PROCEEDINGS, Fire and Evacuation Modeling Technical Conference (FEMTC) 2014
Gaithersburg, Maryland, September 8-10, 2014

NUMERICAL INSIGHTS INTO THE SOLUTION OF THE FDS PRESSURE EQUATION: SCALABILITY AND ACCURACY

Dr. Susanne Kilian
hhpberlin - Ingenieure für Brandschutz GmbH
Rotherstr. 19
10245 Berlin
e-mail: s.kilian@hhpberlin.de

ABSTRACT

The scalability of the *Fire Dynamics Simulator* (FDS) with respect to massively parallel computers largely relies on the efficient solution of the pressure equation which is closely coupled with the computation of all other thermodynamic quantities. The current FDS pressure solver is based on the meshwise use of direct *Fast Fourier Transformations* (FFT) which are iteratively coupled with an additional averaging process along internal mesh boundaries. This purely local way of proceeding can cause difficulties concerning the reliable approximation of global data dependencies encompassing the whole computational domain which in turn may impair the effective use of multi-mesh computations whilst retaining a high level of accuracy at the same time.

As an alternative approach the solver package *Scalable Recursive Clustering* (ScaRC) is under development consisting of a selection of different iterative solution techniques of domain decomposition and multigrid type which belong to the most efficient and robust solvers for huge systems of equations. Based on elaborated numerical combinations of local and global mechanisms for capturing the overall physical effects, ScaRC has already proven considerably enhanced scalability and accuracy properties for a variety of different test and verification cases. This article is intended to give an overview of the underlying concepts as well as the current state of development accompanied by some illustrative numerical test examples and comparisons with the current FFT-solver.

INTRODUCTORY OVERVIEW

The computer-based simulation of realistic fire scenarios on modern multicore architectures places highest demands on the efficiency of the underlying numerical solution procedures. For the optimal utilization of the current technologies it is indispensable to sophisticatedly transfer formerly approved methods - which were usually designed for considerably smaller problem constellations on single processor systems - to the new architectural features.

The most widely used methods are based on *domain decomposition* techniques: The considered computational domain is subdivided into smaller subdomains which are treated more or less independently on the different processors of a parallel computer and are coupled together by closely coordinated data exchanges among each other. Special care must be taken to ensure that this artificial subdivision doesn't impair the immanent global connectivity of the underlying problem. In particular, it must be guaranteed that the parallelization process preserves the convergence order and approximation quality of the original serial methodology to the best possible extend.

Accordingly, the world-wide used computational fluid dynamics program FDS which simulates the propagation of fire and smoke was originally designed for single-mesh constellations on single-processor systems and includes highly optimized serial components. Motivated by the rapid development in the area of high performance computers, FDS meanwhile also offers the possibility to be executed in parallel on many processors. This multicore application may reduce the computing time for a given constellation and enlarge the class of computable problems to a great extent.

Based on a subdivision into cubic subdomains with rectangular grid cells inside, domains with irregular geometric structures may now be decomposed into regular subdomains. The assignment of the subdomains to the single processors as well as the coupling among each other is done by means of the communication library *Message Passing Interface* (MPI) which is the present standard.

At first sight, FDS has a high potential for parallelization since it is mainly based on the application of an explicit time stepping method which is responsible for the simulation of all the thermodynamic quantities under consideration. While performing a new time step, methods like that only utilise values on neighboring grid cells from the previous time step which have already been computed before. This suggests that only local data exchanges between directly neighboring submeshes are required which can be performed with high computational efficiency on modern parallel architectures. Unfortunately, the situation reveals to be more complicated as will be explained below.

THE FDS PRESSURE EQUATION

The reliable parallelization of FDS is largely determined by the correct approximation of the pressure equation

$$\nabla^2 \mathcal{H} = -\frac{\partial(\nabla \cdot \mathbf{u})}{\partial t} - \nabla \cdot \mathbf{F} \qquad (1)$$

which has to be solved several times in every single step of the time stepping scheme, see [1].

As the vector $\mathbf{F}$ contains a collection of momentum flux terms from the previous time step, the computation of the pressure term $\mathcal{H}$ is closely concatenated with the computation of many other important quantities in FDS.

From a mathematical point of view, equation (1) is an elliptic partial differential equation of *Poisson*-type. One fundamental characteristic of elliptic equations consists of an extremely fast propagation velocity for information which implies a very strong global connectivity.

