use a value outside the computational domain, a so-called ghost value, if the gradient along the boundary is computed. In this context, code developers often have to use suitable assumptions in order to solve such problems.

To give an example, suppose that a cell-centered value φ is used. To compute the face-centered gradient of φ in first order accuracy we use equation (3). Inside the computational domain we can compute the gradient at the cell face A without any problem. At the boundary the situation is somewhat more tricky. To compute the difference of φ at the cell face B we need a value of φ outside of the computational domain or another formula to compute the gradient at the boundary. Therefore boundary conditions can play a major role for the quality of numerical results.

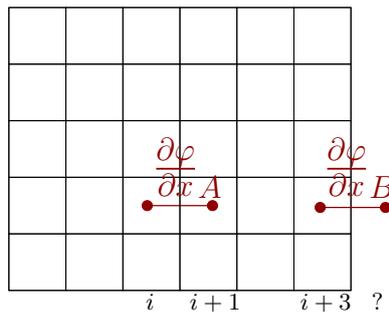


Figure 3: Example: Compute the gradient of φ .

Although periodic boundary conditions are of small importance for practical fire safety engineering problems, they are important to test numerical schemes because they eliminate the influence of boundary effects, such that the quality of the pure solver method can be evaluated. Meanwhile this type of “theoretical” boundary condition is implemented in the source code of FDS.

2.2. Consistency, convergence and stability

As demonstrated in the previous paper [7] of this series, numerical discretization schemes have a crucial influence on a CFD code’s quality. It is beyond the scope of the present paper to provide more than a rough overview of the related theory for convergence investigations, but some basics are necessary to understand the main principles, see e.g. [21].

The current von-Neumann computer operates with a finite precision representation of real numbers called floating-point numbers. It can form only a finite number of such floating point values and can store only a finite number of them in its memory space. As a consequence, it cannot handle continuum problems described by differential equations such as

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial x} = 0 \quad (1)$$

directly. Instead, the continuum equations are approximated by discrete analogues through a “discretization scheme”. A common way to derive discrete approximations, e.g., for the partial derivatives in (1), uses Taylor series expansions. The Taylor expansion for $\varphi(x_{i+1})$ of a function $\varphi(x)$ around $x = x_i$ reads as

$$\varphi_{i+1} = \varphi_i + \left(\frac{\partial \varphi}{\partial x}\right)_i \cdot \Delta x + \left(\frac{\partial^2 \varphi}{\partial x^2}\right)_i \cdot \frac{\Delta x^2}{2!} + \left(\frac{\partial^3 \varphi}{\partial x^3}\right)_i \cdot \frac{\Delta x^3}{3!} + \dots \quad (2)$$

A change in the sequence of terms leads to an approximation for the gradient of φ at the point i

$$\left(\frac{\partial \varphi}{\partial x}\right)_i \approx \frac{\varphi_{i+1} - \varphi_i}{\Delta x} - \underbrace{\left(\frac{\partial^2 \varphi}{\partial x^2}\right)_i \cdot \frac{\Delta x}{2!} - \left(\frac{\partial^3 \varphi}{\partial x^3}\right)_i \cdot \frac{\Delta x^2}{3!} - \dots}_{\text{truncation error } \mathcal{O}(\Delta x)} \quad (3)$$

For the approximation of the gradient of φ only a finite number of terms in (3) can be considered. The rest is necessarily neglected and remains as a “truncation error”. This type of discretization is widely used in CFD programs and there exists a big range of concepts for the justification of the accuracy and correctness of the resulting numerical approximation schemes.

- **Order of consistency:** The quality of the numerical solution will depend on the order of the scheme, described by the truncation error. The “order of a discretization” is determined by the power of the discretization parameter (here Δx) that appears in the first neglected term of the Taylor expansion. Therefore, the discretization in (3) is of first order $\mathcal{O}(\Delta x)$. For modern CFD programs second order discretizations are state of the art.
- **Convergence:** 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, and at a rate determined by the order of the scheme,

$$\lim_{\Delta x \rightarrow 0} \left(\frac{\partial \varphi}{\partial x} - \frac{\Delta \varphi}{\Delta x} \right) = 0. \quad (4)$$

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 theoretical tests, one cannot expect that it produces reliable results for realistic problems. Therefore, authorities should insist on a detailed documentation of convergence tests before accepting data derived from numerical simulations.

- **Stability:** There will be a large number of input data x defined by the user, for many of which only coarse estimates will be available. Essentially, a numerical scheme $F(x)$ for evaluating a function $f(x)$ is called stable if small input errors result in controlled, small changes in the computed output, i.e., $|F(x + \delta x) - F(x)| \rightarrow 0$ as $\delta x \rightarrow 0$. In the graphical illustration the stability of the numerical scheme means that the ratio between the hatched area of the input deviation and the hatched area of the total deviation of the numerical result must be limited.

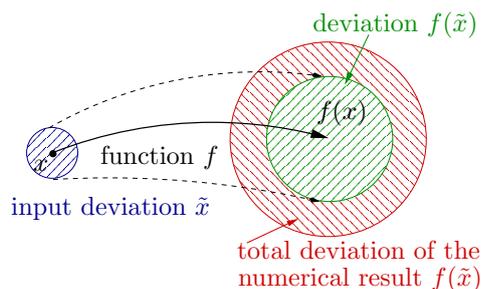


Figure 4: Stability and error propagation

All these requirements have been known for a long time. Already in 1902 the French mathematician Hadamard identified consistency, convergence and stability as necessary conditions for a useful mathematical model.

With regard to the discussion of the convergence order in section 4 we would like to point out that the hydrodynamic solver in FDS consists on more than only the pressure solver which was solved with the FFT scheme as yet. Therefore potential deficiencies in the remaining components of FDS are not affected by the new SCARC scheme by which the FFT scheme was replaced. This means the improvement of the new scheme can only be seen in a relative comparison between FDS-FFT and FDS-SCARC, rather than in absolute values.

Additionally, the efficiency of the code is another very important quality criterion because the computational results must be available in a reasonable time. Ultimately, everything depends on the correctness of the underlying numerics.

2.3. Practical relevance

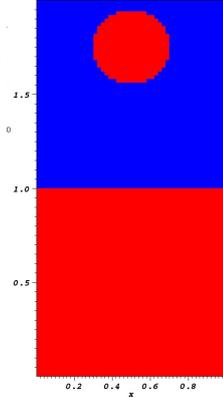


Figure 5: Falling droplet

Apart from the mathematical and numerical argumentation, there is even a physical relevance. An often used argument of practitioners is that as long as important input data can only be roughly estimated, digits after the decimal point can be neglected. In the context of numerical schemes this argumentation fails. Even the small terms of the Taylor series (3) represent physical properties. This should be demonstrated by a simple numerical experiment, the falling droplet test.

Let us assume that we drop a droplet into a fluid surface, as shown in figure 5. Now we simulate the falling droplet. For this purpose, we use a CFD program [13], which can switch between first and second order accuracy by neglecting the second term (1st order) or third term (2nd order) in the approximation of the gradient (3).

Now we compare the density and velocity field after the impact of the droplet on the fluid surface at the same time. Whereas in the right picture (2nd order) a compact wave rolls to the right, the left picture (1st order) looks more like a small fluid hill with a flow along a line to the right upper edge of the picture.

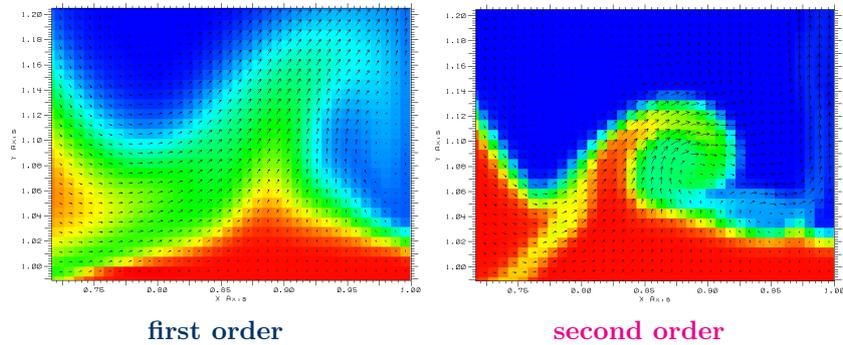


Figure 6: Comparison of a first and second order solution

The reason is that the approximation of a curvature, necessary to form waves, needs the second derivative. Nevertheless, this curvature term is neglected in the first order approximation.

$$\left(\frac{\partial\varphi}{\partial x}\right)_i \approx \underbrace{\frac{\varphi_{i+1} - \varphi_i}{\Delta x}}_{\text{1st order approximation}} - \underbrace{\left(\frac{\partial^2\varphi}{\partial x^2}\right)_i \cdot \frac{\Delta x}{2!} - \left(\frac{\partial^3\varphi}{\partial x^3}\right)_i \cdot \frac{\Delta x^2}{3!}}_{\text{2nd order approximation}}$$

2.4. Usefulness for code testing

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 to solve 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 the case of FDS the scheme should be of second order accuracy in time and space (see [9]). Therefore convergence tests provide an appropriate quality criteria to prove CFD programs.

3. Concept and Strategy

One important feature of the FDS program package is the possibility to decompose the computational 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 usage of multi-meshes in serial as well as parallel simulations in FDS may cause inaccuracies or instabilities, as demonstrated by different authors e.g. [3, 6, 7, 12, 18].

These errors result from deficiencies in the domain decomposition strategy in conjunction with the FFT-solver used to solve the pressure equation in FDS. Therefore completely new strategies for the solution of the pressure equation should be developed. As a consequence a new parallelization concept, the generalized domain decomposition/multigrid method SCARC, was presented in part I [7].

In the present part we are describing a comprehensive test strategy to prove the correctness of this new strategy. Following the idea of component-level tests and the described V&V rules, the test strategy focuses on the hydrodynamic solver and the domain decomposition method first. Nevertheless the concept can be expanded by testing other submodules and solvers as demonstrated in [11].

3.1. Classification

There will be different sources for reference data, which can be used for V&V work. The presented concept distinguishes between:

A Analytical tests

The results of these analytical tests are known because of mathematically or numerically based considerations. One example is the “PIPE_2D” test described in subsection 4.3.

SE **Semi-experimental tests**

Semi-experimental tests are based on a clearly restricted number of physical or chemical processes / submodels to proof the interaction. For example we focus only on heat conduction.

N **Numerical tests**

Numerical tests are comparisons with results from more detailed or higher qualified programs. See the example in subsection 4.3, “CD_VA_2D” test example.

E **Experimental tests**

These are small- or full-scale fire tests as well as complex buoyancy-driven fluid flow experiments.

To realize the component-level strategy the classification differentiates between the physical and chemical processes and more numerical criteria like order, convergence and symmetry. Additionally the implementation of boundary conditions plays an important role for error-detection. At the current state, we subsume these criteria under the term “structure test” (DD: domain decomposition, OC: order and convergence, PA: parallelization, BC: boundary condition, SY: symmetry). A comprehensive test table will be presented at the end. The classification of each test is described in this table.

Test	Type	Physical. components							
		Gasdynamic	Gravitation	Viscosity	Radiation	Turbulence	Combustion	...	Structure test
CD_NSA_2D	A	✓							DD,OC
CD_VA_2D	N	✓							DD,OC
Pipe_2D	A	✓							DD,OC,BC
...									

Table 1: V & V Test table

4. Tests

To demonstrate the advantages of the new generalized domain decomposition technique FDS-SCARC and the power of the numerically orientated test strategy, we are presenting the results of some test examples. For all computations we use the official code version 5.4.3 revision 5210. This code version is the basis of the official binary version FDS 5.4.3 provided by the NIST download server.

Following the concept described in chapter 3 we are differentiating between single components of the FDS scheme. Because the FDS-SCARC technique replaces the FFT pressure solver, an important component of the gasdynamic solver scheme of FDS, we are focusing our tests on this part of the hydrodynamic

solver and the boundary conditions involved. Especially the numerical scheme for the turbulence modeling must be switched off in order to prevent errors resulting from its empirically based modus operandi. As a result we get the characteristics of the pure gasdynamic scheme. The gasdynamic scheme is of great importance because all chemical and physical schemes, e.g. turbulence, combustion, radiation, . . . , are based on it. Even if we neglect these components here, they must be proved in a similar way.

On the base of the previous limitations we are considering the time integration of variable density non-reaction flows. As described in subsection 2.1, boundary conditions can play a major role for the quality of numerical results. Although periodic boundary conditions are of small importance for practical fire safety engineering problems, they are very well suited for the test of numerical schemes, because they eliminate the influence of boundary effects.

