Implementation and Performance Analyses of a Highly Efficient Algorithm for Pressure-Velocity Coupling

Implementierung und Untersuchung einer hoch effizienten Methode zur Druck-Geschwindigkeits-Kopplung

Master-Thesis von Fabian Gabel

Tag der Einreichung:

1. Gutachten: Prof. Dr. rer. nat. Michael Schäfer
2. Gutachten: Dipl.-Ing Ulrich Falk

TECHNISCHE UNIVERSITÄT DARMSTADT

Studienbereich CE
FNB

Implementation and Performance Analyses of a Highly Efficient Algorithm for Pressure-Velocity Coupling
Implementierung und Untersuchung einer hoch effizienten Methode zur Druck-Geschwindigkeits-Kopplung

Vorgelegte Master-Thesis von Fabian Gabel

1. Gutachten: Prof. Dr. rer. nat. Michael Schäfer
2. Gutachten: Dipl.-Ing Ulrich Falk

Tag der Einreichung:

Erklärung zur Master-Thesis

Hiermit versichere ich, die vorliegende Master-Thesis ohne Hilfe Dritter nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 6. März 2015

(Fabian Gabel)

Contents

Nomenclature **5**

1 Introduction **7**

2 Fundamentals of Continuum Physics for Thermo-Hydrodynamical Problems **8**
2.1 Conservation of Mass – Continuity Equation . . . 8
2.2 Conservation of Momentum – Cauchy-Equations . . . 8
2.3 Closing the System of Equations – Newtonian Fluids . . . 9
2.4 Conservation of Scalar Quantities . . . 9
2.5 Necessary Simplification of Equations . . . 9
2.5.1 Incompressible Flows and Hydrostatic Pressure . . . 9
2.5.2 Variation of Fluid Properties – The Boussinesq Approximation . . . 10
2.6 Final Form of the Set of Equations . . . 11

3 Finite Volume Methods for Incompressible Flows – Theoretical Basics **12**
3.1 Numerical Grid . . . 12
3.2 Approximation of Integrals and Derivatives . . . 13
3.3 Treatment of Non-Orthogonality of Grid Cells . . . 14
3.3.1 Minimum Correction Approach . . . 14
3.3.2 Orthogonal Correction Approach . . . 14
3.3.3 Over-Relaxed Approach . . . 14
3.3.4 Deferred Non-Orthogonal Correction . . . 14
3.4 Numerical Solution of Non-Linear Systems – Linearization Techniques . . . 15
3.5 Numerical Solution of Linear Systems with Krylov Subspace Methods . . . 15

4 Implicit Finite Volume Method for Incompressible Flows – Segregated Approach **16**
4.1 Discretization of the Mass Balance . . . 16
4.2 A Pressure-Weighted Interpolation Method for Velocities . . . 16
4.3 Implicit Pressure Correction and the SIMPLE Algorithm . . . 19
4.4 Discretization of the Mass Fluxes and the Pressure Correction Equation . . . 22
4.5 Discretization of the Momentum Balance . . . 23
4.5.1 Linearization and Discretization of the Convective Term . . . 23
4.5.2 Discretization of the Diffusive Term . . . 24
4.5.3 Discretization of the Source Terms . . . 25
4.6 Discretization of the Temperature Equation . . . 25
4.7 Boundary Conditions . . . 26
4.7.1 Dirichlet Boundary Conditions . . . 26
4.7.2 Treatment of Wall Boundaries . . . 26
4.7.3 Treatment of Block Boundaries . . . 28
4.8 Treatment of the Singularity of the Pressure Correction Equation with Neumann Boundaries . . . 28
4.9 Structure of the Assembled Linear Systems . . . 28

5 Implicit Finite Volume Method for Incompressible Flows – Fully Coupled Approach **31**
5.1 The Fully Coupled Algorithm – Pressure-Velocity Coupling Revised . . . 31
5.2 Coupling to the Temperature Equation . . . 32
5.2.1 Decoupled Approach – Explicit Velocity-to-Temperature Coupling . . . 32
5.2.2 Implicit Velocity-to-Temperature Coupling . . . 32
5.2.3 Temperature-to-Velocity/Pressure Coupling – Newton-Raphson Linearization . . . 33
5.3 Boundary Conditions on Domain and Block Boundaries . . . 34
5.4 Assembly of Linear Systems – Final Form of Equations . . . 34

6 CAFFA Framework **38**
6.1 PETSc Framework . . . 38
6.2 Grid Generation and Preprocessing . . . 38
6.3 Preprocessing . . . 39

6.4 Implementation of CAFFA . . . 39
6.4.1 The Message-Passing Model . . . 39
6.4.2 Convergence Control . . . 39
6.4.3 Indexing of Variables and Treatment of Boundary Values . . . 40
6.4.4 Domain Decomposition, Exchange of Ghost Values and Parallel Matrix Assembly . . . 40

7 Verification of the developed CAFFA Framework **42**
7.1 The Method of Manufactured Solutions for Navier-Stokes Equations . . . 42
7.2 Manufactured Solution for the Navier-Stokes Equations and the Temperature Equation . . . 42
7.3 Measurement of Error and Calculation of Order . . . 44
7.4 Influence of the Under-Relaxation Factor for the Velocities . . . 44

8 Comparison of Solver Concepts **47**
8.1 Parallel Performance . . . 47
8.1.1 Employed Hardware and Software – The Lichtenberg-High Performance Computer . . . 47
8.1.2 Measures of Performance . . . 47
8.1.3 Preliminary Upper Bounds on Performance – The STREAM Benchmark . . . 48
8.1.4 Optimization of Sequential Solver Configuration . . . 49
8.1.5 Speedup Measurement and Impact of Coupling Algorithm for Analytic Test Cases . . . 49
8.2 Realistic Testing Scenario – Complex Geometry . . . 49
8.3 Classical Benchmarking Case – Heat-Driven Cavity Flow . . . 53