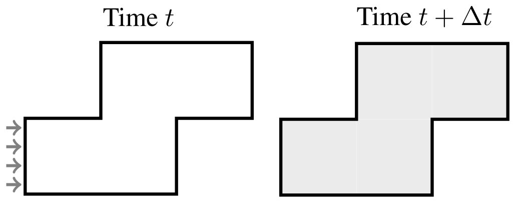


Figure 1: Fast propagation velocity for information

Only a single time step may suffice to spread a new information over the whole computational domain as illustrated in figure 1 for a simple two-dimensional pipe-shaped domain. Local effects have immediate impact on the overall solution.

Current FFT pressure solver

For the discretisation of equation (1) FDS relies on a cell-centered finite difference method of second-order accuracy. In the 2D-case the usual 5-point Laplacian stencil

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

is used. Based on a Taylor series expansion for the second order derivative, this stencil describes the dependencies of the single grid cells among each other as illustrated in the enlarged section in figure 2. The generalization to the well-known 7-point stencil for the 3D-case is straightforward, see for example [4].

In case of a single-mesh constellation this discretization process transforms equation (1) into a corresponding system of equations

$$Ax = b \qquad (2)$$

with one global matrix A and vectors x and b.

In case of a multi-mesh constellation with M submeshes a very natural parallelization strategy is used: Instead of one global system of equations, each subdomain holds its own system of equations

$$A_i x_i = b_i\,, \quad i = 1, \ldots, M \qquad (3)$$

with local matrices A_i and vectors x_i and b_i. Subsequently, the locally computed solutions x_i are coupled together by data exchanges between neighboring meshes.

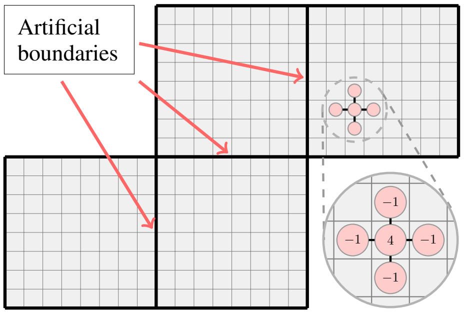


Figure 2: Discretization of a 2D pipe geometry with 5-point matrix stencil

As illustrated in figure 2, the mathematical solvability of the local subdomain problems requires the definition of new artificial boundary conditions along the internal subdomain boundaries which don't have a physical equivalent.

The current FDS pressure solver for the solution of the resulting systems of equations is based on highly optimized FFT-techniques taken from the solver package CRAYFISHPACK.FFT-methods belong to the class of so-called *direct* methods which compute the exact solution of a linear system of equations (up to machine precision) in one computational cycle which may be very complex.

In the single-mesh case only one globally defined FFT-method is performed which has proven to be extremely efficient and robust over the past years. Figure 3 illustrates the situation for the already known pipe example. Since FDS is only able to scope with rectangular subdomains, the computational domain has to be enlarged correspondingly.

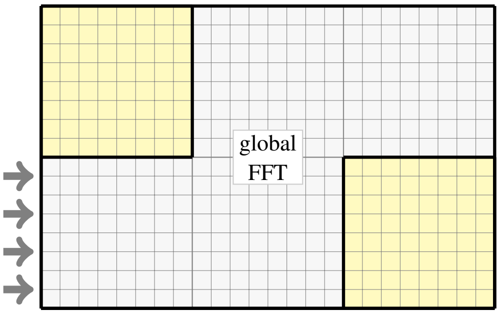


Figure 3: Single-mesh discretization with global FFT-method for the 2D pipe geometry

In order to avoid this additional need of storage space and computational time, it would be preferable to subdivide the domain into more regular subdomains which is done in the multi-mesh case as shown in figure 4. There, each subdomain performs its own, local FFT method accompanied by the subsequent coupling of the local solutions to a global one.

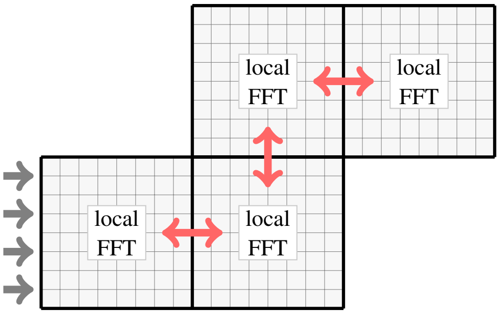


Figure 4: Multi-mesh discretization with local FFT-methods for the 2D pipe geometry

This purely locally oriented approach possesses an extremely high *parallel efficiency* because it only uses computationally cheap data exchanges between meshes which are direct neighbors.