4.1. *CD_NSA_2D test example*

This analytical test case is widely used by developers of numerical schemes to test the advection properties of a gasdynamic solver, e.g. [2, 10, 13, 22]. The main developers used this case in the FDS verification guide for a serial 1-mesh-geometry in order to demonstrate the second-order accuracy of the underlying numerical scheme.

Furthermore, an accurate advection of vortices is an important prerequisite for the simulation of smoke spreading. Therefore, it's an ideal test case for the new FDS-SCARC scheme, especially with respect to its scalability towards higher numbers of meshes.

The following tests are based on the same initial conditions as the test in the FDS verification guide. Because the current concept for the definition of FDS-input files doesn't offer the possibility to define the corresponding initial conditions, they had to be hardcoded in the FDS source code. The test case describes a viscous-free advection of vortices in a simple square in two dimensions with periodic boundary conditions. We use a uniform grid with $\Delta x = \Delta y$. The physical domain of the problem is a square of length $L = 2\pi$. The grid spacing is uniform $\Delta x = \Delta y = L/N$ in each direction with $N = \{16, 32, 64, 128\}$.

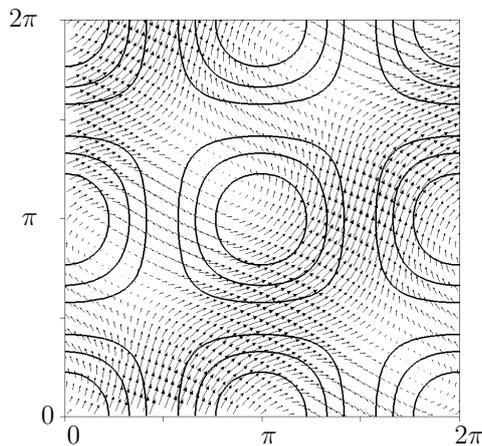


Figure 7: Initial velocity \mathbf{u} (arrows) and vorticity ω (contourlines) for $[-3, -2, -1, 1, 2, 3]$

The solution is spatially periodic on an interval 2π in each direction, and is temporally periodic on 2π .

Because of the absence of physical viscosity this test is suitable to prove the numerically indicated dissipation of the scheme. Figure 7 shows the velocity field and an isoline plot of the vorticity. The vorticity is a characteristic value for the location and magnitude of vortices in a flow field. Mathematically speaking, it is the curl of the flow velocity $\omega = \nabla \times \mathbf{u}$, a vector quantity, whose direction corresponds to the axis of the flow field rotation. In our two-dimensional test case, the vorticity vector is perpendicular to the $x - y$ plane.

From a physical point of view the vorticity is a quantity to describe the spin of a flow, which influences the mixing process (e.g. of species concentrations, particles, ...). Therefore the correct computation of vortices transport processes is an important issue for CFD programs in the area of fire safety engineering.

The initial and analytical solution for this test case is

$$\begin{aligned} \rho(x, y, t) &= 1 \\ u(x, y, t) &= 1 - 2 \cos((x - t)) \sin((y - t)) \\ v(x, y, t) &= 1 + 2 \sin((x - t)) \cos((y - t)) \\ p(x, y, t) &= 1 \\ \tilde{p}(x, y, t) &= -\cos(2(x - t)) - \cos(2(y - t)). \end{aligned} \quad (5)$$

Therefore the density ρ is constant and the divergence constraint of the initial velocity field $\mathbf{u} = [u, v]^T$ is

$$\nabla \cdot \mathbf{u} = 0. \quad (6)$$

Instead of the hydrodynamic pressure \tilde{p} FDS uses the variable \mathcal{H} as a modified pressure term to solve the divergence constraint (see part I [7]). Therefore the corresponding analytical solution for \mathcal{H} is

$$\mathcal{H} = \frac{|\mathbf{u}|^2}{2} + \frac{\tilde{p}}{\rho_\infty}. \quad (7)$$

However, for the verification it is more useful to investigate the velocity \mathbf{u} and the hydrodynamic pressure \tilde{p} separately.

4.1.1. Test of the hydrodynamic solver

Subsequently, different subdivisions of the underlying computational domain into $M \times M$ submeshes will be considered. Starting with $M = 1$ we will investigate the solution of the hydrodynamic solver in an one-mesh computation. To this end, we will use FDS in the DNS mode and switch-off the viscosity term. These are important prerequisites to force the FDS program to use only the hydrodynamic solver and neglect the LES turbulence model. Because all sides of the domain are of simple periodic type we can prevent errors from boundary condition implementations.

A main criterion to prove the correctness of the solver is the advection of the vortices. The vortices move diagonally from the lower left to the upper right

edge of the domain. Figure 8 compares the isolines of the vorticity at the initial time $t = 0$ and after one period $t = 2\pi$.

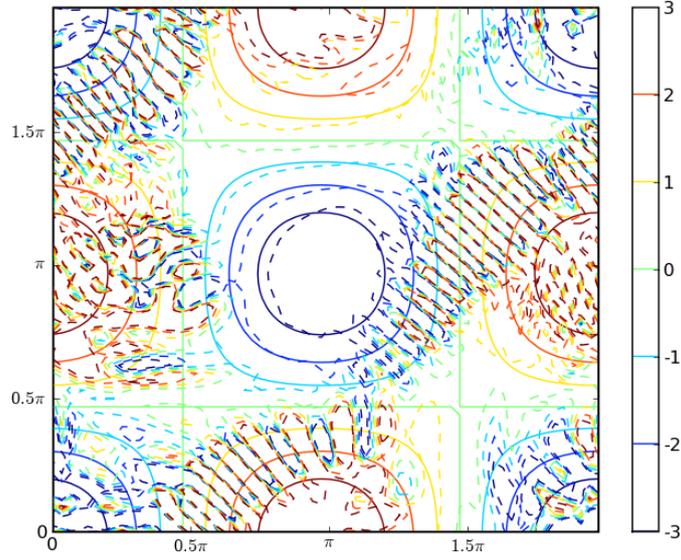


Figure 8: Isolines of the vorticity of an one-mesh $N = 64$ FDS-FFT computation at $t = 0$ (continuous lines) and $t = 2\pi$ (dashed lines) for $[-3, -2, -1, 1, 2, 3]$.

To analyze the computed solution we plot the vorticity along the diagonal line from $(0, 0)$ to $(2\pi, 2\pi)$ after one period.

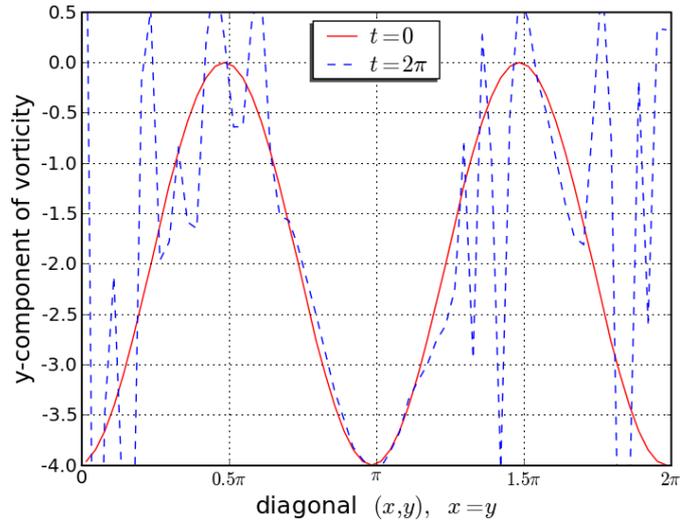


Figure 9: Vorticity of an one-mesh $N = 64$ FDS-FFT computation at $t = 0$ (continuous lines) and $t = 2\pi$ (dashed lines) along the diagonal from $(0, 0)$ to $(2\pi, 2\pi)$.

Figure 9 shows an insufficient vortex advection after one period. From a theoretical point of view the FDS hydrodynamic solver should be able to compute this test case very well. Because we used a self-defined FDS input file we compared our input file with that of the verification test case documented in the FDS verification guide. In this input file the FDS developers used the additional condition $CFL_MAX = 0.25$ to limitate the CFL time step computation. And indeed, with this additional limitation in our input file FDS computes the test case very well.

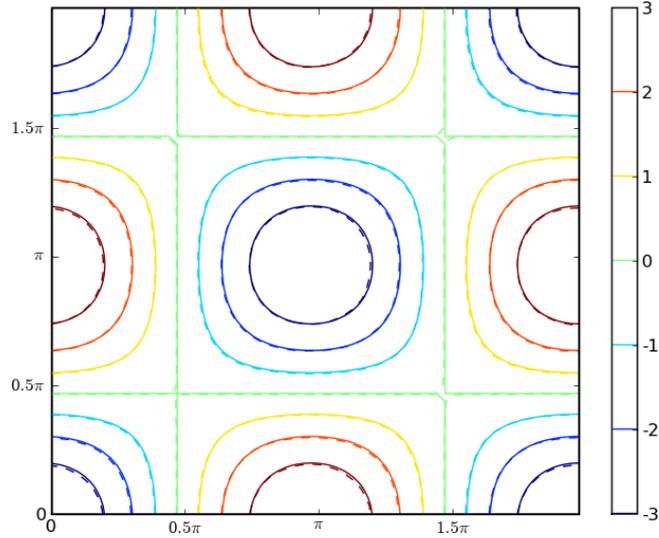


Figure 10: Isolines of the vorticity of an one-mesh $N = 64$ FDS-FFT computation at $t = 0$ (continuous lines) and $t = 2\pi$ (dashed lines) with $CFL_MAX = 0.25$ for $[-3, -2, -1, 1, 2, 3]$.

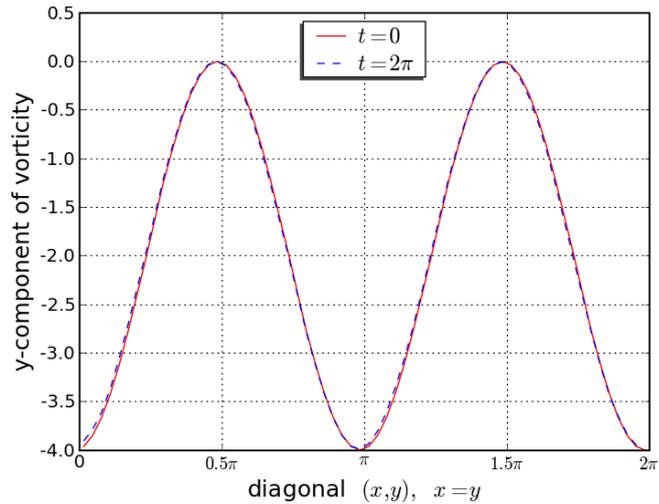


Figure 11: Vorticity of an one-mesh $N = 64$ FDS-FFT computation at $t = 0$ (continuous lines) and $t = 2\pi$ (dashed lines) along the diagonal from $(0, 0)$ to $(2\pi, 2\pi)$ with $CFL_MAX = 0.25$.

The bad results in case that no additional “CFL_MAX-restriction” is used seem to be caused by an inadequate adjustment of the time step computation which obviously results in too big time intervals. To compute the time step for a numerically solved advection problem the Courant Friedrich Levy (CFL) condition is a widely used limitation algorithm. The condition should prevent a wave with a finite travelling speed from jumping over a discrete finite volume (cell) during one time step. The time step must be less than the time the wave needs to travel to the adjacent finite volume.

$$\frac{u_{max} \Delta t}{\Delta x} < 1. \quad (8)$$

The CFL condition is a necessary but not sufficient condition for stability. To be conservative an additional factor, the CFL number ($0 < CFL < 1$), can be used to adjust the time step computation to smaller ranges. Of course, the computational costs increase with smaller time steps.

To get adequate computational results the FDS developers used the additional limitation “CFL_MAX = 0.25” in this test case. This option adjusts the computed time step of FDS to guarantee

$$CFL_MAX \geq \Delta t \text{MAX} \left(\frac{|u|}{\Delta x}, \frac{|v|}{\Delta y}, \frac{|w|}{\Delta z} \right). \quad (9)$$

As shown in figure 11 the computed results of FDS-FFT reproduce the analytical solution of (5) after a time period of $t = 2\pi$ very accurately with this additional time step limitation. The solution is periodic in time and the difference between the isolines of the advected vorticity are very small.