9 Conclusion and Outlook **54**

References **55**

List of Figures

1 Vertex centered, cell centered and staggered variable arrangement . . . 12
2 Block structured grid consisting of two blocks . . . 13
3 Minimum correction, orthogonal correction and over-relaxed approach . . . 15
4 Possible interpretation of a virtual control volume (grey) located between nodes P and Q . . . 17
5 Non-zero structure of the linear systems used in the SIMPLE algorithm for a block structured grid consisting of one $2 \times 2 \times 2$ cell and one $3 \times 3 \times 3$ cell block . . . 30
6 Non-zero structure of block submatrices of the linear systems used in the coupled solution algorithm for a block structured grid consisting of one $2 \times 2 \times 2$ cell and one $3 \times 3 \times 3$ cell block. The blue coefficients represent the pressure-velocity coupling, the red coefficients correspond to the velocity-to-temperature coupling and the green coefficients result from the Newton-Raphson linearization technique. . . . 35
7 Non-zero structure the linear system used in the coupled solution algorithm for a block structured grid consisting of one $2 \times 2 \times 2$ cell and one $3 \times 3 \times 3$ cell block. The variables have been ordered such that each major matrix block refers to only one variable or the coupling between exactly two variables. . . . 36
8 Non-zero structure of the linear system used in the coupled solution algorithm for a block structured grid consisting of one $2 \times 2 \times 2$ cell and one $3 \times 3 \times 3$ cell block. The variables have been interlaced and the matrix consists of blocks as shown in figure 5.4. . . . 37
9 Comparison of calculated error for different under relaxation factors $\alpha_{\mathbf{u}}$ on a grid with different grid resolutions unknowns . . . 46
10 Sustainable memory bandwidth as determined by the STREAM benchmark (Triad) for different process binding options on one node of the MPI1 section . . . 49
11 Sustainable memory bandwidth as determined by the STREAM benchmark (Triad) for different process binding options on one node of the MPI2 section . . . 50
12 Wall clock time comparison for segregated and fully-coupled solution algorithm solving for an analytical solution on a grid with 128^3 cells . . . 50
13 Wall clock time comparison for segregated and fully-coupled solution algorithm solving for an analytical solution on a grid with 256^3 cells . . . 51
14 Sketch of the channel flow problem domain . . . 51
15 East boundary of the numerical grid for the channel flow problem . . . 52
16 West boundary of the numerical grid for the channel flow problem . . . 52
17 Blocking for the two different obstacles within the problem domain of the channel flow . . . 53

List of Tables

1 Comparison of the errors of the velocity calculated by the segregated and the coupled solver for different grid resolutions and the resulting order of accuracy 45
2 Comparison of the errors of the pressure calculated by the segregated and the coupled solver for different grid resolutions and the resulting order of accuracy 45
3 Comparison of the errors of the temperature calculated by the segregated and the coupled solver for different grid resolutions and the resulting order of accuracy 45
4 Characteristic problem properties used in the channel flow test case 52
5 Performance analysis results of the channel flow problem 52
6 Characteristic problem properties used in the channel flow test case 53
7 Performance analysis results of the heated cavity flow problem 54

List of Algorithms

1 SIMPLE Algorithm 22
2 Fully Coupled Solution Algorithm 32

TODO LIST

- Bilder zum Konvergenzverlauf sind gut
- wall boundary condition treatment could improve convergence
- say that viscous dissipation is not accounted for in the energy balance
- midpoint integration rule or midpoint rule of integration
- remove equation numbering when there is no reference
- change the matrix coefficient indexing to $a_P^{u_i,p}$ like in [15].
- extend nomenclature
- align equations using the **alignat** environment
- check pressure correction equation on iteration indices (gradients)
- check simple chapter, since the right hand side lacks of deferred corrector and under-relaxed velocities
- mention the boundary conditions for the pressure correction
- check all headings for correct spelling
- mention that for an unknown velocity field the partial differential equation for the temperature is non-linear as well
- consistent use of either temperature or energy equation
- check the signs in the Boussinesq approximation
- pressure weighted interpolation method for large body forces?
- not only speed but also improvement of robustness
- instationary flows?
- align exponents in the equation for the Newton-Raphson linearization
- define the term consistent
- add citations to clipper, opencascade and maple, ICEM CFD
- a priori exact solution use this formulation
- Falsche Wahl der problem domain, führt zu global nicht erfüllter kontinuitätsgleichung. residuum der druckkorrectur entspricht dem massenfluss
- reference Comparison of finite-volume numerical methods with staggered and colocated
- read klaij again and use some of its arguments grids
- emphasize the flexibility of the implementation
- This is often referred to as "matrix-free", though it is still a Mat in PETSc (Mat in PETSc means "finite-dimensional linear operator").

8.1 Parallel Performance

In many cases, scientific code is used to solve complex problems regarding memory requirements to make calculation results available within short time. In both scenarios, code that is able to run in parallel can alleviate the mentioned challenges. Code that runs in parallel can allocate more memory resources which makes the calculation of complex problems feasible. If the code is scalable the program execution can be shortened by using more processors to solve a problem of constant size.

The solver framework that has been developed in the course of the present thesis, has been parallelized using the PETSc library. After introducing the used hardware and software, the central measures of parallel performance are presented. Then preliminary test results using low-level benchmarks are performed, which establish upper performance bounds on the parallel efficiency and the scalability of the developed solver framework. The results of the efficiency evaluation of the solver framework is presented in the last subsections.

8.1.1 Employed Hardware and Software – The Lichtenberg-High Performance Computer

All performance analyses that are presented in this thesis were conducted on the Lichtenberg-High Performance Computer, also known as *HHLR* (*Hessischer Hochleistungsrechner*) [1]. The cluster consist of different sections according to the used hardware. Throughout the thesis, tests were performed using the first and the second MPI section of the cluster. The first section consists of 705 nodes of which each runs two Intel®Xeon®E5-2670 processors and offers 32GB of memory. The second section consists of 356 nodes of which each runs two Intel Xeon E5-2680 v3 processors and offers 64GB of memory. As interconnect for both sections FDR-14 InfiniBand is used.

All tests programs were compiled using the Intel compiler suite version 15.0.0 and the compiler options

```
-O3 -xHost
```

As MPI implementation Open MPI version 1.8.2 was chosen. Furthermore the PETSc version 3.5.3 was configured using the options

```
--with-blas-lapack-dir=/shared/apps/intel/2015/composer_xe_2015/mkl/lib/intel64/ \
--with-mpi-dir=/shared/apps/openmpi/1.8.2_intel \
COPTFLAGS="-O3 -xHost" \
FOPTFLAGS="-O3 -xHost" \
CXXOPTFLAGS="-O3 -xHost" \
--with-debugging=0 \
--download-hypre \
--download-ml
```

It should be noted that as the configurations options show, to maximize the efficiency of PETSc, a math kernel library should be used that has been optimized for the underlying hardware architecture as is in the case of the present thesis the Intel *MKL* (*Math Kernel Library*). It should be noted that also the Open MPI library has been compiled using the Intel compiler suite.

8.1.2 Measures of Performance

This section establishes the needed set of measures to evaluate the performance of a solver program, which will be used in the following sections. The first measure is the plain measure of runtime T_P taken by a computer to solve a given problem, where $P \in \mathbb{N}$ denotes the number of involved processes. This so called *wall-clock* time can be measured directly by calling subroutines of the underlying operating system and corresponds to the human perception of the time, that has passed. It must be noted, that this time does not correspond to the often mentioned *CPU* time. In fact, CPU time is only one contributor to wall-clock time. Wall-clock time further contains the time needed for communication and I/O and hence considers idle states of the processor. On the other side CPU time only considers the time in which the processor is actively working. This makes wall-clock time not only a more complete but also more accurate time measure when dealing with parallel processors, since processor idle times due to communication are actively considered while neglected in CPU time.

