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 January 17, 2015

(F. Gabel)

Contents

List of Figures

List of Tables

1 Introduction

This thesis is about.

Figure 1: Comparison of vertex oriented and cell center oriented variable arrangement

3 Finite Volume Method for Incompressible Flows – Theoretical Basics

This section deals with the fundamentals of the numerical solution via a finite volume method of the formerly presented set of partial differential equations. The focus of this section is, to provide an overview over the methods to be used in the present thesis. The information contained in this section is based on (Peric,Schäfer,Muzaferja,Jsak). The overview starts by mentioning the different grid types to be used and the discretization techniques to be applied. On the basis of integral formulations of the equations to be solved, the therein contained integrals and differential operators have to be discretized. Since the accuracy of the default concepts for discretizing differential operators degrades with decreasing grid quality, this chapter furthermore presents different approaches to handle corrections for cases in which the cause of degrading grid quality is increased non-orthogonality.

The goal of the finite volume method is to provide algebraic equations which can be used to determine an approximate solution of a partial differential equation. This system of linear algebraic equations can be solved by means of algorithms to be presented in the end of this section. However since the Navier-Stokes equations are in general non-linear an intermediate step has to be taken, by linearizing the discrete equations. This leads to the need of an iteration process, the *Picard iteration*, which will be explained briefly.

3.1 Numerical Grid

In this subsection a brief overview of the general grid structure to be used in the present thesis is given. The main idea behind finite volume methods is to solve partial differential equations by integrating them over the specified continuous problem domain and dividing this domain into a finite number of subdomains, the so called control volumes. The result of the this finite partition of a continuous problem domain is called the numerical grid. The grid consists of a finite number of grid cells which represent the boundaries of a discrete domain of integration. Depending on whether the numerical solution of an equation is to be calculated on the boundary vertices of grid cell or in the center of the cell, the variable arrangement is denoted to be vertex or cell center oriented. As the methods of employed in the present thesis are intended to be generally applicable to complex geometries the cell centered approach offers more flexibility (3.1). DONT CONFUSE WITH STAGGERED AND COLLOCATED ARRANGEMENT.

Regarding the treatment of domain boundaries and the ordering of the cells within the problem domain different types of numerical grids can be distinguished. The present thesis makes use of so called block structured grids with hexahedron cells. A structured grid is characterized by a constant amount of of grid cells in each coordinate direction. The high regularity of structured grids benefits the computational efficiency of algorithms to be used on this type of grid. A block structured grid consists of different grid blocks of which each considered individually is structured, but if the topology of the grid is considered it is unstructured. An example of a block structured grid with distinguishable grid blocks is given in figure 3.1. The use of block structured grids is motivated by the need to increase the adaptivity of structured grids by maintaining high computational efficiency. Furthermore it naturally embraces the concept of domain decomposition which facilitates the implementation of parallel algorithms for the decomposed computational domain.

Inside a structured grid block, cells with the shape of hexahedrons are used. In addition to the geometric boundaries of each control volume a numerical grid also provides a mapping that assigns to each control volume with index P a set of indexes of neighbouring control volumes $NB(P) := \{W, S, B, T, N, E\}$, which are named after the geographic directions. Figure 3.1 shows a single grid cell with its direct neighbours. The faces $\{S_w, S_s, S, b, S_t, S_n, S_e\}$ of each hexahedral control volume represent the mentioned geometric boundaries.