Important remark for practitioners

This result indicates the basic correctness of the FDS implementation of the advection scheme for nonviscous flows with constant density. But it must be remarked that this is only true with the problem-dependent additional time step limitation mentioned above, which most probably will never be used for practical problems by any user. With the default time step computation FDS damps out vortices. Furthermore, only a convergence study can show if FDS is of second order accuracy for this problem.

4.1.2. Test of the domain decomposition

To compare the analytic solution with the numerical results in conjunction with the domain decomposition method, the domain is split up to $M = 2, 4$ and 8 subdomains in x - and y - direction as shown in figure 12.

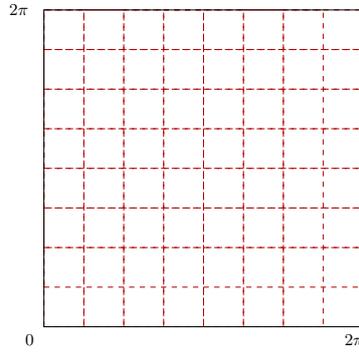


Figure 12: Domain decomposition

As in the previous subsection, we are using the advection of the vorticity ω as reference criterion. Therefore we investigate the vorticity values along the diagonal from $(0, 0)$ to $(2\pi, 2\pi)$ for different topologies. As shown in figure 13 and 14 the deviations between the analytical and advected diagonal vorticity seem to be small for both schemes. As mentioned before we need a convergence study to prove the second order accuracy.

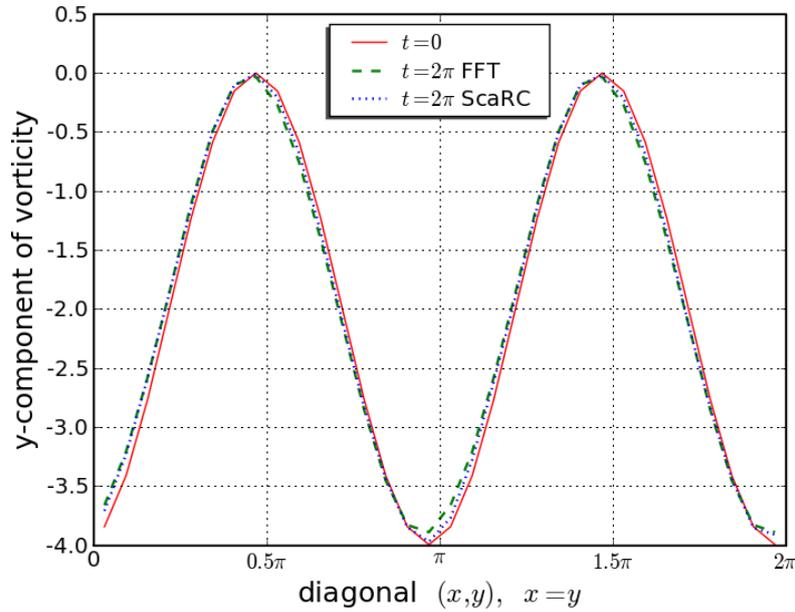


Figure 13: Vorticity of a 4×4 -mesh $N = 32$ FDS-FFT and FDS-SCARC computation at $t = 0$ (continuous lines) and $t = 2\pi$ (dashed lines) along the diagonal from $(0, 0)$ to $(2\pi, 2\pi)$ with $CFL_MAX = 0.25$.

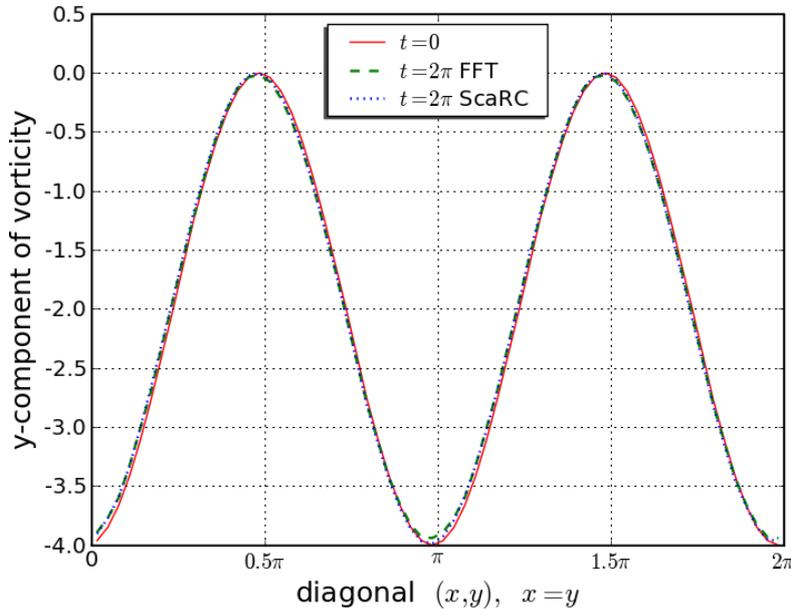


Figure 14: Vorticity of a 8×8 -mesh $N = 64$ FDS-FFT and FDS-SCARC computation at $t = 0$ (continuous lines) and $t = 2\pi$ (dashed lines) along the diagonal from $(0, 0)$ to $(2\pi, 2\pi)$ with $CFL_MAX = 0.25$.

4.1.3. Convergence study

At first sight there seems to be a good visual correspondence for FDS-FFT in case of different $M \times M$ subdivisions. However, a closer look reveals degradations in the approximation quality leading away from the second order convergence when the number of subdomains is increased. In contrast to that FDS-SCARC shows a consistent convergence behavior of second order independent of the number of subdomains. This fact is illustrated in the figures 15 and 16, which show the L2-errors of FDS-FFT and FDS-SCARC for the velocity in x-direction in the case of the 4×4 - and 8×8 -subdivisions in logarithmic representation. Due to symmetry reasons the results for the velocity in y-direction are principally the same and are omitted for the sake of simplicity.

Both figures are related to a sequence of different grid sizes N which are displayed by dots (FDS-SCARC) and squares (FDS-FFT). Figure 15 is based on $N = 16, 32, 64$ and 128 cells per direction. In contrast to that figure 16 only relies on $N = 32, 64$ and 128 cells because the case with $N = 16$ cannot be subdivided into 8×8 submeshes. For a more suggestive visualization, both figures contain comparative lines indicating first and second order convergence.

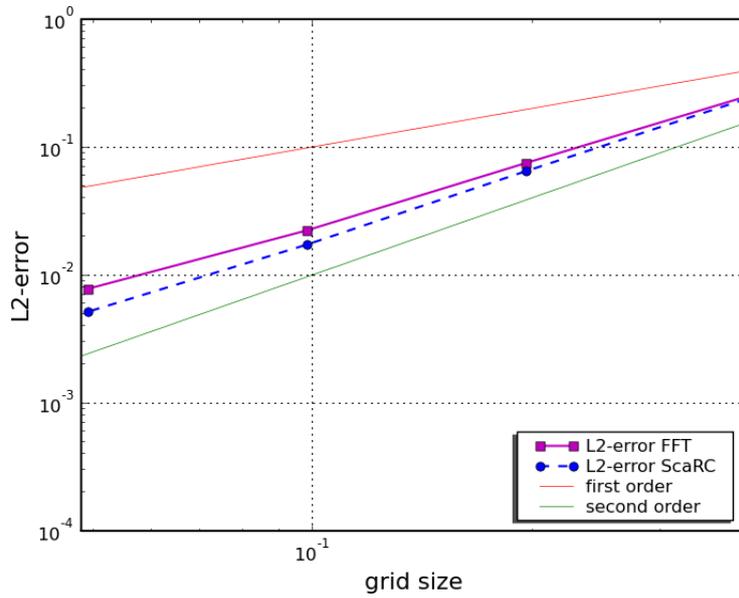


Figure 15: L2-errors of a 4×4 -mesh FDS-FFT and FDS-SCARC computation with $CFL_MAX = 0.25$ for $N = 16, 32, 64$ and 128 .

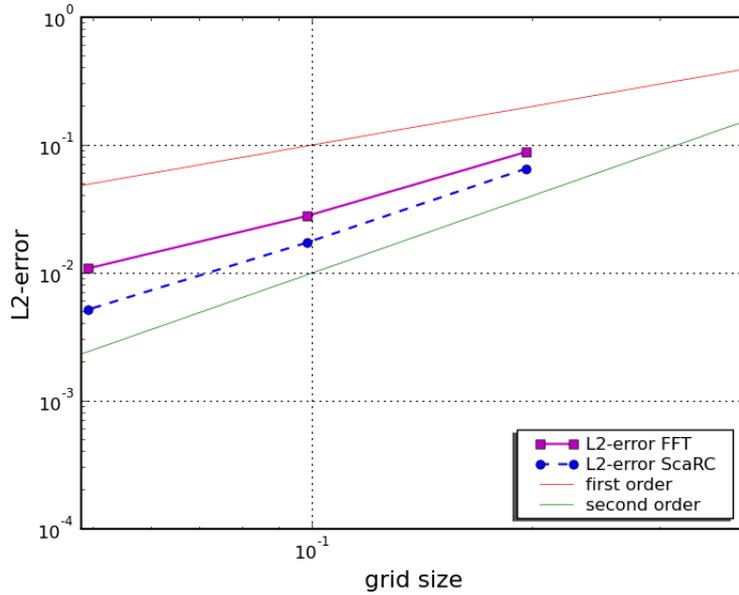


Figure 16: L2-errors of an 8×8 -mesh FDS-FFT and FDS-SCARC computation with $CFL_MAX = 0.25$ for $N = 32, 64$ and 128 .

Second order convergence clearly implies that the resulting convergence graph is parallel to the comparative line of second order convergence which obviously holds true for FDS-SCARC. In contrast, the slope of FDS-FFT approaches more and more the comparative line for first order convergence.

4.2. CD_VA_2D test example

This test case considers the advection of a vortex by a constant background flow, originally proposed by *Gresho and Chan* [4] and used by e.g. [11, 20, 22]. In the original setting the domain is rectangular with size $[0, 4] \times [0, 1]$ and has periodic boundary conditions at the short sides and walls at the long sides. In comparison to the CD_NSA_2D test case there is now an interaction with free-slip boundary conditions at the long sides.

The initial conditions are

$$\begin{aligned} \rho &= 1 \\ u &= u_{adv} - v_\theta \sin(\theta) & v_\theta &= \begin{cases} 5r u_{max}, & \text{for } 0 \leq r < \frac{1}{5} \\ (2 - 5r) u_{max}, & \text{for } \frac{1}{5} \leq r < \frac{2}{5} \\ 0, & \text{for } \frac{2}{5} \leq r \end{cases} \\ v &= v_\theta \cos(\theta) \\ p &= 1 & u_{adv} &= u_{max} = 1 \end{aligned}$$

The radius r is computed by $r = \sqrt{(x - \frac{1}{2})^2 + (y - \frac{1}{2})^2}$ and θ is the deflection angle of r . Figure 17 shows the velocity distribution of the vortex if $u_{adv} = 0$, in the test case we are using $u_{adv} = 1$ of course.

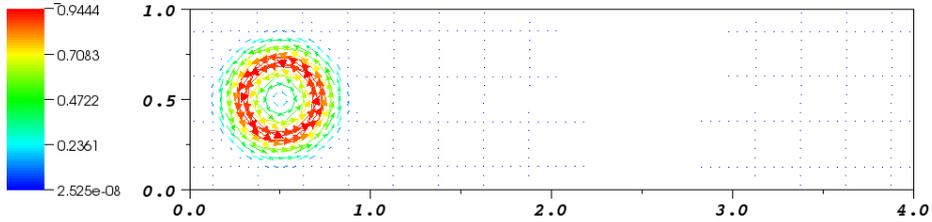


Figure 17: Initial velocity data with $u_{adv} = 0$.

The vortex rotates with the maximal tangential velocity u_{max} and is advected with the velocity u_{adv} . Therefore the exact velocity data for $t > 0$ can be computed by $u(x, y, t) = u(x - u_{adv}t, y, 0)$ and $v(x, y, t) = v(x - u_{adv}t, y, 0)$. The computational domain of the original test case consists of 80×20 grid cells and was performed on a uniform grid with $\Delta x = \Delta y$.

4.2.1. Test of the hydrodynamic solver

To analyze the hydrodynamic solver we first compute the solution in an one-mesh case. As in the previous test case we are using the DNS mode and switching off all other physical submodules or components. Because this flow is a nonviscous flow, the vortex must be simply advected, therefore we are using the vorticity as a reference quantity. Based on the experience gained in the previous test case “CD_NSA_2D” we are using the additional time step constraint “CFL_MAX=0.25” right from the beginning. Figure 18 shows the isolines for the computed FDS-FFT solution at $t = 0, 1, 2$ and 3 for the one-mesh case.