While wall-clock time is an absolute measure that can be used to compare different solver programs, further relative measures are needed to evaluate the efficiency of one program regarding the parallelisation implementation. The main purpose of these measures is to attribute the different causes of degrading efficiency due to heavy parallelisation to the different contributing factors. A simple model [22, 56] considers three contributions, that form the total efficiency

$$E_P^{tot} = E_P^{num} \cdot E_P^{par} \cdot E_P^{load}.$$

The *numerical efficiency*

$$E_P^{num} := \frac{\text{FLOPS}(1)}{P \cdot \text{FLOPS}(P)}$$

considers the degradation of the efficiency of the underlying algorithm due to the parallelisation. Many efficient algorithms owe their efficiency to recursions inside the algorithm. In the process of decomposing this recursions, the efficiency of the algorithm degrades. It follows that this efficiency is completely independent of the underlying hardware.

The *parallel efficiency*

$$E_P^{par} := \frac{\text{TIME(parallel Algorithm on one processor)}}{P \cdot \text{TIME(parallel Algorithm on } P \text{ processors)}}$$

describes the impact of the need for inter process communication, if more than one processor is involved in the solution process. It should be noted, that this form of efficiency does explicitly exclude any algorithm related degrading, since the time measured corresponds to the exact same algorithms. It follows that the parallel efficiency only depends on the implementation of the communication and the hardware related latencies.

The *load balancing efficiency*

$$E_P^{load} := \frac{\text{TIME(calculation on complete domain)}}{P \cdot \text{TIME(calculation on biggest subdomain)}}$$

is formed by the quotient of the wall times needed for the complete problem domain and partial solves on subdomains. This measure does neither depend on hardware nor on the used implementation. Instead it directly relates to the size and partition of the grid.

It is not possible to calculate all three efficiencies at the same time using only plain wall clock time measurements of a given application. Different solver configurations have to be used to calculate them separately. Since the focus of investigation of the present thesis does not lie on load balancing, for the remainder of the thesis $E_P^{load} = 100\%$ is assumed. This does not present a considerable drawback, since an ideal load balancing is easily obtainable nowadays by the use of sophisticated grid partitioning algorithms [?] REFERENCES. Using identical algorithms for different numbers of involved processes implicitly achieves $E_P^{num} = 100\%$. In this case the parallel efficiency of an application can be measured through the quotient of the needed wall clock time. To measure the numerical efficiency of an algorithm the respective hardware counters have to be evaluated. This can be done using the built in log file functionality of PETSc as presented in section REFERENCE. Hence the determination of numerical efficiency does not rely on wall clock time.

Another common performance measure is the *Speed-Up*

$$S_P = \frac{T_1}{T_P} = P \cdot E_P^{tot}.$$

Speedup and parallel efficiency characterize the parallel scalability of an application and determine the regimes of efficient use of hardware resources.

8.1.3 Preliminary Upper Bounds on Performance – The STREAM Benchmark

Scientific applications that solve partial differential equations rely on sparse matrix computations, which usually exhibit the sustainable memory bandwidth as bottleneck with respect to the runtime performance of the program [29]. The purpose of this section is to establish a frame in terms of an upper bound on performance in which the efficiency of the developed solver framework can be evaluated critically. As common measure for the maximum sustainable bandwidth, low-level benchmarks can be used, which focus on evaluating specific properties of the deployed hardware architecture. In this case the STREAM benchmark suite [44, 45] provides apt tests, which are designed to work with data sets that exceed the cache size of the involved processor architecture. This forces the processors to stream the needed data directly from the memory instead of reusing the data residing in their caches. These types of tests can be used to calculate an upper bound on the memory bandwidth for the CAFFA framework.

In terms of parallel scalability, the STREAM benchmark can also be used as an upper performance bound. According to [5] the parallel performance of memory bandwidth limited codes correlates with the parallel performance of the STREAM benchmark, i.e. a scalable increase in memory bandwidth is necessary for scalable application performance. The intermediate results of the benchmark can then be used to test different configurations that bind hardware resources to the involved processes. Before presenting results the different binding configurations will be explained.

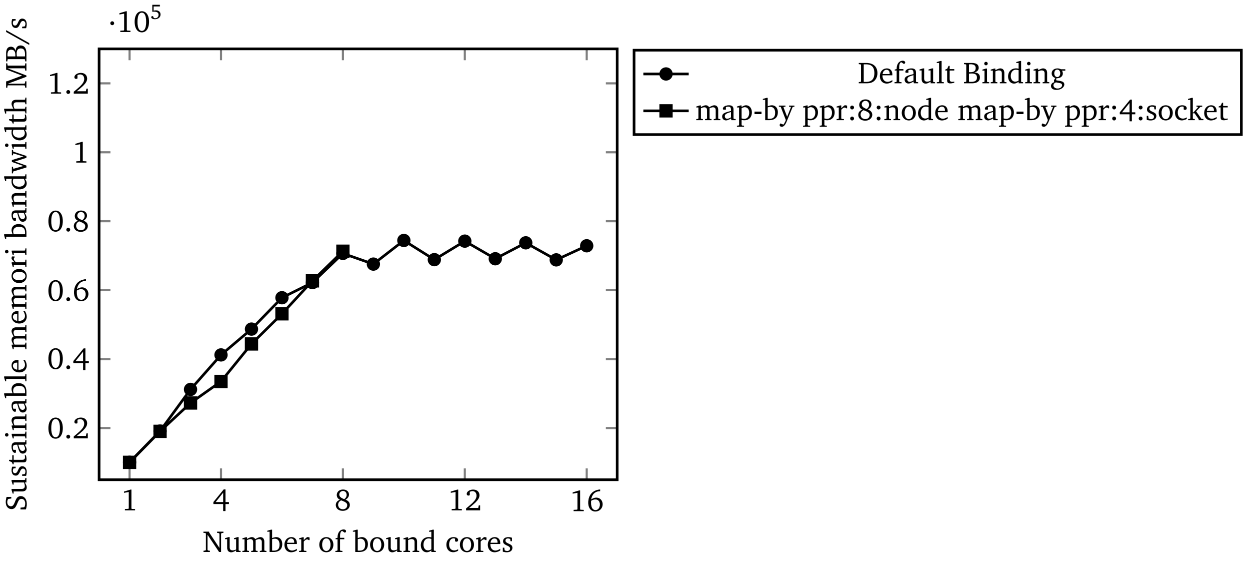

Figure 10: Sustainable memory bandwidth as determined by the STREAM benchmark (Triad) for different process binding options on one node of the MPI1 section

The first configuration sequentially binds the processes to the cores beginning on the first socket. When every core has a bound process the binding algorithm binds the following processes to cores of the second socket. The second configuration binds the processes in a round robin manner regarding the sockets. This configuration in difference to the second configuration binds one process to three cores. Figures **??**,**??** and **??** demonstrate the different binding options for two sockets and processors with twelve cores each, when eight processes are to be bound to the resources.