However, the excellent efficiency of direct methods mainly rests on highly recursive and domain-spanning data dependencies. With a special view to the very high propagation velocity of the pressure equation it soon comes clear that the rather unphysical breakup of the global connectivity induced by the subdivision process cannot be sufficiently compensated by this very local way of proceeding.

An information exchange from one domain part to the other one can only take place in a stumbling way with the detour over all meshes in-between. Therefore, global dependencies can only be reproduced in a delayed temporal way, see figure 5.

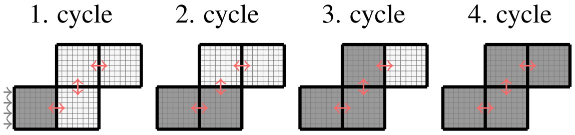


Figure 5: Stumbling data transfer of multi-mesh FFT for the 2D pipe geometry

The higher the number of subdomains and the corresponding fragmentation of the global connectivity is, the larger this effect may be. In the worst case this can lead to massive deteriorations of the approximation quality and the stability of the whole method.

Unfortunately, the efficient parallelization of the global FFT-methodology is extremely difficult. In the single-mesh case all data required for the computations are easily accessible in the storage of the only processor in use. But in the multi-mesh case the global dependencies are broken up arbitrarily by the domain decomposition process. The required data are distributed over all processors and have to be exchanged frequently which is very inefficient from a computational point of view.

Usually, the underlying parallelization strategies are based on purely algebraic considerations and are not in agreement with geometrically motivated subdivisions. This fact immensely impedes the efficient parallelization of this class of methods.

In order to remedy the situation, the default mechanism for the pressure solution was extended with release 6 of FDS: Embedded in a surrounding iteration, the solution of the local FFT methods is not only performed once but a couple of times, until the normals of the single velocity components along internal boundaries match up to a specified 'velocity tolerance'. Additionally, a maximum number of allowed iterations may be defined, e.g.

```
&PRES  VELOCITY_TOLERANCE= 0.01
       MAX_PRESSURE_ITERATIONS = 100
```

If nothing is specified by the user, a velocity tolerance of 'characteristic mesh size'/10 and a maximum of 10 iterations are taken as default values. Figure 6 shows a graphical visualization of the methodology.

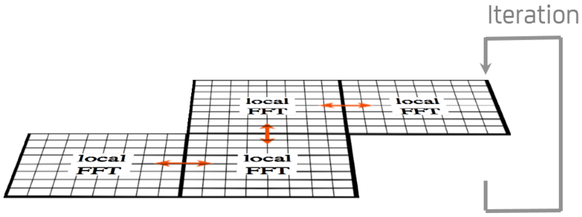


Figure 6: Multi-mesh FFT with additional averaging iteration for the 2D pipe geometry

In fact, the iterative multi-mesh FFT has led to notable improvements for many problem constellations. But for cases with frequent variations of the global information flow, this purely local approach may be insufficient. In the worst case no reliable scalability to high numbers of subdomains can be reached. The computational accuracy of this approach doesn't seem to be universally predictable and the convergence speed may possibly be low. Besides, it seems to be difficult to find an optimal value for the tolerance a-priori.

Alternative ScaRC pressure solver

These difficulties suggest to use a totally different approach for the solution of the pressure equation. Subsequently, the alternative pressure solver ScaRC is demonstrated which tries to fill the presented gap and reach a substantial improvement in the quality of the pressure approximation.

ScaRC is based on a purely *iterative* methodology. It operates as some kind of modular system that combines the use of:

- *domain decomposition* methods which are responsible for the subdivision of the problem,
- *multigrid methods* which are responsible for the accuracy and speed of convergence.

Multigrid methods belong to the most efficient methods for the solution of large systems of equations arising from the discretization of partial differential equations. They are characterized by the ability to reach excellent convergence rates which are independent of the grid resolution while having an asymptotically optimal complexity at the same time, see e. g. [4, 6].

Starting from an initial guess, they need multiple updating cycles during which the sequence of iterates converges more and more to the exact solution of the linear system. Thereby, a single iteration is considerably less complex than the one and only pass through a direct method such as FFT and can be led back to a series of core components (matrix-vector products, vector-linear combinations, scalar products), which are very well suited for a hardware-optimized implementation.