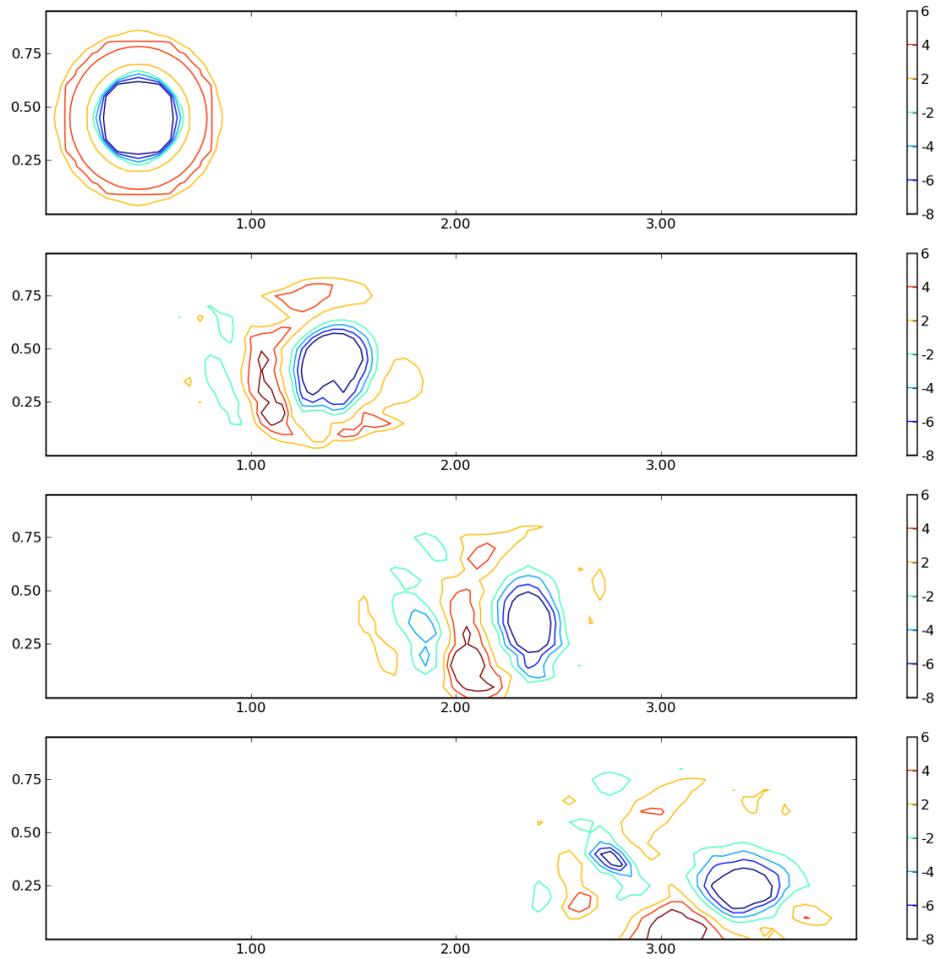


Figure 18: Isolines of the vorticity of an one-mesh $N = 80 \times 20$ FDS-FFT computation at $t = 0, 1, 2$ and 3 with $CFL_MAX = 0.25$ for $[-8, -6, -4, -2, 2, 4, 6]$.

As figure 18 shows the advected vortex smears of and the core of the advected vortex approaches the bottom of the computational domain instead of moving along the center line. Indeed, due to the coarse discretization a considerable deformation of the vortex was to be expected. However, in view of the previous test case “CD_NSA_2D” we’ve anticipated a much more accurate advection of the vortex core.

Therefore, we analyzed a variety of different discretizations with comprehensive variations of the underlying grid sizes and time step restrictions. Furthermore, we changed the boundary conditions to be completely periodic at all sides of the computational domain as in the “CD_NSA_2D” case. Nevertheless, we were not able to compute a solution of the advected vortex without a strong smearing of the circular vortex structure.

For comparison we also computed the same test case with the research code *MOLOCH*, see figure 19. This code is based on a comparable zero-Mach scheme of second order accuracy [13]. Although the numeric scheme of this code produces more numerical dissipation than FDS, it doesn't smear of the vortex structure as much as FDS. Especially the vortex core is advected almost along the center line.

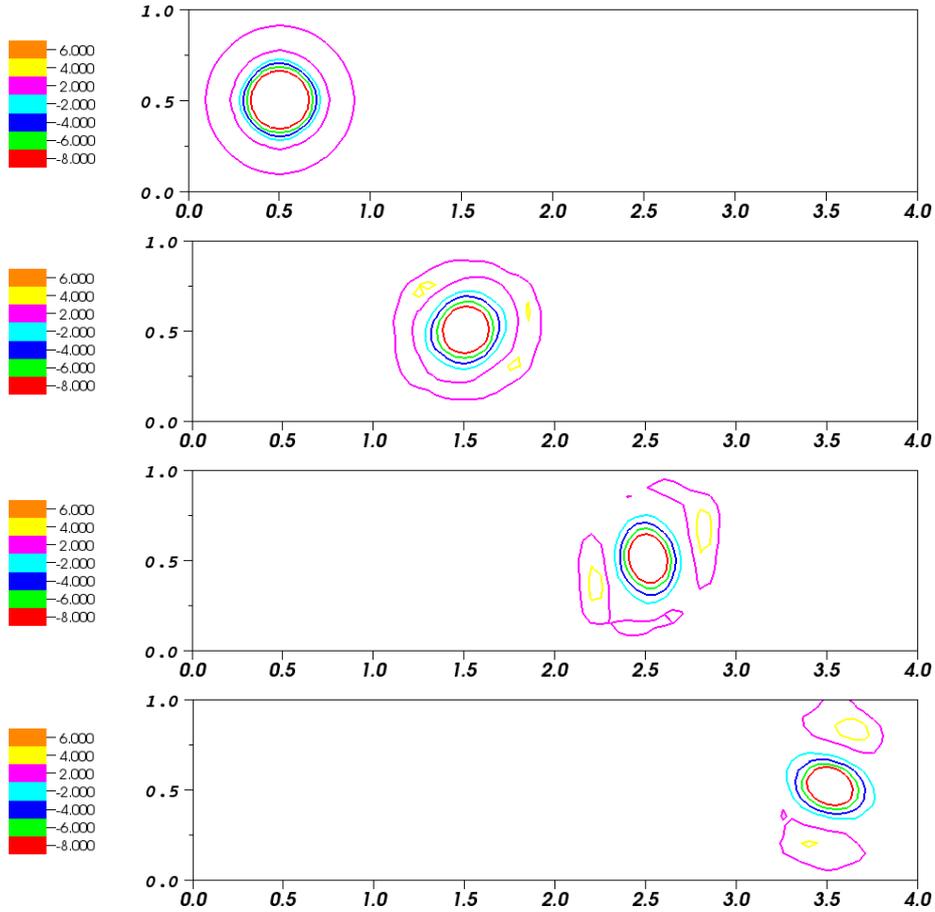


Figure 19: Isolines of the vorticity of an one-mesh $N = 80 \times 20$ MOLOCH computation at $t = 0, 1, 2$ and 3 with $CFL = 0.25$ for $[-8, -6, -4, -2, 2, 4, 6]$.

Because of the insufficient results in case of the one-mesh FDS-FFT computation, a deeper investigation of the differences between the FFT and SCARC-scheme for several multi-mesh constellations doesn't seem to be sensible at that time. Vortex advection is an important challenge for CFD programs for fire safety issues, therefore further investigations are necessary to analyze the reason for this behaviour.

4.3. PIPE_2D test example

The pipe test is an analytical test case to analyze the hydrodynamic solver in conjunction with nonviscous inflow, open boundary and free-slip wall conditions [14, 15]. At the left side a channel is impinged with an accelerated velocity flow $u(t)$. The right side of the channel is open. To test the accuracy of the domain decomposition method, we subdivide the computational domain in $M = 1$ to 8 subdomains as demonstrated in figure 20.

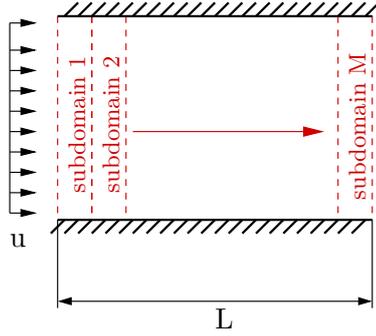


Figure 20: Multi-mesh pipe test

The gradient of the pressure drop Δp between the left and right side of the channel can be analytically determined from the momentum equation

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) + \nabla p = \mathcal{S}_{\rho \mathbf{u}}.$$

This is true because the flow velocity is spatially homogeneous, therefore the advection term of the momentum equation vanishes. Furthermore, the source term $\mathcal{S}_{\rho \mathbf{u}}$ in this example is zero. This leads to the equation

$$\rho \frac{\partial \mathbf{u}}{\partial t} = -\nabla p \approx -\frac{\Delta p}{\Delta x}. \quad (10)$$

To investigate the hydrodynamic solver of FDS we use two definitions for the velocity acceleration $u(t)$. With (10), $\Delta x = L$ and the defined $u(t)$ the mean pressure drop between in- and outflow is

1. for the linear acceleration $u(t) = u_0 t$

$$\Delta p_{exact} = -\rho u_0 L, \quad (11)$$

2. for the sinusoidal acceleration $u(t) = \sin(2\pi t)$

$$\Delta p_{exact} = -\rho u_0 2\pi \cos(2\pi t) L. \quad (12)$$

In this special test configuration the pressure difference Δp is identical with the hydrodynamic pressure $\Delta \tilde{p}$, used in FDS as described in part I of this series [7].

4.3.1. Test of the hydrodynamic solver

Starting with $M = 1$ we investigate the solution of an one-mesh computation with $u_0 = 1 \frac{m}{s}$, $\rho = 1.19882 \frac{kg}{m^3}$, and $L = 0.8 m$. Subsequently, p_{fds} represents the computed FDS-solution \tilde{p} and p_{exact} the exact solution. As shown in figure 21 the computed results of FDS-FFT reproduce the analytical solution of (11). However, there are two perceptible deviations between the pressure graphs for both accelerations.

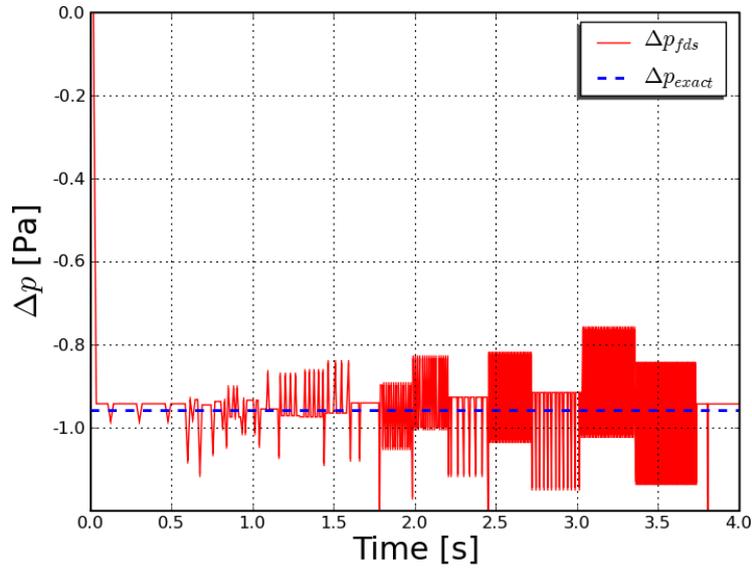


Figure 21: Result of an one-mesh FDS-FFT computation with $u(t) = t$.

First, there is a small difference between the mean values of p_{fds} and p_{exact} . The second observation consists in the small irregular peaks in the pressure graph. For the sinusoidal test case we get similar results. As shown in figure 22 the computed results of FDS-FFT reproduce the analytical solution of (12). However, there are oscillations at the minima and maxima of the graph, too.