As can be seen from figures REFERENCE, the scaling of the sustainable bandwith behaves rather erratic, such that for process counts up until eight for the MPI1 section no reliable results can be obtained from the STREAM benchmark. It is assumed that this kind of behaviour is due to the automatic turbo boost the deployed processors apply, which can be controlled other than by turning it off. Since all performance measures are relative but the plain measurement of wall clock time the reference value for the following performance measurements will be taken from the program execution for the maximum number of processes that can be bound without overlap to one deployed socket.

8.1.4 Optimization of Sequential Solver Configuration

Compare runtime for different solver configurations BiCGStab+ICC, different multigrid algorithms

8.1.5 Speedup Measurement and Impact of Coupling Algorithm for Analytic Test Cases

This section presents the results from the speedup measurements conducted on the supercomputer HHLR which has been presented in section **??**. Furthermore the impact of the scaling algorithm on the running time of the corresponding solver program will be shown.

8.2 Realistic Testing Scenario – Complex Geometry

Fluid flow inside closed applications is a common situation in mechanical engineering. The flow through a channel with rectangular crosssection can be seen as a simple testcase that is a part of complex applications. This section compares the single process performance of the segregated and the fully coupled solution algorithm for a flow problem within a complex geometry. The geometry of the domain is based on a channel flow problem with square cross section, with the special property that inside the channel reside two obstacles with a square cross section of which one has been twisted against the other. Figure 14 shows a sketch of the problem domain.

This case exercises all of the previously introduced boundary conditions for flow problems including the treatment of non-matching block boundaries. For the velocities at the inflow boundary the parabolic distribution

$$\mathbf{u}(x_1, x_2, x_3) = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} \frac{16 * 0.45 * x_2 * x_3 * (0.41 - x_2) * (0.41 - x_3)}{0.41^4} \\ 0 \\ 0 \end{bmatrix}$$

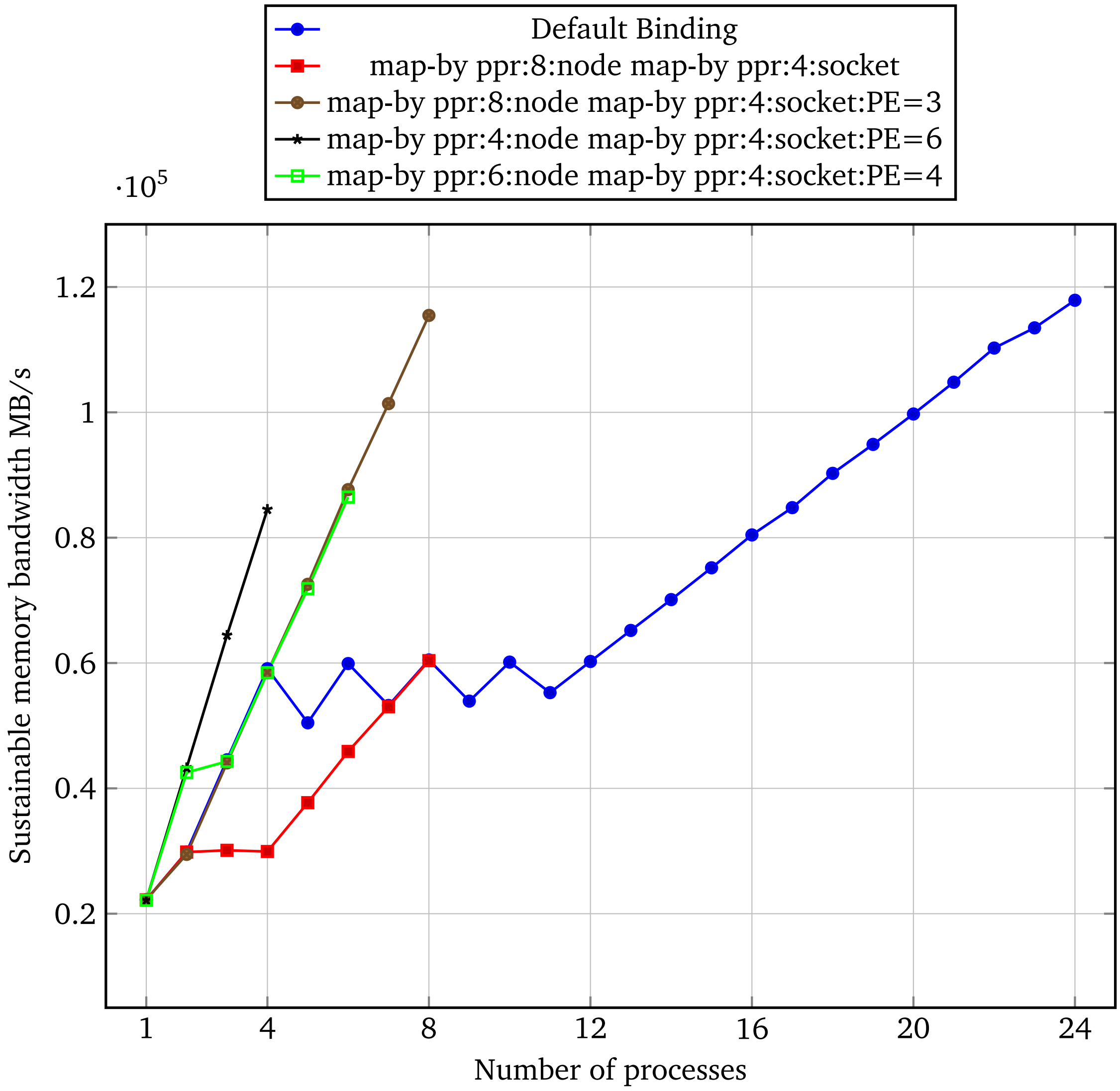

Figure 11: Sustainable memory bandwidth as determined by the STREAM benchmark (Triad) for different process binding options on one node of the MPI2 section