- talk about grid quality
- talk about local refinement

- talk about variable arrangement

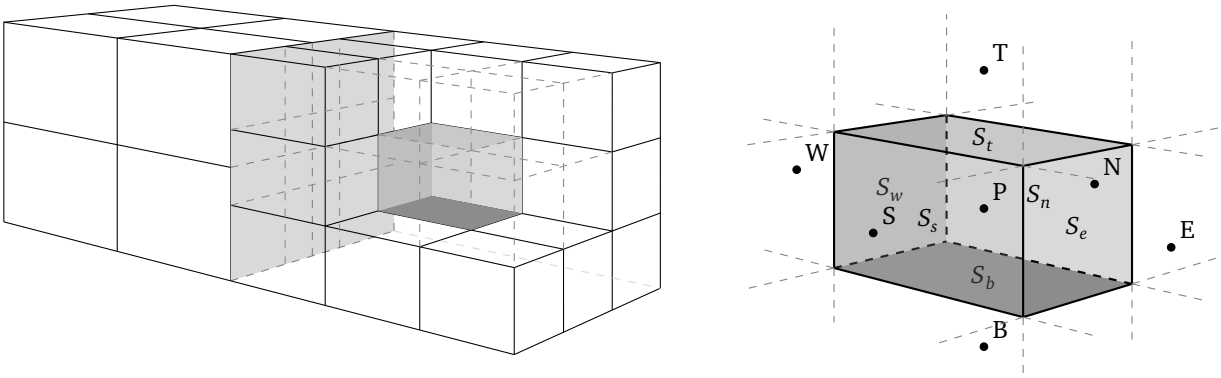

Figure 2: Block structured grid consisting of two blocks

3.2 Approximation of Integrals and Derivatives

In the course of transforming a partial differential equation into a system of linear algebraic equations, integrals and derivatives have to be approximated. The simplest method for approximating an integral is by using the *midipoint rule*. This rule is similar to the mean value theorem of integration, which states that there exists a point $\xi \in V$ for a Riemann integrable function ϕ such that $\phi(\xi)\int_V \mathrm{d}V = \int_V \phi(x)\mathrm{d}V$. For the midpoint rule ξ is taken to be the center of mass of V. If the integration domain V is indeed a Volume, fortunately the calculation of $\phi(\xi)$ with $\xi := \left(\int_V x_i \mathrm{d}V / \int_V \mathrm{d}V\right)_{i=1,\dots,3}$ presents no difficulties since due to the collocated variable arrangement the value of ϕ is stored in the cell center, which corresponds to the location ξ. However if the domain of integration is a surface, a preceding interpolation step is necessary.

On the other hand to transform a partial differential equation into a linear algebraic equation it is necessary to discretize the differential operators of the equations. For numerical reasons two different discretization techniques are used in this thesis. A common task is to discretize expressions of the form

$$(\nabla\phi)_e \cdot \mathbf{n}_e,$$

where $(\nabla\phi)_e$ is the Gradient of ϕ on a boundary face S_e. One method is to directly interpret this expression as a directional derivative and approximate it with a central difference

$$(\nabla\phi)_e \cdot \mathbf{n}_e \approx \frac{\phi_P - \phi_E}{||\mathbf{x}_P - \mathbf{x}_E||_2}. \tag{14}$$

Another method would be to first calculate the cell center gradients $(\nabla\phi)_P$ and $(\nabla\phi)_E$ and interpolate them linearly before calculating the projection onto $\mathbf{n}_e$

$$(\nabla\phi)_e \cdot \mathbf{n}_e \approx \left[\gamma_e (\nabla\phi)_P + (1-\gamma_e)(\nabla\phi)_E\right] \cdot \mathbf{n}_e,$$

where $\gamma_e := ||\mathbf{x}_P - \mathbf{x}_e||_2 / ||\mathbf{x}_P - \mathbf{x}_E||_2$ is a geometric interpolation factor. For calculating the cell center gradients a method based on Gauss' integration theorem and the midpoint rule for volume integration is employed

$$(\nabla\phi)_{i,P} = \left(\frac{\partial\phi}{\partial x_i}\right)_P \approx \frac{\int_V \left(\frac{\partial\phi}{\partial x_i}\right)_P \mathrm{d}V}{|V|}. \tag{15}$$

Briefly explain the idea behind the quadrature via the midpoint rule. Talk about central differences and the approximation via the gauss theorem. Maybe talk about the resulting order of the truncation error.