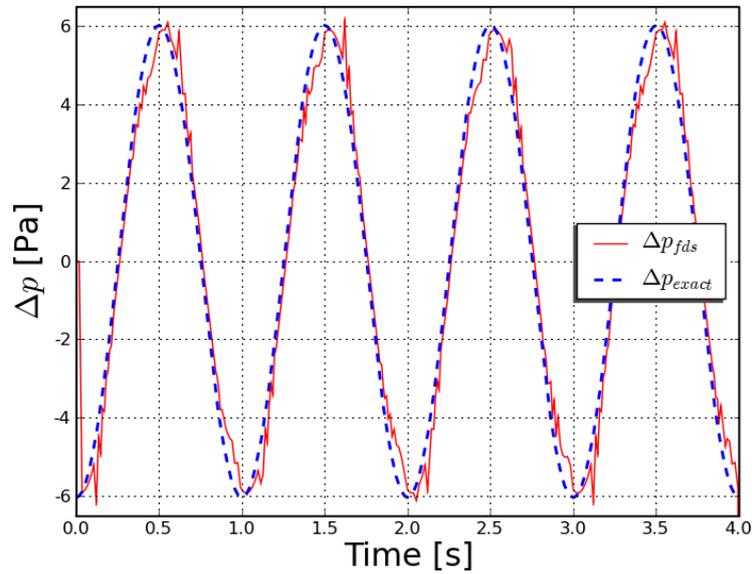


Figure 22: Result of an one-mesh FDS-FFT computation with $u(t) = \sin(2\pi t)$

To verify that these oscillations are not an inevitable consequence of the underlying zero-Mach scheme, we compare the result with the research code *MOLOCH*, see figure 23. This code is based on a comparable zero-Mach scheme of second order accuracy [13]. Obviously, this code matches the analytical solution very well and no differences or oscillations are observable at all. This indicates that the differences and oscillations are a result of deficiencies in the implementation of the numerical scheme in the FDS program.

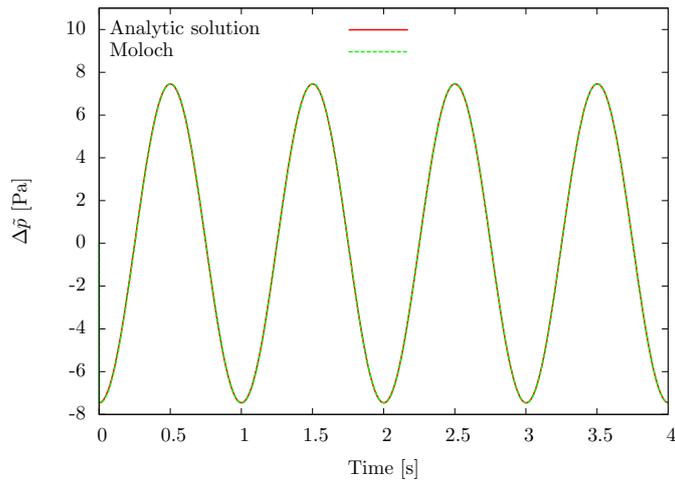


Figure 23: Result of a computation with the research code *MOLOCH* and $u(t) = \sin(2\pi t)$

Analyzing the deviations

Using our component-level strategy, we have carefully investigated all parts of the FDS code, which are involved in the computation of these results. In the underlying numerical scheme of the FDS solver the pressure drop between the left and right side of the channel guarantees the compliance of the divergence constraint (see part I [7] eq. (10)). In consideration of the analytical solution (11) the pressure drop depends on ρ , u_0 and L .

A first investigation verifies that the computed velocity field is a constant parallel flow ($\mathbf{u} = u(t), 0, 0$) for each single time step. Therefore the divergence constraint is fulfilled at each time step. By using the very simple acceleration definition $u(t) = t$ we detected, that FDS computes the data at the time $t - \Delta t$ instead of t . Therefore the divergence constraint is fulfilled at the wrong computational time.

Furthermore the output of the hydrodynamic pressure is related to the cell center of the grid cell adjacent to the domain boundary even though the correct geometrical position of the boundary face is defined in the output line of the FDS file. Therefore there is an offset of half of the gridwidth $0.5 \Delta x$ between the location of the cell-centered pressure data and the cell face boundary. As a consequence the numerical distance between in- and outflow is $L - \Delta x$.

Recapitulatory, FDS can't print out the pressure difference between the real in- and outflow boundary faces, but only slightly shifted. For a numerically correct comparison of the computed FDS solution we therefore have to define a "numeric" pressure difference instead of the exact pressure difference Δp_{exact} which explicitly uses the adjacent cell-center values instead of the real boundary values.

$$\Delta p_{numeric} = -\rho u_0 2\pi \cos(2\pi t)(L - \Delta x). \quad (13)$$

If we take these limitations into account, the scheme in the corrected FDS program works very well. The detected errors are being discussed with the FDS developers and will be corrected in the next future. Furthermore, a similar test case will be used as standard test in the FDS verification guide.

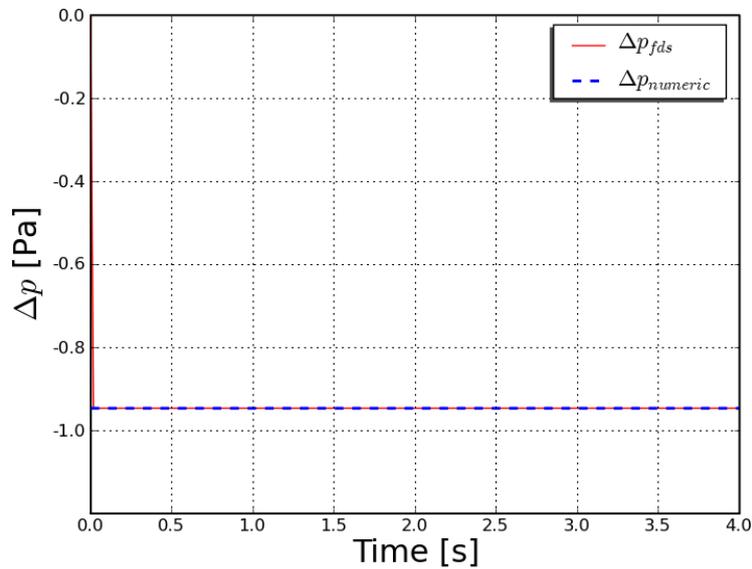


Figure 24: Results of a corrected one-mesh FDS-FFT computation with $u(t) = t$ without any differences or oscillations.

The new FDS-SCARC scheme can be used as an alternative of the FDS-FFT scheme. It must be noted that this new scheme replaces only the pressure solver in FDS. All other surrounding components of the FDS program are not affected by this replacement. For this reason the FDS-FFT and FDS-SCARC schemes produce the same results in case of a one-mesh computation. The correctness for the single mesh case is demonstrated in figure 25.

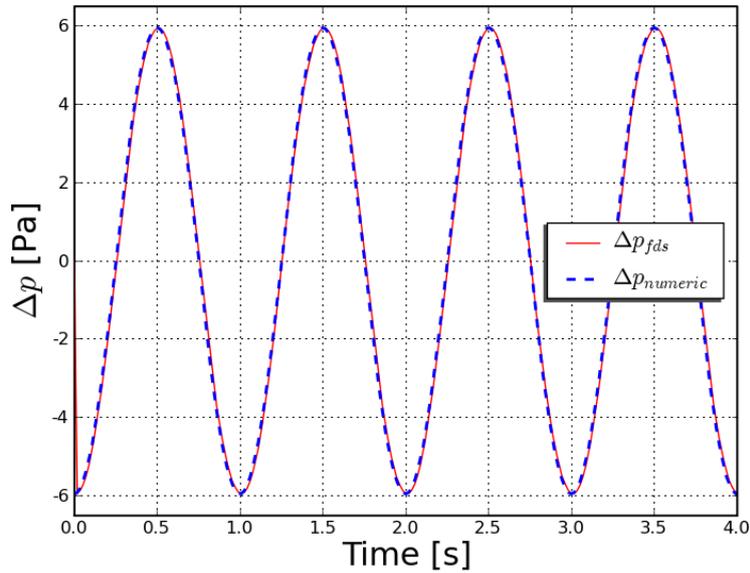


Figure 25: Results of a corrected one-mesh FDS-SCARC computation with $u(t) = \sin(2\pi t)$

The solution presented in figure 24 and 25 already relies on a temporally correct evaluation of the inflow data and is compared to $p_{numeric}$ instead of p_{exact} .

Figure 25, which is the same for FDS-FFT and FDS-SCARC, indicates an absolute conformance in the one mesh case. But if we zoom in one maximum of figure 25, there are still small deviations between the numerically exact p_{exact} and the computed solution p_{fds} . To quantify this error a convergence study is indispensable.

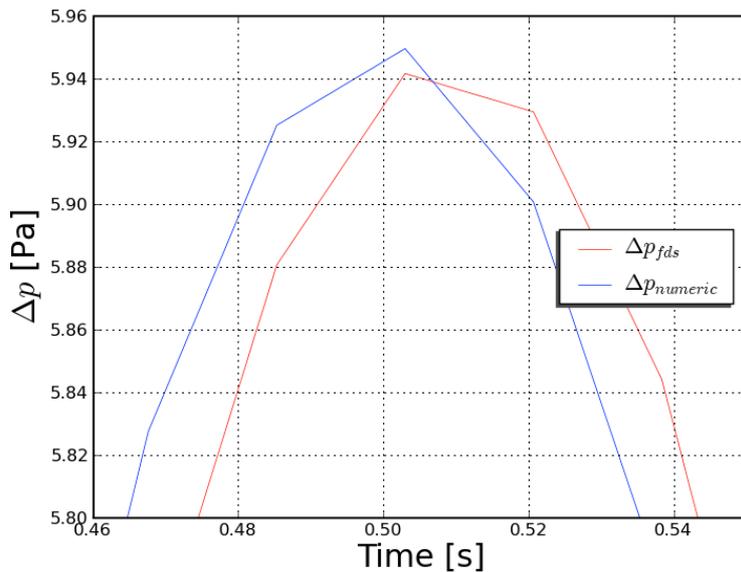


Figure 26: Zoom into one maximum of the one-mesh computation of FDS-FFT for $u(t) = \sin(2\pi t)$

To be precise, the small irregular peaks in the pressure graph of figure 21 and 22 are caused by a slightly coding error in FDS which will be resolved in future versions of the code. The small pressure difference between the computed and analytical solution is caused by a limitation of the underlying discretization scheme. This is not an error, but rather a program limitation. Users of FDS should take this limitation into consideration.

4.3.2. Test of the domain decomposition

To test the gasdynamic solver in conjunction with the domain decomposition method, the domain is split up from $M = 1$ to 8 subdomains. For these computations the described error in the source code of FDS 5.4.3 revision 5210 was already fixed in our version. Furthermore we have compared the pressure drop against the numerical correct solution (13) instead of the analytical exact solution (12).

However, as shown in figure 27 the computed results of the FDS program become erroneous if the computational domain is divided into single subdomains. Although this case is a simple parallel flow with constant density, FDS-FFT is not able to compute the correct results in this case.

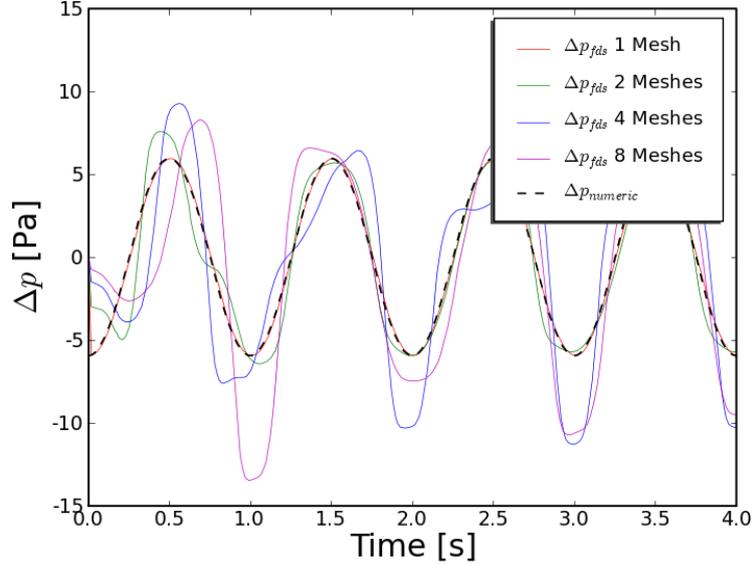


Figure 27: Results of corrected FDS-FFT computation with $M = 1, 2, 4$ and 8 subdomains and $u(t) = \sin(2\pi t)$

Taking the differences between the numerically correct solution (13) and the results for different M-mesh computations gives unacceptable errors up to 7.65 Pa, as illustrated in figure 28.