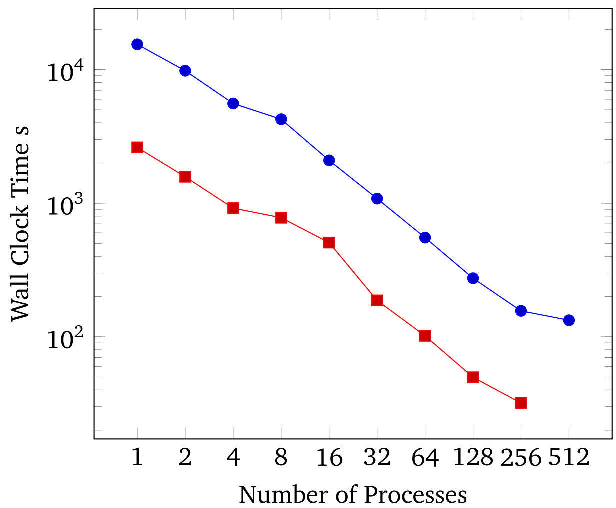

Figure 12: Wall clock time comparison for segregated and fully-coupled solution algorithm solving for an analytical solution on a grid with 128^3 cells

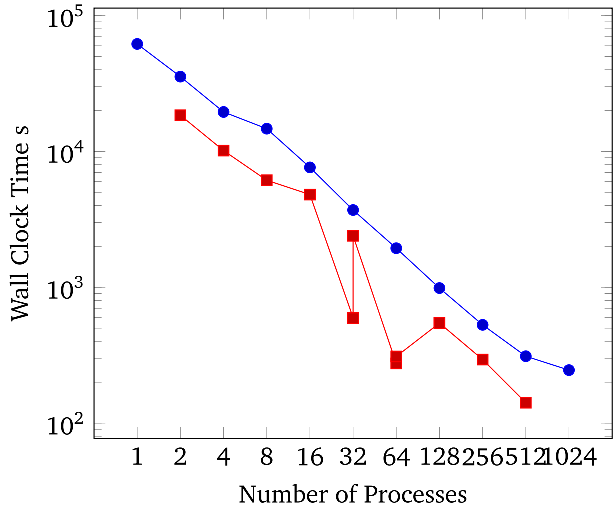

Figure 13: Wall clock time comparison for segregated and fully-coupled solution algorithm solving for an analytical solution on a grid with 256^3 cells

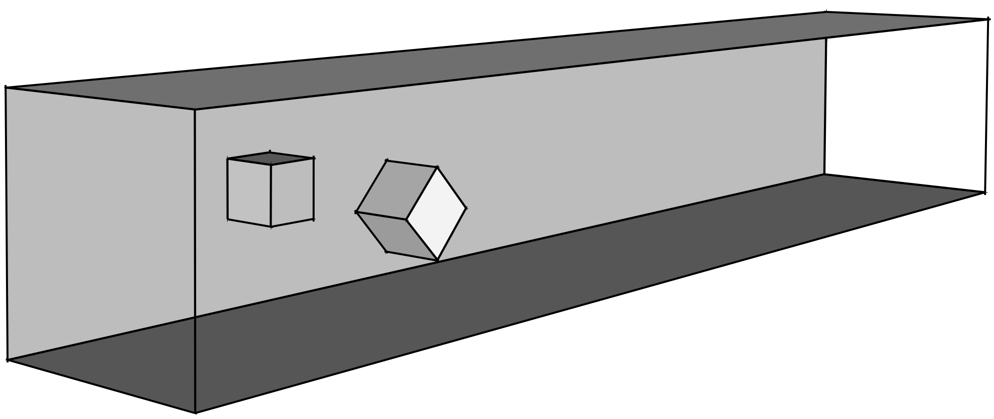

Figure 14: Sketch of the channel flow problem domain

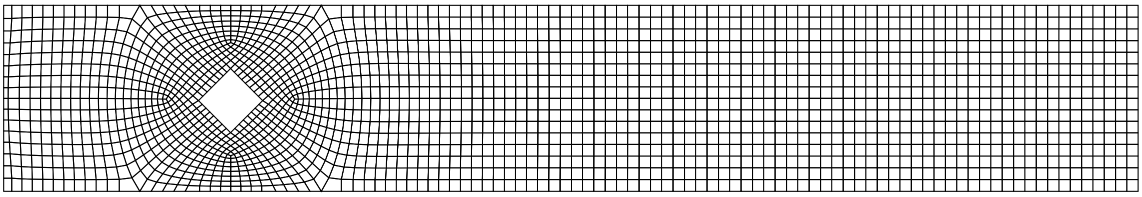

Figure 15: East boundary of the numerical grid for the channel flow problem

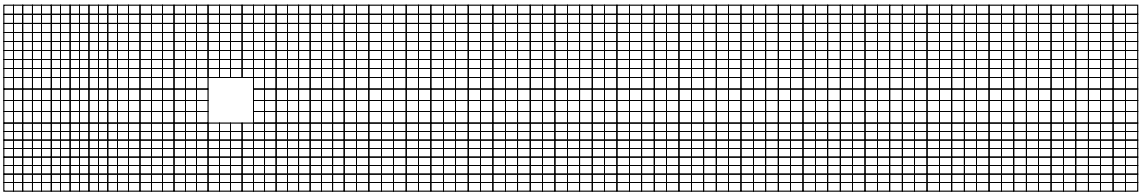

Figure 16: West boundary of the numerical grid for the channel flow problem

was chosen. All the other problem parameters were chosen such that the flow problem resides in the regime of a non-turbulent stationary flow for which the presented solver framework has been developed. Table 4 lists the remaining material and geometrical characteristics of the test case.

Property	Value	Unit
Density	1E-3	kg/m^3
Viscosity	1E-0	K.A.
Height	0.41	m
Length	2.5	m
Side length cube	0.1	m
Under relaxation u	0.9	
Under relaxation p	0.1	

Table 4: Characteristic problem properties used in the channel flow test case

The present test case shows one advantage of the treatment of block boundaries, which has been introduced in section REFERENCE. Since no assumptions on the geometry of a neighboring block are necessary each block can be constructed independently which increases the flexibility of the meshing of geometries. Furthermore, because of the fully implicit handling of block boundaries, the number of used blocks does not impact on the convergence properties of the deployed linear solvers. Figure 8.2 and figure 8.2 show the mesh at the left and right bounding walls. It is evident that this mesh leads to non-trivial transitions between the blocks. Figure **??** shows the domain decomposition into structured grid blocks around the two obstacles within the problem domain and emphasizes the need for accurate handling of non-matching block boundaries.

The solution of the linear systems resulting from the discretization of the problem takes up more time for the coupled solution algorithm than in the segregated coupled algorithm. For small problem sizes the additional overhead for the solution methods for linear systems and the property that the segregated solution algorithm does not need many outer iterations to converge leads to the conclusion that moderate to big problems with respect to the number of involved unknowns are necessary for the coupled solution algorithm to dominate through performance.