Multigrid methods are based on the application of so-called *relaxation methods*. Typical representatives of those methods are the damped Jacobi method or the Gauss-Seidel method. During the iteration process relaxation methods usually achieve a very fast reduction of the high frequency error components whereas the low frequency error components are usually damped out very slow.

This special property is described by the term *smoothing property*. If the solution vector has already undergone a sufficient number of smoothing steps and is then restricted by an averaging process to the next coarser grid, lets say with the double step width, then the error components which previously appeared to be low frequent on the finer grid turn out to be high frequent again on the coarser grid and may be damped out by an additional relaxation method there.

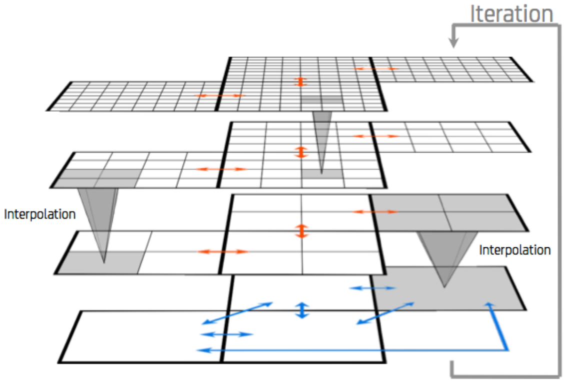


Figure 7: Geometric multigrid variant of ScaRC with different grid levels

As illustrated in figure 7, this process can be repeated recursively until a final coarse grid level has been reached on which an exact solution can be computed. From there, the resulting solution vector can successively be interpolated back to the next finer grid level, until the finest level is reached again. It is advisable to perform a certain number of post-smoothing steps on each level again when going the way back from the coarsest to the finest grid level.

In contrast to the current multi-mesh FFT which only relies on the application of a fine grid ScaRC thus uses a complete hierarchy of increasingly coarser grids. On all grid levels except of the coarsest one only next-neighbor data exchanges are needed. But on the coarsest grid level each mesh has to exchange data with each other one which is more expensive from a computational point of view but seems to be necessary with respect to the strong global connectivity.

Regarding the employed multigrid techniques two main classes are distinguished:

- *Geometric multigrid methods* (GMG) use the geometric information of the underlying problem for the definition of the single method components. The grid coarsening is usually based on a doubling of the grid size. On all grid levels the same matrix stencils can be used. GMG can be applied for broad classes of problems which possess certain regularity properties und usually achieve excellent convergence rates there.

- *Algebraic multigrid methods* (AMG) are only based on algebraic informations which are available in the linear system of equation. On the single grid levels usually different matrix stencils (with more or less computational complexity) arise. Due to their greater flexibility, they may particularly be applied for more complex problems, especially for irregular geometric situations. Usually they possess very good convergence properties, but the convergence theory is still more incomplete compared to GMG.

Both variants are successfully used in many scientific fields and can be applied to more general linear systems than most direct methods.

In ScaRC both variants are available whereby the implementation of GMG is currently more advanced than that of AMG. But independently of the special type of the multigrid method, the passing through the complete grid hierarchy as well as the solution of the coarse grid problem contribute to a very strong global connectvity. Each grid level is responsible for the coupling of a different range of the global information.

In contrast to the multi-mesh FFT, this approach uses a global discretization for the whole problem based on a domain-spanning matrix. In fact, this global matrix A is a purely formal construct which is never assembled as a whole, but only exists in a distributed sense, namely as a set of restrictions A_i to the single subdomains. But the whole iterative process is globally defined and uses the solution of the subdomain problems only for the approximate correction of the global residual which is measured in the mentioned relaxation methods.

All matrix-vector and vector-vector operations produce the same result as a hypothetic serial computation would do (if it really was available for the regarded case) which is denoted as *data-parallel execution*. There is no need to impose artificial boundary conditions along the interfaces of the subdomains because internal boundary cells can be treated as usual internal cells with respect to the virtual global matrix.

At the end of the computation a global solution is produced where the values on different subdomains are consistent along internal boundaries. The normals of the single velocity components automatically match up to machine precision.

As already mentioned, the termination of the iterative process requires the definition of a stopping criterion specifying which accuracy is required for the residual. Again, this stopping criterion has to be chosen very carefully.