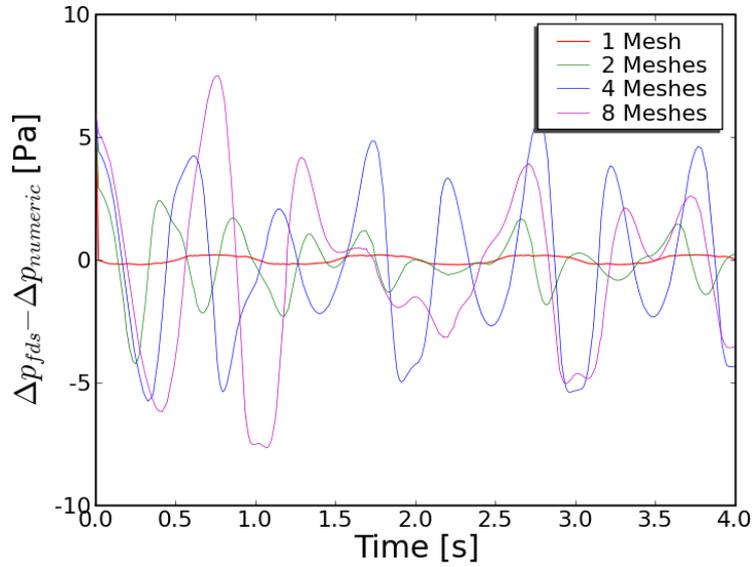


Figure 28: Error between numerically correct solution and computed pressure drop of a corrected FDS-FFT computation with $M = 1, 2, 4$ and 8 subdomains and $u(t) = \sin(2\pi t)$

However, in comparison with these insufficient FDS-FFT results, the new FDS-SCARC technique demonstrates the advantage of a numerical scheme following the mathematical characteristics of the underlying set of equations.

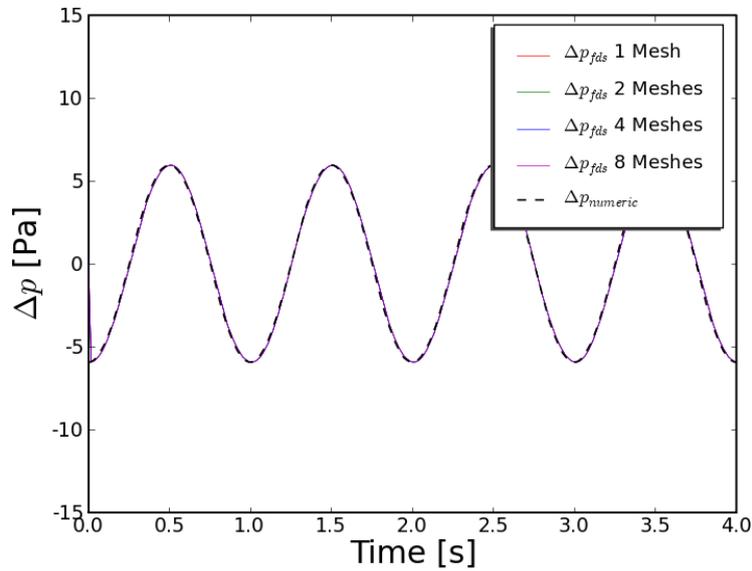


Figure 29: Result of corrected FDS-SCARC computation with $M = 1, 2, 4$ and 8 subdomains and $u(t) = \sin(2\pi t)$

As figure 29 demonstrates, the computed solution is independent from the amount of subdomains or meshes, which is a major prerequisite for the parallelization of numerical computations.

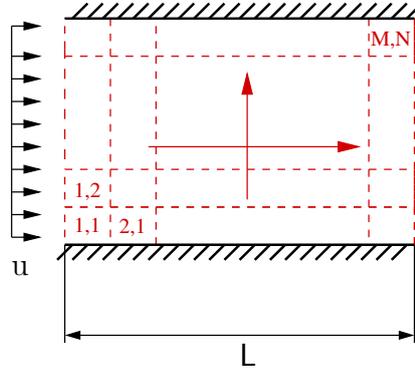


Figure 30: Extended domain decomposition in x- and y-direction

Furthermore, if we split the computational domain in M submeshes in x-direction and y-direction as shown in figure 30, the errors of the FDS-FFT scheme increase, whereas the FDS-SCARC scheme is not affected by the domain decomposition, no matter how many submeshes are used. This fact is demonstrated in figure 31 for the FDS-FFT and in figure 32 for the FDS-SCARC technique.

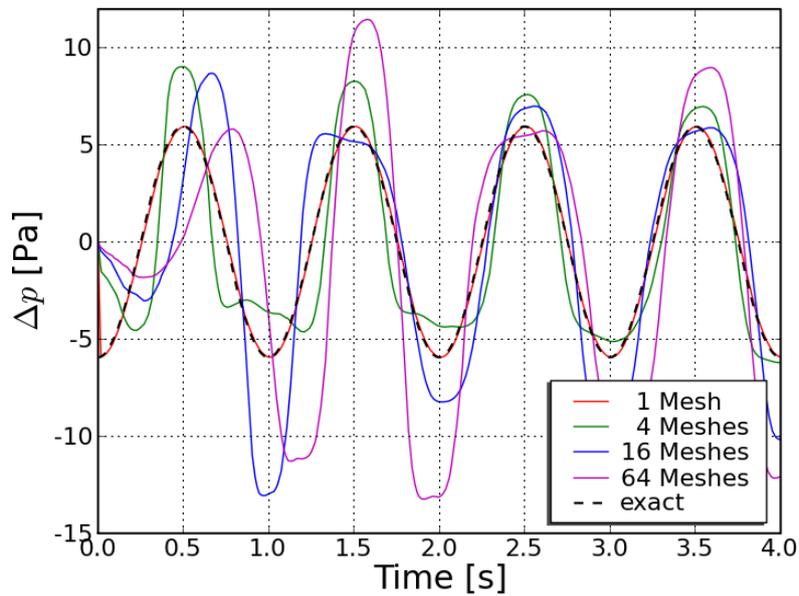


Figure 31: Results of a corrected FDS-FFT computation with subdomains in x- and y-direction and $u(t) = \sin(2\pi t)$

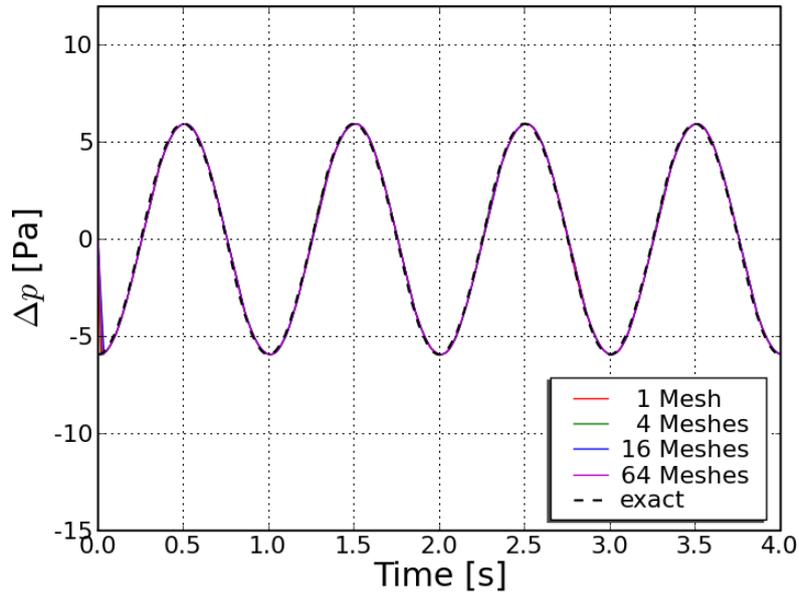


Figure 32: Results of a corrected FDS-SCARC computation with subdomains in x- and y-direction and $u(t) = \sin(2\pi t)$

4.3.3. Convergence study

Obviously, the upper figures demonstrate basic problems of the current FDS-FFT scheme with respect to multi-mesh computations. Although the new FDS-SCARC scheme seems to provide much more correct results, only a convergence study can prove its real convergence order.

For several FDS-FFT and FDS-SCARC computations with respect to different $M \times 1$ -subdivisions, figures 33 and 34 show the resulting convergence orders for the velocity in x-direction with respect to the euclidean L2-norm.

As expected after the previous discussion, FDS-FFT suffers from deteriorations of its convergence order which isn't even of first order any more. In contrast to that, the results for the FDS-SCARC scheme are independent of the number of submeshes and show its big potential for improvement. Here, the L2-error lies in the range of machine accuracy which indeed can be expected from a modern numerical scheme for a simple velocity progression like this.

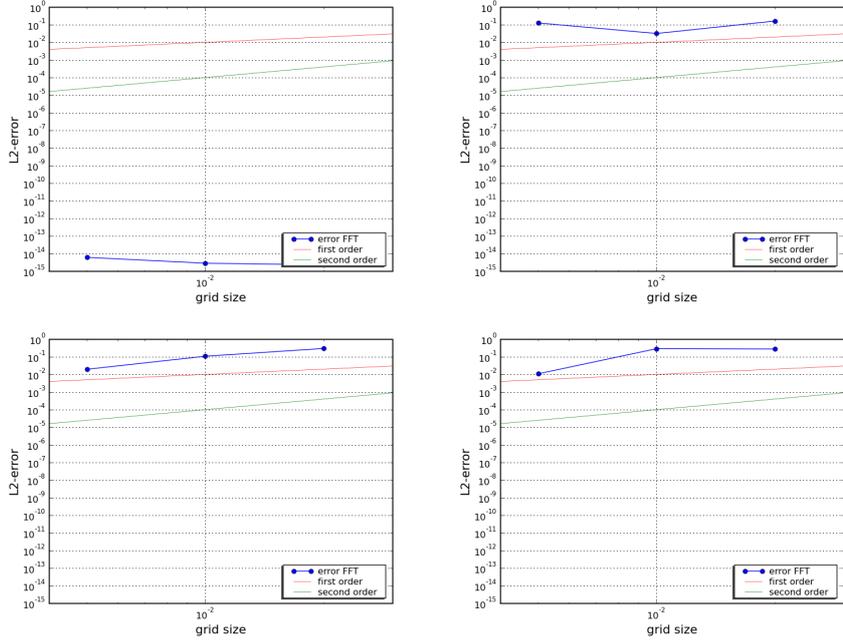


Figure 33: L2-errors of the velocity in x-direction for an FDS-FFT computation using the sinusoidal acceleration with $M = 1$ (top left), $M = 2$ (top right), $M = 4$ (bottom left) and $M = 8$ (bottom right) meshes.

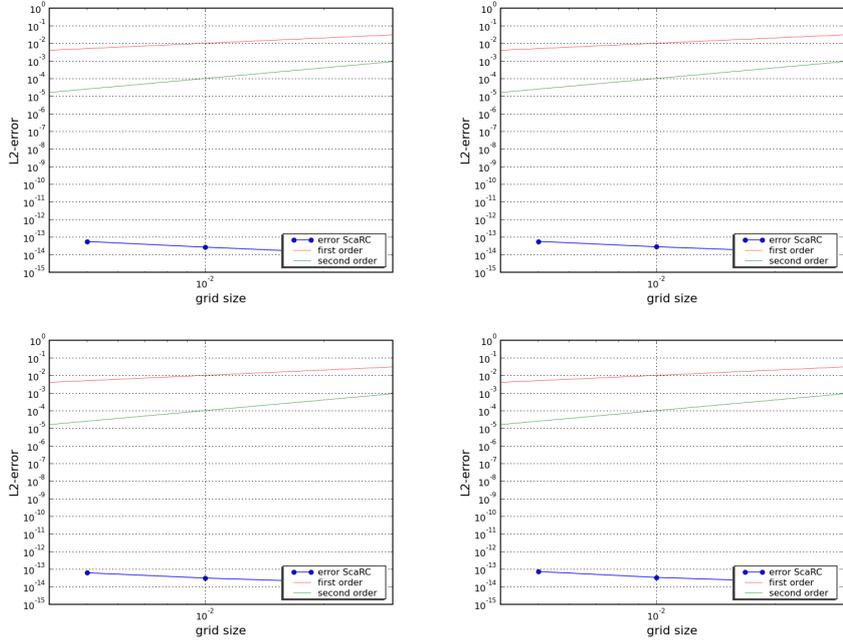


Figure 34: L2-errors of the velocity in x-direction for an FDS-SCARC computation using the sinusoidal acceleration with $M = 1$ (top left), $M = 2$ (top right), $M = 4$ (bottom left) and $M = 8$ (bottom right) meshes.