3.3 Treatment of Non-Orthogonality of Grid Cells

Unfortunately real applications involve complex geometries, which in turn affects the orthogonality of the grid. On non-orthogonal meshes the directional derivative in direction of the face normal unit vector $\mathbf{n}_e$ can no longer be approximated as in (14). On the other side the exclusive usage of (15) is not desirable due to the bigger truncation error that comes with this approximation (PROOF?). Hence a compromise is made and the surface vector $\mathbf{S}_e := S_e \mathbf{n}_e$ is decomposed as

$$\mathbf{S}_e = \Delta + \mathbf{k}, \tag{16}$$

where Δ is parallel to the vector $\mathbf{d}_e := (\mathbf{x}_E - \mathbf{x}_P)$ that directly connects the center of the control volume P with the center of its neighbour E. This vector controls the *orthogonal* contribution to the directional derivative. The vector $\mathbf{k}$ controls the influence of the *non-orthogonal* contribution. In the next paragraphs the three main decompositions of the surface vector $\mathbf{S}_e$ will be presented by stating the respective expression for Δ. The resulting vector $\mathbf{k}$ can be calculated by using (16). One important characteristic that all of the presented approaches have is common is that the non-orthogonal contribution vanishes as expected, when an orthogonal grid is used. For simplicity the presentation of the decompositions is chosen to be two dimensional. An geometrical interpretation of the tree approaches is given in 3.3.3. The last subsection handles the integration of one generic approach into the discretization process.

3.3.1 Minimum Correction Approach

This is the approach as proposed in Muzaferja. The reader should note, that even though Ferziger/Peric reference the work of Muzaferja they use a different approach to be presented in the next paragraph. This method is designed to keep the non-orthogonal contribution minimal by always choosing $\mathbf{k}$ to be orthogonal to Δ, which leads to

$$\Delta = (\mathbf{d} \cdot \mathbf{S}_e) \frac{\mathbf{d}}{||\mathbf{d}||_2}.$$

It should be noted that the Influence of the orthogonal contribution decreases with increasing non-orthogonality of the grid.

3.3.2 Orthogonal Correction Approach

The following method for decomposing the surface normal vector is presented in Ferziger/Peric and the approach implemented in the developed solvers. In this approach a simple projection is used which is independent of the non-orthogonality of the grid. As a result the orthogonal contribution $||\Delta||_2 = ||\mathbf{S}_e||_2$ and is thus modelled as

$$\Delta = S_e \frac{\mathbf{d}}{||\mathbf{d}||_2}.$$

3.3.3 Over-Relaxed Approach

The last approach is used in Jsak and Darwish and is characterized by an increasing influence of the orthogonal contribution with increasing grid non-orthogonality, as opposed to the minimum correction approach. The orthogonal contribution is calculated as

$$\Delta = S_e^2 \frac{\mathbf{d}}{\mathbf{d} \cdot \mathbf{S}_e}.$$

3.3.4 Deferred Non-Orthogonal Correction

In order to reduce the computational stencil that would be necessary to handle the non-orthogonal correction implicitly the correction will be treated explicitly using a deferred correction which guarantees that in the case of a fully converged solution only the face normal derivative has been taken into account. Generally the discretization using a non-orthogonal correction would yield

$$(\nabla \phi)_e \cdot \mathbf{S}_e \approx (\nabla \phi)_e \cdot \Delta + (\nabla \phi)_e \cdot \mathbf{k}.$$

Where the first term can be approximated using a central differencing scheme for the directional derivative and the second by interpolating cell center gradients. If one furthermore uses the fact that this method comes to play in an solution algorithm for a non-linear system of partial differential equations a deferred correction can be implemented which ensues a smaller error from the non-orthogonality. In the case of the previously mentioned discretization techniques for partial derivatives a possible deferred correction approach reads

$$(\nabla \phi)_e \cdot \mathbf{S}_e \approx ||\Delta||_2 \frac{\phi_P - \phi_E}{||x_P - x_E||_2} - \left[(\nabla \phi)_e \cdot (\Delta - \mathbf{k})\right]^{(n-1)}.$$

It should be noted that the use of a deferred correction in conjunction with the requirement that the non-orthogonal correction vanishes on orthogonal grid introduces a inconsistent discretization of $(\nabla \phi)_e \cdot \Delta$.