The iterative process may be continued until machine precision for the measured residual has been reached. In this case, the resulting solution should in fact be the same as for a hypothetic single-mesh computation. In order to save computational costs, the iterative process can be stopped earlier. In this case, only an approximation to the required solution is reached.

Because of the strong global coupling this may possibly impair the accuracy of the whole method such that a proper compromise between time efficiency and accuracy has to be found which is not always an easy task. Experience has shown that a residual accuracy of 10^{-10} or even 10^{-8} is usually enough to reach a satisfactory accuracy of the global solution.

NUMERICAL TEST CASES

In agreement with the main developers ScaRC was already integrated into the FDS source code. In the course of several verification tests the convergence and scalability properties for both, the multi-mesh FFT and ScaRC, have been widely analyzed and compared, see for example [3].

The results of those tests are encouraging: Especially for test cases where no steady state may be reached ScaRC leads to comprehensive improvements with respect to accuracy and scalability compared to the current multi-mesh FFT.

An essential criterion for the design of the test cases was to separate the effects of the pressure solver from those of the other solution components to the best possible extend. This seems to be the only way to draw meaningful conclusions on the quality of the pressure solver itself. Due to very complex interactions of the single procedural components, superpositions or even effacements of different effects may happen otherwise which are only difficult to comprehend and which impede a subjective evaluation.

Two- as well as three-dimensional test cases were considered. A particular attention was placed on the requirement that the test cases should possess significant global effects such that the ability of the different solvers to scope with a strong global information transfer could be thoroughly checked.

Verification test 'scarc3d'

The 'scarc3d'-case follows the upper design criterion. It is defined on a simple cube-shaped domain in 3D and can be regarded as the generalization of the already existing 2D-test case 'scarc2d' from the official verification directory of FDS to 3D.

At three sides of the cube a ramp-based inflow is defined, see figure 8. One special feature of this case is that the different inflow velocities are steadily modified on all three regarded sides.

The different inflows rise up and down from a velocity of 0 to 2 m/s in small time intervals of $0.05\ s$, starting with different initial velocities each.

Due to this special setting this case has to scope with continually changing informations encompassing the whole domain being an endurance test for the different pressure solvers.

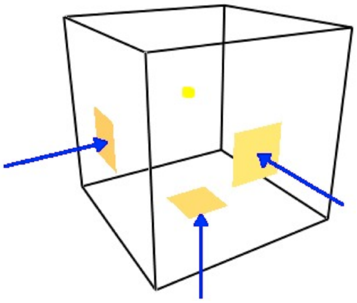


Figure 8: scarc3d-geometry with steadily changing inflows from three sides

Now, the focus is on how good both multi-mesh methods are able to preserve the accuracy of the basic single-mesh method and how good they can be scaled to large numbers of submeshes.

The whole cube is refined in $N = 128$ grid cells in each direction. As shown in figure 9, two different (regular) subdivisions with a proportional number of grid cells for each submesh are considered:

- 4x4x4: 64 meshes, 32^3 grid cells each,
- 8x8x8: 512 meshes, 16^3 grid cells each.

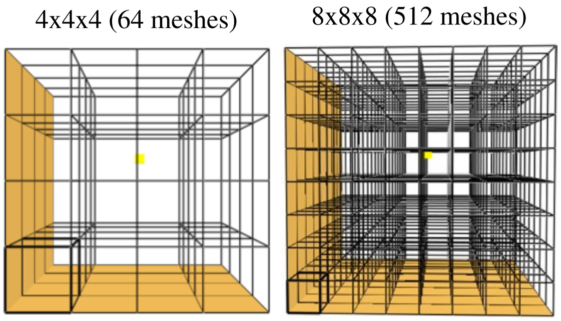


Figure 9: 64- and 512-mesh decomposition for the 'scarc3d'-case

Approximately in the middle of the domain, a pressure device is defined which records the corresponding pressure values over the complete course of the computation.

For all constellations, the values of the multi-mesh FFT and ScaRC are compared to each other. Thereby, different settings for the velocity tolerance of the multi-mesh FFT are regarded, namely the already mentioned default setting as well as the values $tol = 10^{-2}, 10^{-3}$ and 10^{-4} together with a corresponding enlargement of the admissible number of iterations.