In contrast to the segregated algorithm, the fully coupled solution algorithm achieves an approximately constant amount of needed outer iterations, independent of the number of involved unknowns. The tests regarding the weak scaling of the coupled solution algorithm emphasize this property. Table 5 compares the measured wall clock time for different numbers of unknowns. The mesh shown in 8.2 and 8.2 was generated for the first number of unknowns and successively refined to achieve higher mesh resolutions. The two other numbers result from up to three times bisectioning the mesh in each direction, every time scaling the number of unknowns by a factor of eight. The tests were conducted on the formerly presented HHLR cluster, using the MPI2 and MEM2 section.

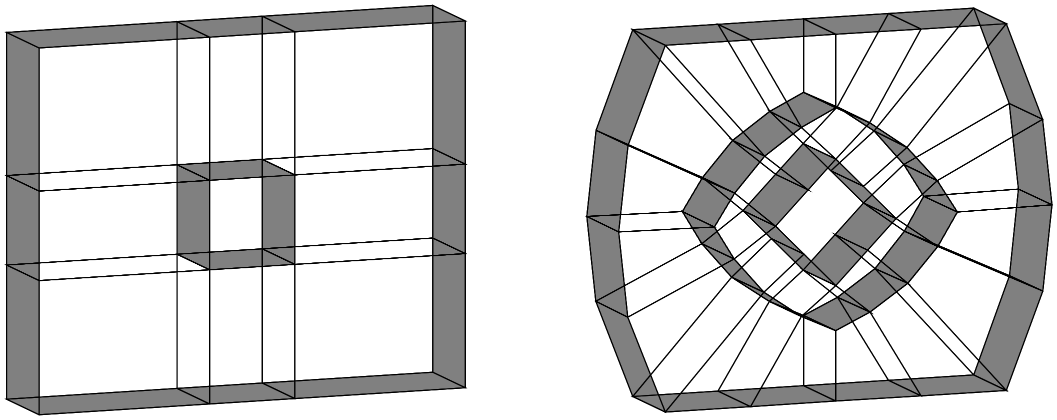

Figure 17: Blocking for the two different obstacles within the problem domain of the channel flow

No. of Unknowns	Seg. - time	Cpld - time	Seg. - its	Cpld - its
75768	0.2226E+02	0.3645E+02	151	67
408040	0.4053E+03	0.1500E+03	355	42
2611080	1.1352E+05	0.3105E+04	1592	39

Table 5: Performance analysis results of the channel flow problem

The timing results show, that already after the first grid refinement step the fully coupled solution algorithm performs better with respect to the needed wall clock time for computation. This effect is even more clearly visible for higher mesh resolutions.

8.3 Classical Benchmarking Case – Heat-Driven Cavity Flow

This section deals with the evaluation and comparison not only of the pressure-velocity coupling but also of the velocity-to-temperature coupling through the Boussinesq approximation and the temperature-velocity/pressure coupling through the Newton-Raphson linearization of the convective term in the temperature equation. For this the standard heat-driven cavity flow [13,16] is adapted for a three dimensional domain and the material and geometric parameters are chosen such that a non-turbulent stationary flow exists, which means that the solution lies within the regime of the approximations made by the solver.

Essential for this benchmarking case is the nature of the flow. The fluid motion is a consequence of the effect of volume forces caused by temperature differences in the solution domain, hence the mathematical problem exhibits a strong coupling bewteen the involved variables velocity, pressure and temperature. This relation is represented by the Rayleigh and Prandtl number. It is assumed that for this kind of flow the fully implicit treatment of the temperature coupling will yield further benefits with respect to wall time, compared with solution approaches that solve for the temperature separately. Table 6 lists the geometrical and solver parameters used for the performance analysis of this section.

The tests were conducted on the formerly presented HHLR cluster, using the MPI2 and MEM2 section. For the tests involving higher resolutions the relative tolerance for convergence was increased to $1E-4$. Table **??**

The presended results are in good aggreement with [65], which show monotonic decrease of the number of iterations with increased implicit coupling. It is notable that the pressure velocity coupling is responsible for the biggest decrease in the number of non-linear iterations. Different to the results presented in section [**?**] this does not yield significant performance benefits with respect to wall-clock time. In order to achive the benefits of a fully coupled solution algorithm the coupling has to be increased to also involve temperature-to-velocity/pressure and velocity-to-temperature coupling. Furthermore, the solely use of velocity-to-temperature coupling results in no benefits compared to the coupled solution for velocities and pressure combined with a decoupled solve for the temperature equation. This is accredited to the treatment of the non-linearity of the temperature equation in the TCPLD solver configuration. Even though the momentum balances implicitly use the temperature from the next iteration, the temperature equation does not use the velocities of the next iteration. Instead the convective fluxes are calculated with the velocities from the previous iteration. Even though the implicit velocity-to-temperature coupling reduces the number of needed non-linear iterations an increase in the needed wall-clock time for computation can be noticed. This is attributed to the augmented costs during the application of the

Property	Value	Unit
Density	1E-3	kg/m^3
Viscosity	1E-0	K.A.
Height	0.41	m
Length	2.5	m
Side length cube	0.1	m
Under relaxation u	0.9	
Under relaxation p	0.1	
Relative tolerance	1E-8/1E-4	

Table 6: Characteristic problem properties used in the channel flow test case

Resolution	Solver configuration	Time	Non-linear it.
32x32x32	SEG		
	CPLD		
	TCPLD		
	NRCPLD		
64x64x64	SEG	0.1997E+04	804
	CPLD	0.7687E+03	63
	TCPLD	0.1278E+04	59
	NRCPLD	0.4240E+03	17
128x128x128	SEG	0.5197E+05	3060
	CPLD	0.1860E+05	74
	TCPLD	0.1950E+05	50
	NRCPLD	0.6155E+04	18
256x256x256	SEG		
	CPLD		
	TCPLD		
	NRCPLD		

Table 7: Performance analysis results of the heated cavity flow problem

linear solver algorithm as a result of the additional degree of freedom that the linear system embraces. It can be concluded that in order to profit from the benefits of implicit coupling the key lays in the temperature-to-velocity/pressure coupling. However, it should be noted that the implicit consideration of the corresponding terms furthermore increases the amount of memory needed for computation and hence does not come without drawbacks.

References

[1]

[2] Anderson, D. A., Tannehill, J. C. and Pletcher, R. H. *Computational Fluid Mechanics and Heat Transfer*. Hemisphere Publishing Corporation, Wahington, 1984.

[3] Aris, R. *Vectors, Tensors and the Basic Equations of Fluid Mechanics*. Prentice-Hall, Inc., 1962.