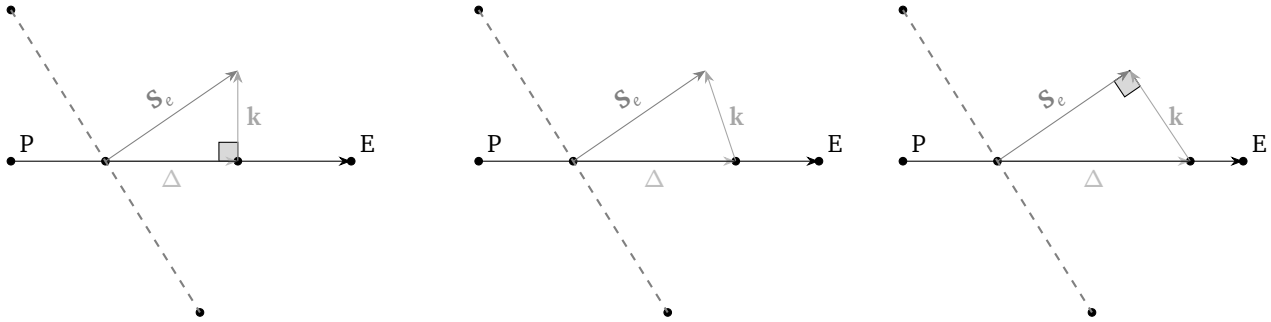

Figure 3: Minimum correction, orthogonal correction and over-relaxed approach

3.4 Numerical Solution of Non-Linear Systems – Linearization Techniques

Introduce the concept of the Picard-Iteration as a linearization technique. Introduce the notions of inner and outer iterations. Refer to later chapters when it comes to deferred correction.

3.5 Numerical Solution of Linear Systems

3.5.1 Stone's SIP Solver

Basic Idea as in Schäfer or Peric. Emphasize that this is a very problem specific approach, that cannot be generalized that easily in opposition to the general purpose linear solvers from PETSc. Present ether BiCGStab or GMRES, the one which performs better and is used throughout the thesis.

3.5.2 Krylov Subspace Methods

- General concept of cyclic vector spaces of $\mathbb{R}^n$,
- talk about bases of krylov subspaces and the arnoldi algorithm, talk about polynomials and linear combinations
- mention the two major branches (minimum residual approach, petrov and ritz-galerkin approach)
- name some representative ksp algorithms, importance of preconditioning, not as detailed as in bachelor thesis
- in cases there is a nonempty Nullspace what happens?

4 Finite Volume Method for Incompressible Flows – Segregated Approach

The purpose of this section is to present the discretization applied to the set of equations (**??**). The applied discretization techniques depend on the different terms of each equation, thus at first every equation will be discretized individually. The finite volume method relies on the discretization of integral equations, which will be derived at the beginning of each subsection that relies on them. Since the system of partial differential equations to be solved always exhibits coupling at least between the dependent variables pressure and velocity a first solution algorithm, namely the *SIMPLE* algorithm addressed to resolve the pressure velocity coupling is introduced. The efficient coupling of the Navier-Stokes equations to the temperature equation is one part of the present thesis and will be addressed in a separate subsection. Furthermore every problem modelled by partial differential equations needs to provide valid boundary conditions. The discretization of those boundary conditions, that are relevant for the present thesis will be presented in their own subsection.

4.1 Discretization of the Mass Balance

Integration of equation (11) over the integration domain of a single control volume P yields after the application of Gauss' integration theorem and the additivity of the Riemann integral

$$\iint_S u_i n_i \mathrm{d}S = \sum_{f\in\{w,s,b,t,n,e\}} \iint_{S_f} u_i n_i \mathrm{d}S = 0$$