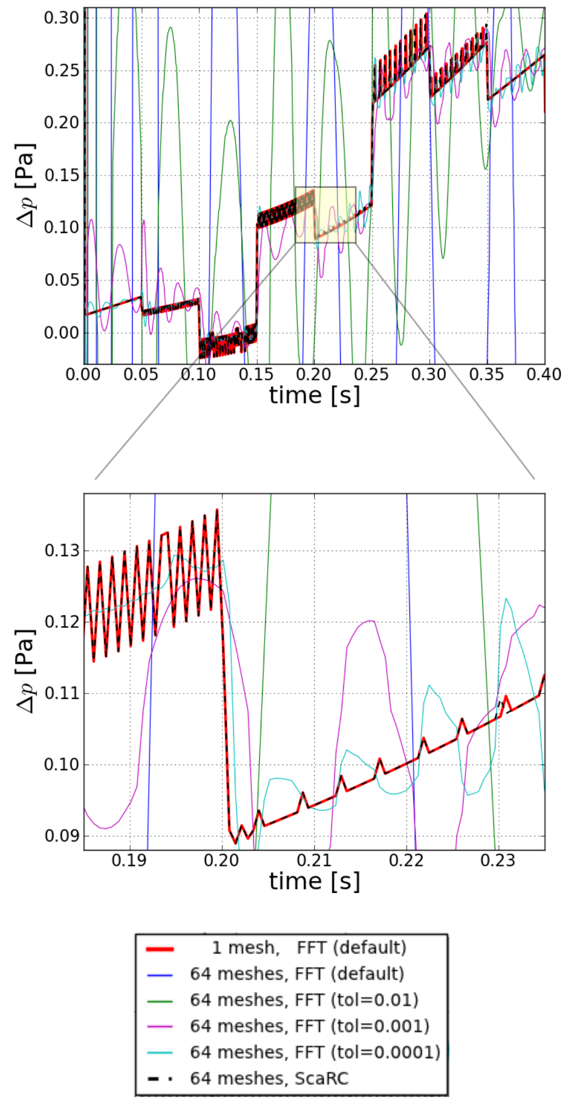


Figure 10: Comparison of the pressure devices for the 64-mesh cases against the 1-mesh case

As a best possible reference value, the pressure device of the corresponding single-mesh case (based on a global FFT-solver) is compared against the computed pressure devices for the different multi-mesh cases, see the upper picture of Figure 10.

The device for the single-mesh case is plotted in red. Obviously, the default multi-mesh FFT (blue line) doesn't show any consistency with the single-mesh result for the 64-mesh case. Reducing the velocity tolerance leads to slow improvements: While a tolerance of 10^{-2} (green line) hasn't much effect, further reductions to values of of 10^{-3} (magenta line) and especially 10^{-4} (cyan line) show an increasingly better accordance, but without completely matching the course of the single-mesh device. This can be seen in more detail in the lower picture of figure 10 which enlarges the indicated area. Here, the differences between the single- and multi-mesh FFT-computations (even for the finest velocity tolerance of 10^{-4}) get more obvious.

In contrast to that, ScaRC (dashed line) fits completely with the single-mesh course, both for the oscillating parts as well as the jumps which correspond to the points in time when the global inflow situations are modified (multiples of 0.05 s).

Due to the higher degree of fragmentation the differences for the multi-mesh FFT are still more distinct in the 512-mesh case. Figure 11 shows the corresponding course of the pressure devices for the different computations. But even for this high number of subdomains, ScaRC reproduces the oscillating behavior of the pressure device quite well.

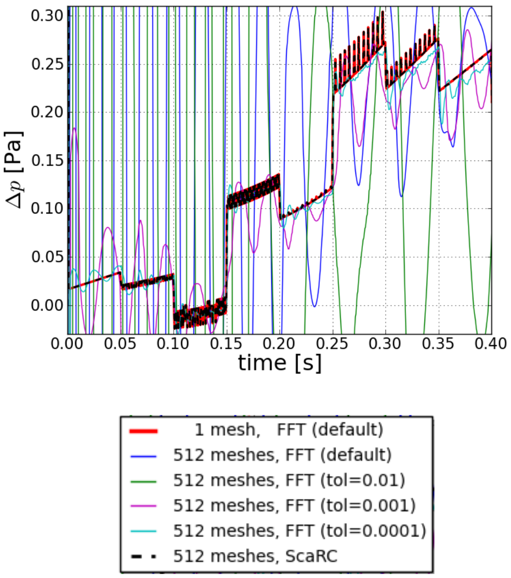


Figure 11: Comparison of the pressure devices for the 512-mesh cases against the 1-mesh case

Furthermore, figure 12 shows that each decrease of the velocity tolerance leads to a comprehensive increase of the number of iterations needed by the multi-mesh FFT to reach the specified tolerance. Especially at those points in time when the inflow conditions change, a sudden growth can be experienced reflecting the fact that this method has trouble to catch global effects in time.