[4] Balay, S., Abhyankar, S., Adams, M. F., Brown, J., Brune, P., Buschelman, K., Eijkhout, V., Gropp, W. D., Kaushik, D., Knepley, M. G., McInnes, L. C., Rupp, K., Smith, B. F. and Zhang, H. PETSc users manual. Tech. Rep. ANL-95/11 - Revision 3.5, Argonne National Laboratory, 2014.

[5] Balay, S., Abhyankar, S., Adams, M. F., Brown, J., Brune, P., Buschelman, K., Eijkhout, V., Gropp, W. D., Kaushik, D., Knepley, M. G., McInnes, L. C., Rupp, K., Smith, B. F. and Zhang, H. PETSc Web page. `http://www.mcs.anl.gov/petsc`, 2014.

[6] Balay, S., Gropp, W. D., McInnes, L. C. and Smith, B. F. Efficient management of parallelism in object oriented numerical software libraries. In *Modern Software Tools in Scientific Computing* (1997), E. Arge, A. M. Bruaset and H. P. Langtangen, Eds., Birkhäuser Press, pp. 163–202.

[7] Bond, R. B., Ober, C. C., Knupp, P. M. and Bova, S. W. Manufactured solution for computational fluid dynamics boundary condition verification. *AIAA Journal 45*, 9 (Sep 2007), 2224–2236.

[8] Chen, Z. and Przekwas, A. A coupled pressure-based computational method for incompressible/compressible flows. *Journal of Computational Physics 229*, 24 (2010), 9150 – 9165.

[9] Chisholm, T. and Zingg, D. W. A fully coupled newton-krylov solver for turbulent aerodynamics flows, 2002.

[10] Choi, S. K. Note on the use of momentum interpolation method for unsteady flows. *Numerical Heat Transfer, Part A: Applications 36*, 5 (1999), 545–550.

[11] Choi, S.-K., Kim, S.-O., Lee, C.-H. and Choi, H.-K. Use of the momentum interpolation method for flows with a large body force. *Numerical Heat Transfer, Part B: Fundamentals 43*, 3 (2003), 267–287.

[12] Choudhary, A., Roy, C., Luke, E. and Veluri, S. Fluid Dynamics and Co-located Conferences. American Institute of Aeronautics and Astronautics, Jun 2011, ch. Issues in Verifying Boundary Conditions for 3D Unstructured CFD Codes.

[13] Christon, M. A., Gresho, P. M. and Sutton, S. B. Computational predictability of time-dependent natural convection flows in enclosures (including a benchmark solution). *International Journal for Numerical Methods in Fluids 40*, 8 (2002), 953–980.

[14] Dahmen, W. and Reusken, A. *Numerik für Ingenieure und Naturwissenschaftler*, 2. ed. Springer Verlag, Berlin, 2008.

[15] Darwish, M., Sraj, I. and Moukalled, F. A coupled finite volume solver for the solution of incompressible flows on unstructured grids. *Journal of Computational Physics 228*, 1 (2009), 180 – 201.

[16] De Vahl Davis, G. Natural convection of air in a square cavity: A bench mark numerical solution. *International Journal for Numerical Methods in Fluids 3*, 3 (1983), 249–264.

[17] Deng, G., Piquet, J., Vasseur, X. and Visonneau, M. A new fully coupled method for computing turbulent flows. *Computers & Fluids 30*, 4 (2001), 445 – 472.

[18] Deng, G. B., Piquet, J., Queutey, P. and Visonneau, M. A new fully coupled solution of the navier-stokes equations. *International Journal for Numerical Methods in Fluids 19*, 7 (1994), 605–639.

[19] Elman, H., Howle, V. E., Shadid, J., Shuttleworth, R. and Tuminaro, R. A taxonomy and comparison of parallel block multi-level preconditioners for the incompressible navier-stokes equations. *J. Comput. Phys. 227*, 3 (Jan. 2008), 1790–1808.

[20] Elman, H., Howle, V. E., Shadid, J. and Tuminaro, R. A parallel block multi-level preconditioner for the 3d incompressible navier-stokes equations. *Journal of Computational Physics 187* (2003), 504–523.

[21] Falk, U. and Schäfer, M. A fully coupled finite volume solver for the solution of incompressible flows on locally refined non-matching block-structured grids. In *Adaptive Modeling and Simulation 2013* (Barcelona, Spain, June 2013), J. P. M. de Almeida, P. Diez, C. Tiago and N. Parez, Eds., pp. 235–246.

[22] Ferziger, J. H. and Perić, M. *Numerische Strömungsmechanik*. Springer Verlag, Berlin, 2002.

[23] Galpin, P. F. and Raithby, G. D. Numerical solution of problems in incompressible fluid flow: Treatment of the temperature-velocity coupling. *Numerical Heat Transfer 10*, 2 (1986), 105–129.

[24] Gray, D. D. and Giorgini, A. The validity of the boussinesq approximation for liquids and gases. *International Journal of Heat and Mass Transfer 19*, 5 (1976), 545 – 551.

[25] Gresho, P. M. and Sani, R. L. On pressure boundary conditions for the incompressible navier-stokes equations. *International Journal for Numerical Methods in Fluids 7*, 10 (1987), 1111–1145.

[26] Gropp, W., Lusk, E. and Skjellum, A. *Using MPI: portable parallel programming with the message-passing interface*, 2. ed. The MIT Press, Cambridge, Massachussets, 1999.

[27] Gropp, W. D., Kaushik, D. K., Keyes, D. E. and Smith, B. F. High performance parallel implicit cfd. *Parallel Computing 27* (2000), 337–362.

[28] Hadji, S. and Dhatt, G. Asymptotic–newton method for solving incompressible flows. *International Journal for Numerical Methods in Fluids 25*, 8 (1997), 861–878.

[29] Hager, G. and Wellein, G. *Introduction to High Performance Computing for Scientists and Engineers*. CRC, Boca Raton, 2011.

[30] Henderson, A. Paraview guide, a parallel visualization application, 2007.

[31] Hortmann, M., Perić, M. and Scheuerer, G. Finite volume multigrid prediction of laminar natural convection: Bench-mark solutions. *International Journal for Numerical Methods in Fluids 11*, 2 (1990), 189–207.

[32] Hui, W. Exact solutions of the unsteady two-dimensional navier-stokes equations. *Journal of Applied Mathematics and Physics ZAMP 38*, 5 (1987), 689–702.

[33] Hutchinson, B. R. and Raithby, G. D. A multigrid method based on the additive correction strategy. *Numerical Heat Transfer 9*, 5 (1986), 511–537.

[34] Jasak, H. *Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows*. PhD thesis, Imperial College of Science, Technology and Medicine, Jun 1996.