For a more detailed investigation of the properties of FDS-SCARC we consider the more complicated solution of the hydrodynamic pressure drop, used in the previous discussion. Whereas the L2-error of the FDS-FFT computation is far away from first order accuracy (see figure 35).

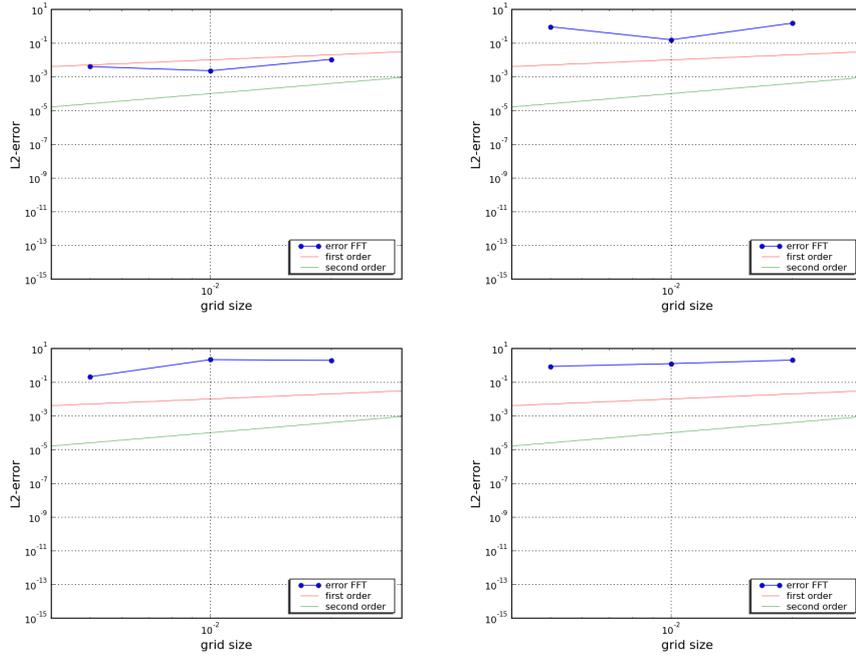
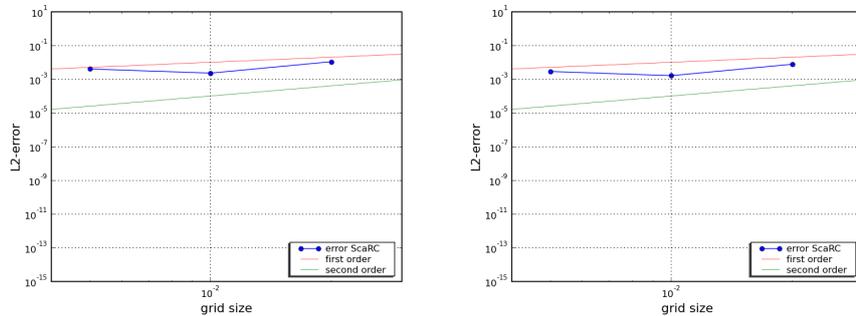


Figure 35: L2-errors of the pressure drop for an FDS-FFT computation using the sinusoidal acceleration with $M = 1$ (top left), $M = 2$ (top right), $M = 4$ (bottom left) and $M = 8$ (bottom right) meshes.

The L2-error of the new FDS-SCARC scheme is located between first and second order accuracy, depending on the kind of subdivision (see figure 36).



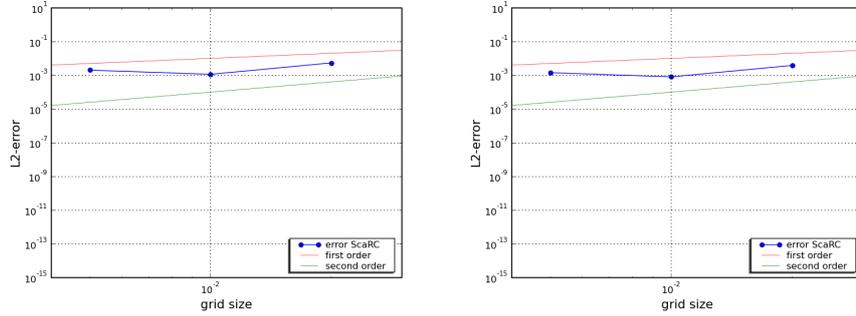


Figure 36: L2-errors of the pressure drop for an FDS-SCARC computation using the sinusoidal acceleration with $M = 1$ (top left), $M = 2$ (top right), $M = 4$ (bottom left) and $M = 8$ (bottom right) meshes.

Nevertheless the pressure drop in figure 35 and 36 doesn't consider all discrete values of the computational domain, because the drop is only computed by the difference between the left and right boundary values over the pipe width.

To consider all values of the computational domain we have computed a discrete pressure derivative by simply taking local difference quotients in x-direction at neighboring pressure nodes. We can not use the pressure value itself, because the solution of the underlying elliptic equation involves an unknown constant, which can differ between the meshes.

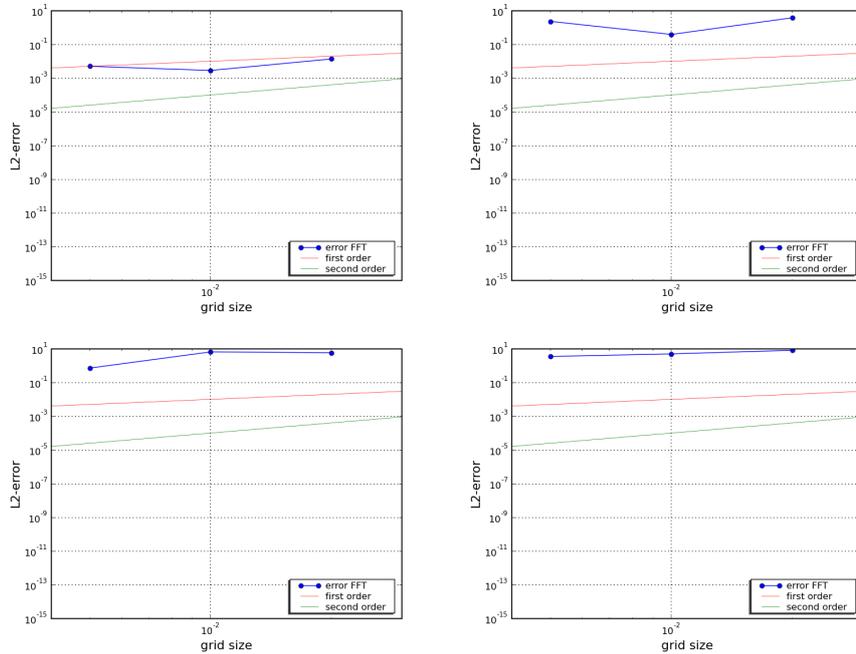


Figure 37: L2-errors of the pressure derivative for an FDS-FFT computation with $M = 1$ (top left), $M = 2$ (top right), $M = 4$ (bottom left) and $M = 8$ (bottom right) meshes.

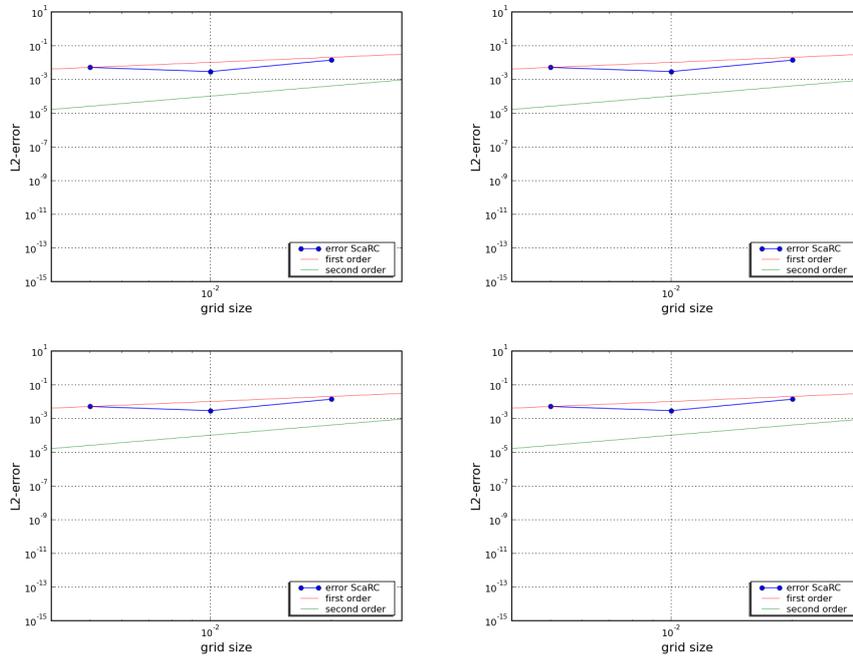


Figure 38: L2-errors of the pressure derivative for an FDS-SCARC computation with $M = 1$ (top left), $M = 2$ (top right), $M = 4$ (bottom left) and $M = 8$ (bottom right) meshes.

Although the results of the FDS-SCARC scheme are unquestionably better than that of FDS-FFT, the resulting slopes give rise to further investigations because they don't indicate a pure second order convergence. Even in the case of the single-mesh computation which is not affected by domain decomposition issues and is independent of the choice of the pressure solver, there are degradations in case of the finest grid width. This may be caused by some open issues in other FDS components. It should be noted that this type of errors can only be detected with such comprehensive numerically oriented test cases.

4.4. Flow around body test example

Whereas the previous test cases focused on the properties of the hydrodynamic solver isolated from many other components of FDS, the next test case is more like a numerical experiment. Although we have no experimental data for this test case and there is no analytical solution available, we use this type of test to demonstrate the importance of the new FDS-SCARC scheme even for more practical situations.

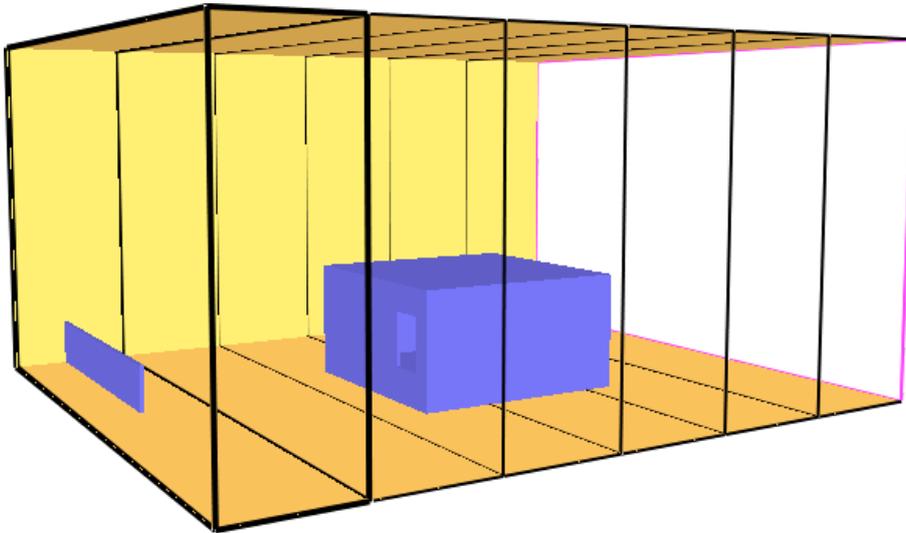


Figure 39: Flow around body test case

As shown in figure 39 we compute the flow around a small body inside a large room. At the left side of the large room an inflow with 1 m/s through a small opening is forced. The right side of the room is an open vent. The computational domain consists of $180 \times 155 \times 90$ cells with a uniform grid width of 10 cm and is subdivided into 6 meshes in x-direction. The computation uses the default settings of FDS including the LES-solver, only the radiation is switched off.

Figure 40 shows the slice plot of the velocity of a FDS-SCARC computation and figure 41 the results for the FDS-FFT computation. Both plots represent the same computational time and use the same color scaling. A simple optical comparison of both plots demonstrates the deficiencies of the FDS-FFT computation with respect to domain decomposition. Only the FDS-SCARC solution is independent from the way of decomposing the computational domain. The numerical consequences have already been discussed in detail in the context of the previous test cases. Therefore we refrain from a more detailed numerical investigation here.