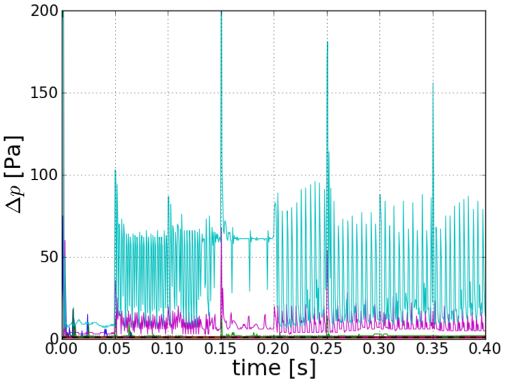


Figure 12: Number of pressure iterations for the 64-mesh FFT solver, $N = 128$

It gets clear that the computational costs may comprehensively rise up in order to reach an accuracy which is comparable to the single-mesh case. Figure 13 shows an additional comparison of the required computing times in case of the 512-mesh subdivision.

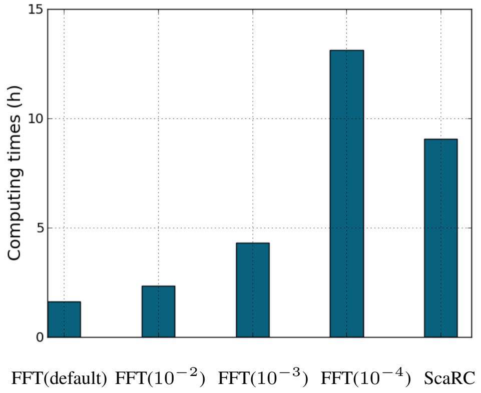


Figure 13: Computing times for the different 512-mesh computations

Regarding the default setting and the tolerances 10^{-2} and 10^{-3}, the computation times for ScaRC are considerably longer than that of the multi-mesh FFT. But a fair comparison should take into account that only in case of a velocity tolerance of 10^{-4} the multi-mesh FFT reaches a nearly comparable accuracy leading to computing times which are notedly higher than for ScaRC.

The relations for the computing times in case of other applications may differ depending on the number of submeshes, the degree of global information flow and the optimal tolerances for the multi-mesh FFT.

Verification test Shunn3

The second test example is based on the 'shunn3'-case from the official FDS Verification Guide [2] called 'Variable-density manufactured solution' which clearly demonstrates the second-order accuracy of the FDS time-stepping algorithm.

Based on the manufactured solution proposed by Shunn et al. [5] the analytical solutions for the mixture fraction Z, the density ρ, the velocities u and v and the hydrodynamic pressure $\tilde{p}$ are available.

The underlying geometry is a square-shaped domain in 2D with side length $L = 2\,m$ which is regarded as a single-mesh case. Different grid resolutions $\Delta x = L/N$ with $N = \{32, 64, 128, 256, 512\}$ in each direction were considered.

As illustrated in figure 14 for the density the solution is spatially periodic and translates diagonally from the lower left to the upper right hand side of the domain whereby the single density peaks fade in and out continuously.

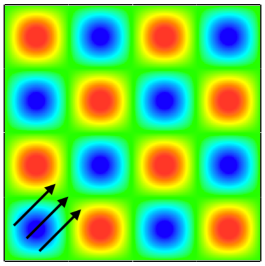


Figure 14: Density for shunn3 in the 1-mesh case

In order to analyze if ScaRC is able to preserve the second-order accuracy of the single-mesh case, subsequently a 4-mesh (2x2) and 16-mesh (4x4) decomposition of the domain were considered.

For different simulation times of the 16-case figure 15 shows the evolution of the density which fits very good to the course of the single mesh case.