[35] Klaij, C. M. and Vuik, C. Simple-type preconditioners for cell-centered, colocated finite volume discretization of incompressible reynolds-averaged navier–stokes equations. *International Journal for Numerical Methods in Fluids 71*, 7 (2013), 830–849.

[36] Kundu, P. K., Cohen, I. M. and Downling, D. R. *Fluid Mechanics*, 5 ed. Elsevier, 2012.

[37] Lange, C. F., Schäfer, M. and Durst, F. Local block refinement with a multigrid flow solver. *International Journal for Numerical Methods in Fluids 38*, 1 (2002), 21–41.

[38] Li, W., Yu, B., Wang, X.-r. and Sun, S.-y. Calculation of cell face velocity of non-staggered grid system. *Applied Mathematics and Mechanics 33*, 8 (2012), 991–1000.

[39] Lilek, Ž., Muzaferija, S., Perić, M. and Seidl, V. An implicit finite-volume method using nonmatching blocks of structured grid. *Numerical Heat Transfer, Part B: Fundamentals 32*, 4 (1997), 385–401.

[40] Majumdar, S. Role of underrelaxation in momentum interpolation for calculation of flow with nonstaggered grids. *Numerical Heat Transfer 13*, 1 (1988), 125–132.

[41] Mangani, L., Buchmayr, M. and Darwish, M. Development of a novel fully coupled solver in openfoam: Steady-state incompressible turbulent flows in rotational reference frames. *Numerical Heat Transfer, Part B: Fundamentals 66*, 6 (2014), 526–543.

[42] Maple. *version 18*. Waterloo Maple Inc. (Maplesoft), Waterloo, Ontario, 2014.

[43] MATLAB. *version 7.10.0 (R2010a)*. The MathWorks Inc., Natick, Massachusetts, 2010.

[44] McCalpin, J. D. Stream: Sustainable memory bandwidth in high performance computers. Tech. rep., University of Virginia, Charlottesville, Virginia, 1991-2007. A continually updated technical report. `http://www.cs.virginia.edu/stream/`.

[45] McCalpin, J. D. Memory bandwidth and machine balance in current high performance computers. *IEEE Computer Society Technical Committee on Computer Architecture (TCCA) Newsletter* (Dec. 1995), 19–25.

[46] Meister, A. *Numerik linearer Gleichungssysteme*, 4. ed. Vieweg+Teubner, Wiesbaden, 2011.

[47] Miller, T. F. and Schmidt, F. W. Use of a pressure-weighted interpolation method for the solution of the incompressible navier-stokes equations on a nonstaggered grid system. *Numerical Heat Transfer 14*, 2 (1988), 213–233.

[48] Oberkampf, W. L. and Trucano, T. G. Verification and validation in computational fluid dynamics. *Progress in Aerospace Sciences 38*, 3 (2002), 209 – 272.

[49] Oliveira, P. J. and Issa, R. I. An improved piso algorithm for the computation of buoyancy-driven flows. *Numerical Heat Transfer, Part B: Fundamentals 40*, 6 (2001), 473–493.

[50] Patankar, S. and Spalding, D. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *International Journal of Heat and Mass Transfer 15*, 10 (1972), 1787 – 1806.

[51] Perić, M. Analysis of pressure-velocity coupling on nonorthogonal grids. *Numerical Heat Transfer 17* (Jan. 1990), 63–82.

[52] Pope, S. B. *Turbulent Flows*. Cambridge University Press, New York, 2000.

[53] Ramamurti, R. and Löhner, R. A parallel implicit incompressible flow solver using unstructured meshes. *Computers & Fluids 25*, 2 (1996), 119 – 132.

[54] Rhie, C. M. and Chow, W. L. Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA Journal 21* (Nov. 1983), 1525–1532.

[55] Salari, K. and Knupp, P. Code verification by the method of manufactured solutions. Tech. Rep. SAND2000-1444, Sandia National Labs., Albuquerque, NM (US); Sandia National Labs., Livermore, CA (US), Jun 2000.

[56] Schäfer, M. *Numerik im Maschinenbau*. Springer Verlag, Berlin, 1999.

[57] Schäfer, M. and Turek, S. Recent benchmark computations of laminar flow around a cylinder, 1996.

[58] Sheng, Y., Sheng, G., Shoukri, M. and Wood, P. New version of simplet and its application to turbulent buoyancy-driven flows. *Numerical Heat Transfer, Part A: Applications 34*, 8 (1998), 821–846.

[59] Sheng, Y., Shoukri, M., Sheng, G. and Wood, P. A modification to the simple method for buoyancy-driven flows. *Numerical Heat Transfer, Part B: Fundamentals 33*, 1 (1998), 65–78.

[60] Sheu, T. W. H. and Lin, R. K. Newton linearization of the incompressible navier–stokes equations. *International Journal for Numerical Methods in Fluids 44*, 3 (2004), 297–312.

[61] Silvester, D., Elman, H., Kay, D. and Wathen, A. Efficient preconditioning of the linearized navier–stokes equations for incompressible flow. *Journal of Computational and Applied Mathematics 128*, 1–2 (2001), 261 – 279. Numerical Analysis 2000. Vol. VII: Partial Differential Equations.

[62] Sotiropoulos, F. and Abdallah, S. Coupled fully implicit solution procedure for the steady incompressible navier-stokes equations. *Journal of Computational Physics 87*, 2 (1990), 328 – 348.

[63] Spurk, J. H. and Aksel, N. *Strömungslehre: Einführung in die Theorie der Strömungen*, 8. ed. Sringer Verlag, Berlin, 2010.

[64] Taylor, G. I. and Green, A. E. Mechanism of the production of small eddies from large ones. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences 158*, 895 (1937), pp. 499–521.

[65] Vakilipour, S. and Ormiston, S. J. A coupled pressure-based co-located finite-volume solution method for natural-convection flows. *Numerical Heat Transfer, Part B: Fundamentals 61*, 2 (2012), 91–115.

[66] Van Doormaal, J. P. and Raithby, G. D. Enhancements of the simple method for predicting incompressible fluid flows. *Numerical Heat Transfer 7*, 2 (1984), 147–163.

[67] Vanka, S. Block-implicit multigrid solution of navier-stokes equations in primitive variables. *Journal of Computational Physics 65*, 1 (1986), 138 – 158.

[68] Yu, B., Tao, W.-Q., Wei, J.-J., Kawaguchi, Y., Tagawa, T. and Ozoe, H. Discussion on momentum interpolation method for collocated grids of incompressible flow. *Numerical Heat Transfer, Part B: Fundamentals 42*, 2 (2002), 141–166.

[69] Zhang, S., Zhao, X. and Bayyuk, S. Generalized formulations for the rhie–chow interpolation. *Journal of Computational Physics 258*, 0 (2014), 880 – 914.