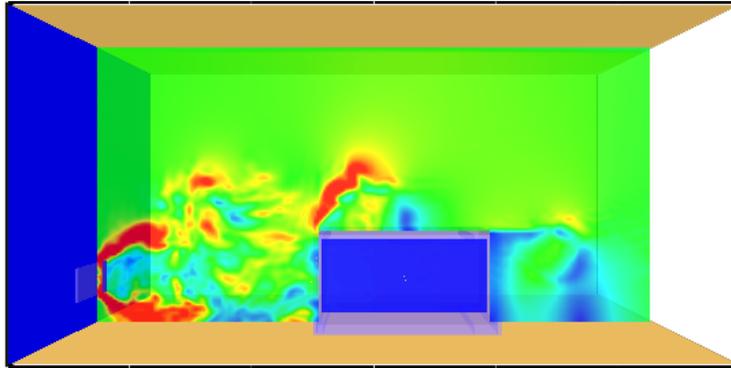


Figure 40: FDS-SCARC: Slice-Plot of the velocity

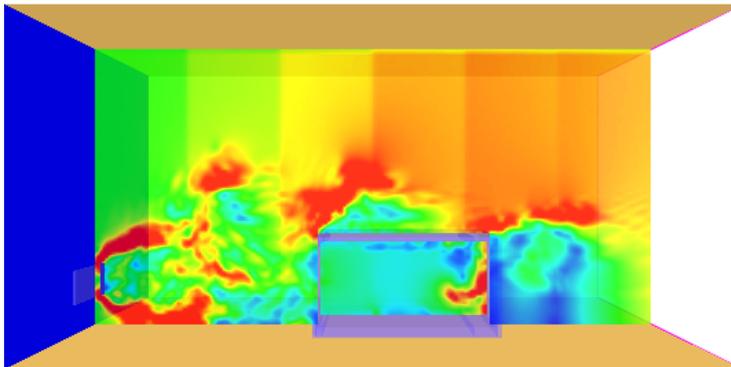


Figure 41: FDS-FFT: Slice-Plot of the velocity

The test case clearly illustrates the stagnant information transfer between different meshes in the FDS-FFT computation. As already explained in part I [7], the efficient parallelisation of an elliptic partial differential equation as the pressure equation in FDS is a particular challenge in many simulation programs for fire-induced flows. This type of equation possesses a very specific character, namely an infinite rate of propagation for information. Local information is spread extremely fast over the whole domain, regional perturbations instantaneously impact the solution on all parts of the domain, resulting in a very strong overall coupling of data. The correct solution of the pressure equation has a direct influence to the fulfillment of the divergence constraint which is extremely important for the progress of the whole method. In case that the pressure equation isn't solved appropriately, the divergence constraint may remain violated.

4.5. Discussion of the test cases

In this context, the open multi-mesh issues being discussed in the underlying article can be led back to the very different treatment of the information transfer in the SCARC and FFT-schemes. Due to its local character, the FFT-scheme

is only able to exchange data between directly neighboring meshes during one time step. To pass an information from one end of the domain to the other a more or less comprehensive sequence of staggered exchange cycles between neighboring meshes is needed. Depending on the number of subdomains, this process may take many time steps and cause a very unphysical slow-down of the propagation velocity which stands in conflict to the real physical behavior. This fact is very well illustrated in figure 41 of the “Flow around body test example”. In contrast to that, SCARC is based on global transfer mechanisms which are able to spread information all over the domain within only one time step and therefore much better map the physical properties of the pressure.

In the “CD_NSA_2D” and “CD_VA_2D” cases, physical changes in the considered quantities are transported very slowly through the domain. The velocity field at the opposite end of the domain only changes with significant temporal delay. Therefore, the consequences of the insufficient global information transport in case of the FFT scheme are only small for these examples. However, for the “PIPE_2D” case an instantaneous information transfer through all subdomains is needed. Due to the divergence constraint the accelerated inflow at the left side of the domain requires an immediate adaption of the velocities in the whole flow area which has to be completed in the course of only one time step, see part I [7]. This cannot be realized by a step-by-step sequence of local data exchanges but only by a suitable global data transfer. Therefore, this example illustrates the insufficient consideration of the elliptic pressure character by the FFT-scheme very clearly. A comparable effect can be seen in case of a local heat source, because the divergence constraint impacts the velocity field up to the boundaries of the whole domain (see e.g. [16]).

5. Summary and Outlook

The interdisciplinary character of Scientific Computing requires the sophisticated interaction of different scientific fields: After a suitable modelling by the corresponding applied sciences as for example physics or chemistry, the essential task of mathematics consists in the design and verification of efficient solver methods on the base of the most modern numerical methodology. With respect to the current hard- and software developments these methods must be transferred to efficient algorithms and ported to different computer architectures, which is mostly an issue of computer sciences. At last, the simulation results must be validated by comparing them with experimental reference values, which again requires the cooperation with the associated applied sciences. Only a perfectly concerted approach leads to a meaningful simulation which is able to replace expensive experiments and provides a reliable evaluation of the considered phenomenon.

In the present article we have illustrated that the widely used comparisons with fire experimental data are not sufficient to check the quality of CFD programs. Beyond doubt, they are necessary and useful but the demonstrated results highlight the necessity of a more comprehensive testing strategy which has to include investigations of numerical quality criteria (convergence, stability and order) and component-level tests. Our test examples show the advantages of analytical and numerical component-level tests.

Obviously, FDS-SCARC produces much better results than FDS-FFT in the case of multi-mesh computations. As described before, the solution of FDS-SCARC is independent with respect to the number of subdomains M , whereas FDS-FFT may lead to increasing errors when the number of meshes is increased. A convergence analysis supports these results. In the multi-mesh case FDS-FFT does no longer guarantee second order accuracy. The reasons are described in [7]. Therefore computations without domain decomposition on single meshes are not affected by this problem. Also it should be noted that the SCARC technique cannot produce better results than the one-mesh FFT version, because it only replaces the FFT pressure solver but not the surrounding parts of the code. Therefore irregularities in an one-mesh computation can not be fixed by using the SCARC technique, but can be identified by a more comprehensive testing strategy as we have demonstrated here. Furthermore, the different results depending on the CFL-settings in the advected vortices test case indicate that there are most probably other open issues.

Until today, the Fire Dynamics Simulator is based on the FFT-solver scheme, which obviously isn't reliable in the multi-mesh case. The exclusive focus on computational costs, the main motivation to use the FFT-scheme, affects the correctness of the underlying numerical scheme. Even if users comply with the instructions related to the proper definition of subgrids given in the User Guide, they cannot expect correct results of second order accuracy in the case of multi-mesh computations. This was impressively demonstrated by the upper pipe-test mentioned above with its very simple parallel flow character.

The consequences of the detected problems in the case of multi-mesh computations in the large area of fire safety applications can not be estimated by the authors. However, authorities and fire safety engineers would be advised to be aware of the current multi-mesh deficiencies. Correct computations are not inconsistent with fast computations. But fast and faulty computations are questionable.

Finally, the authors express a special thanks to all persons who were/are involved in the development of FDS, especially Kevin McGrattan und Randy McDermott, with regard to their high degree of commitment and very successful work. Even if there are many software tools freely available in the internet, the open-source supply of such an extensive and sophisticated CFD-program is far from being a matter of course. Furthermore, the authors don't know any other CFD-program with such a fast and engaged support as the FDS-team continuously provides. Therefore, we are highly interested to make a positive contribution to bring this great work forward. This also includes the basic discussion of open issues and possibilities for improvement with the FDS developers as we already experienced for many times.

Acknowledgements:

The support of and the critical discussions with Prof. Rupert Klein at the Freie Universität Berlin with respect to the presented testing strategy are gratefully acknowledged.

References

- [1] *AIAA Guide for the Verification and Validation of Computational Fluid Dynamics Simulations (G-077-1998e)*. American Institute of Aeronautics and Astronautics, AIAA Standards Series, 1998.
- [2] **Almgren, A.; Bell, J.; Colella, P.; Howell, L.; Welcome, M.:** *A Conservative Adaptive Projection Method for the Variable Density Incompressible Navier-Stokes Equations*. *J. Comp. Phys.*, volume 142:pages 1–46, 1998.
- [3] **Brätz, A.:** *Zur Druckberechnung in der parallelen Version des NIST Fire Dynamics Simulators v5.0*. In M. Bolten (editor), *Beiträge zum Wissenschaftlichen Rechnen. Ergebnisse des Gaststudentenprogramms 2007 des John von Neumann-Instituts für Computing*, number FZJ-JSC-IB-2007-12, pages 23–31. Forschungszentrum Jülich GmbH, Dezember 2007.
- [4] **Gresho, P. M.; Chan, S. T.:** *On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix. Part 2: Implementation.. International Journal for Numerical Methods in Fluids*, volume 11:pages 621–659, 1990.
- [5] **Janicka, J.; Turan, A.; Sadiki, A.; Dreizler, A.; Klein, M.; Knop, P.** (editors): *First Workshop on Quality Assessment of Unsteady Methods for Turbulent Combustion Prediction and Validation*, number 1. Technische Universität Darmstadt, FG Energie- und Kraftwerkstechnik, June 2005.
- [6] **Kilian, S.:** *Iterative Gebietszerlegungskonzepte für den FDS-Drucklöser zur Stärkung der globalen Kopplung*. 2. FDS-Usergroup Workshop, hhp-berlin, Berlin, 4.-5. Dezember 2008.
- [7] **Kilian, S.; Münch, M.:** *A new generalized domain decomposition strategy for the efficient parallel solution of the FDS-pressure equation, Part I: Theory, Concept and Implementation*. Technical report ZR-09-19, Konrad-Zuse-Zentrum für Informationstechnik Berlin, June 2009. ISSN 1438-0064.
- [8] **Kleb, B.; Wood, B.:** *CFD: A Castle in the Sand?*. In *34th AIAA Fluid Dynamics Conference*, CFD Verification and Validation session 88-FD-22. AIAA, June 28 - July 1 2004.
- [9] **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.
- [10] **Minion, M. L.:** *A Projection Method for Locally Refined Grids*. *Journal of Computational Physics*, volume 127:pages 158–178, 1996.
- [11] **Münch, M.:** *Verifikation und Validierung des CFD-Programms MOLOCH*. Freie Universität Berlin, Fachbereich Mathematik & Informatik, Scientific Computing, Arnimallee 6, 14195 Berlin, 1.1 edition, April 2007.

- [12] **Münch, M.:** *Ist eine Gebietszerlegung in mehrere Gitter bei FDS-Simulationen zulässig?*. 2. FDS-Usergroup Workshop, hhpberlin, Berlin, 4.-5. Dezember 2008.
- [13] **Münch, M.:** *MOLOCH – Ein Strömungsverfahren für inkompressible Strömungen - Technische Referenz 1.0*. Technical report PIK-Report Nr. 109, Potsdamer Institut für Klimafolgenforschung e.V. Januar 2008.
- [14] **Münch, M.; Gebauer, J.; Klein, R.:** *Verification and Validation of CFD Models for Smoke Spreading Predictions*. Poster session at 9th International Symposium on Fire Safety Science, Karlsruhe, September 2008.
- [15] **Münch, M.; Klein, R.:** *Anforderungen an numerische Berechnungen der Brand- und Rauchausbreitung im Vorbeugenden Brandschutz*. *vfdB-Zeitschrift*, volume 3:pages 145 – 151, August 2008.
- [16] **Münch, M.; Klein, R.:** *Critical numerical aspects for field model applications*. *EUSAS-Journal*, volume 4:pages 41–54, July 2008.
- [17] **Oberkampf, W. L.; Trucano, T. G.:** *Verification and Validation in Computational Fluid Dynamics*. Technical report, Sandia National Laboratories, P.O. Box 5800, Albuquerque, New Mexico 87185, March 2002.
- [18] **Rogsch, C.:** *Parallelrechnung in der Praxis - Ist FDS5 wirklich besser als FDS4?*. 1. FDS-Usergroup Workshop, hhpberlin, Berlin, 6.-7. März 2008.
- [19] **Schlesinger, S.:** *Terminology for Model Credibility*. *Simulation*, volume 32(3):pages 103–104, 1979.
- [20] **Schneider, T.; Botta, N.; Geratz, K.; Klein, R.:** *Extension of Finite Volume Compressible Flow Solvers to Multi-dimensional, Variable Density Zero Mach Number Flow*. *Journal of Computational Physics*, volume 155(2):pages 248–286, 1999.
- [21] **Schwarz, H. R.:** *Numerische Mathematik*. Teubner, Stuttgart, 4. edition, 1997. ISBN 3-519-32960-3.
- [22] **Vater, S.:** *A New Projection Method for the Zero Froude Number Shallow Water Equations*. PIK Report 97, Potsdam Institute for Climate Impact Research, 2005.