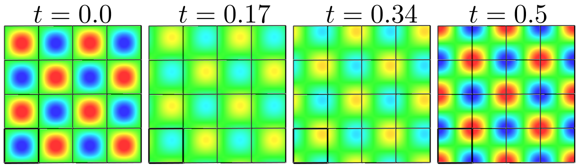
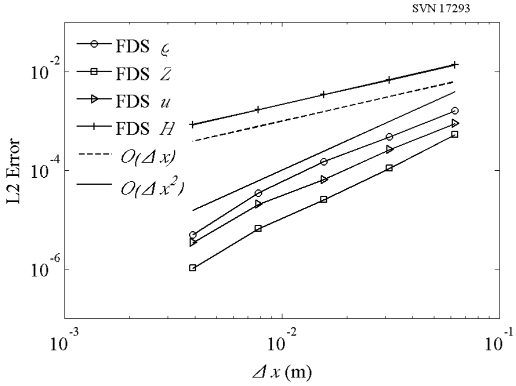


Figure 15: Spatially periodic translation of the density for the 16-case

In parts (A) and (B) of figure 16 the L2-errors for the two different decompositions are plotted as a function of all considered grid spacings.

(A) 2x2-decomposition (4-mesh):



(B) 4x4-decomposition (16-mesh):

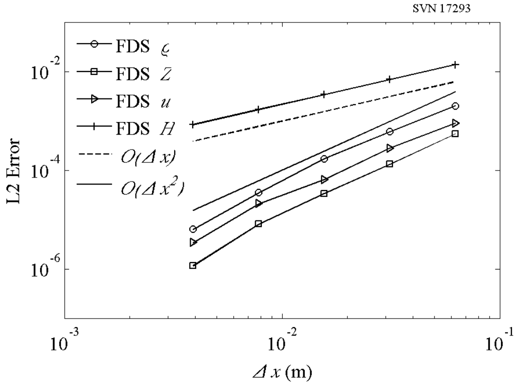


Figure 16: Convergence of ScaRC for shunn3 for different mesh-decompositions

Obviously, the second-order accuracy for density, mixture fraction and velocity, which was achieved for the single-mesh FFT in [2] can also be confirmed for the different multi-mesh ScaRC settings. Because ScaRC solves the same global system of equations as the single-mesh case does, the corresponding convergence behavior is preserved. Similar results were obtained for other test cases from the FDS Verification Guide [2] such as 'pulsating' or 'ns2d'.

Final discussion

The illustrated examples as well as a variety of other tests which have been performed over the past clearly show the gain of accuracy and scalability for the pressure solution which can be reached by the new ScaRC-methodology, especially for cases with a considerable variation of the global information flow. Since ScaRC is based on the solution of the corresponding global system of equations, the basic characteristics of the pressure equation such as the strong global coupling are well reproduced.

Further optimizations of the algebraic multigrid variant as well as the use of meshwise different grid resolutions are currently in work.

The ScaRC method isn't optimized from a computational point of view yet. If there is no namable degree of global information flow, it usually needs higher computing times than the default multi-mesh FFT. However for test cases which don't reach a steady state, the situation changes considerably and ScaRC proves superior. Basically, the exclusive look at the computational times cannot be an adequate evaluation criterion as long as no comparable accuracy is reached at the same time. In fact, the numerical efficiency with regard to accuracy and stability should the most important measure.

REFERENCES

[1] Kevin B. McGrattan, Simo Hostikka, Randall McDermott, Jason Floyd, Craig Weinschenk, and Kristopher Overholt. Fire Dynamics Simulator (Version 6) Technical Reference Guide, Volume 1: Mathematical Model. National Institute of Standards and Technology, 6.1.1 edition, July 2014.

[2] Kevin B. McGrattan, Simo Hostikka, Randall McDermott, Jason Floyd, Craig Weinschenk, and Kristopher Overholt. Fire Dynamics Simulator (Version 6) Technical Reference Guide, Volume 2: Verification. National Institute of Standards and Technology, 6.1.1 edition, July 2014.

[3] M. Münch and S. Kilian. 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.

[4] Y. Saad. Iterative methods for sparse linear systems. Society for Industrial and Applied Mathematics, Philadelphia, PA, second edition, 2003.

[5] L. Shunn, F. Ham, and P. Moin. Verification of variable-density flow solvers using manufactured solutions. J. Comput. Phys., 231:3801–3827, 2012.

[6] U. Trottenberg, C. W. Oosterlee, and A. Schüller. Multigrid. Applied Mathematical Sciences. Elsevier Academic Press, 2007.