In the present work the mass balance is discretized using the midpoint rule for the surface integrals and linear interpolation of the velocity to to center of mass of the surface. This leads to the following form of the mass balance:

$$\begin{aligned}\sum_{f\in\{w,s,b,t,n,e\}} u_{i_f} n_{f_i} S_f &= u_{i_w} n_{w_i} S_w + u_{i_e} n_{e_i} S_e + u_{i_s} n_{s_i} S_s + u_{i_n} n_{n_i} S_n + u_{i_b} n_{b_i} S_b + u_{i_t} n_{t_i} S_t \\ &= (\gamma_w u_{i_W} + (1-\gamma_w) u_{i_P}) n_{w_i} S_w + (\gamma_s u_{i_S} + (1-\gamma_s) u_{i_P}) n_{s_i} S_s \\ &\quad + (\gamma_b u_{i_B} + (1-\gamma_b) u_{i_P}) n_{b_i} S_b + (\gamma_t u_{i_T} + (1-\gamma_t) u_{i_P}) n_{t_i} S_t \\ &\quad + (\gamma_n u_{i_N} + (1-\gamma_n) u_{i_P}) n_{n_i} S_n + (\gamma_e u_{i_E} + (1-\gamma_e) u_{i_P}) n_{e_i} S_e \\ &= 0,\end{aligned}$$

where γ_f for $f \in \{w, e, s, n, b, t\}$ is the geometrical interpolation factor.

4.2 Discretization of the Momentum Balance

The stationary momentum balance integrated over a single control volume P reads as

$$\underbrace{\iint_S (\rho u_i u_j) n_j \mathrm{d}S}_{\text{convective term}} - \underbrace{\iint_S \left(\mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right) n_j \mathrm{d}S}_{\text{diffusive term}} = -\overbrace{\iiint_V \frac{\partial p}{\partial x_i}\mathrm{d}V}^{\text{sourceterm pressure}} - \overbrace{\iiint_V \rho\beta\,(T-T_0)\,\mathrm{d}V}^{\text{sourceterm temperature}}$$

where the different terms to be addressed individually in the following sections are indicated. Note that the form of this equation has been modified by using Gauss' integration theorem The terms residing on the left will be treated in an implicit manner whereas the terms on the right will be treated explicitly.

4.2.1 Calculation of Mass Flux – Rhie-Chow Interpolation

4.2.2 Linearization and Discretization of the Convective Term

The convective term $\rho u_i u_j$ of the Navier-Stokes equations is the reason for the non-linearity of the equations. In order to deduce a set of linear algebraic equations from the Navier-Stokes equations this term has to be linearized. As introduced in section (REFERENCE), the non linearity will be dealt with by means of an iterative process, the Picard iteration. The part dependent of the non dominant dependent variable therefore will be approximated by its value from the previous iteration as $\rho u_i^{(n)} u_j^{(n)} \approx \rho u_i^{(n)} u_j^{(n-1)}$. However this linearization will not be directly visible because it will be covered by the mass flux $\dot{m}_f = \iint_{S_f} \rho u_j^{(n-1)} n_j \mathrm{d}S$. Using the additivity of the Riemann integral the first step is to decompose the surface integral into individual contributions from each boundary face of the control volume P

$$\iint_S (\rho u_i u_j) n_j \mathrm{d}S = \sum_{f\in\{w,s,b,t,n,e\}} \iint_{S_f} \rho u_i u_j n_j \mathrm{d}S = \sum_{f\in\{w,s,b,t,n,e\}} F^c_{i,f}$$

where $F^c_{i,f} := \iint_{S_f} \rho u_i^{(n)} u_j^{(n-1)} n_j \mathrm{d}S$ is the convective flux of the velocity u_i through the face S_f.

To improve diagonal dominance of the resulting linear system while maintaining the smaller discretization error of a higher order discretization, a blended discretization scheme is applied using a deferred correction. Since due to the non-linearity of the equations to be solved an iterative solution process is needed by all means, the overall convergence doesn't degrade noticeably when using a deferred correction. Blending and deferred correction result in a decomposition of the convective flux into a lower order approximation that is treated implicitly and the explicit difference between the higher and lower order approximation for the same convective flux. Since for coarse grid resolutions the use of higher order approximations may lead to oscillations of the solution which may degrade or even impede convergence, the schemes can be blended by a control factor $\eta \in [0,1]$. To show the generality of this approach all further derivations are presented for the generic boundary face S_f that separates control volume P from its neighbour $F \in NB(P)$. This decomposition then leads to

$$F^c_{i,f} \approx \underbrace{F^{c,l}_{i,f}}_{\text{implicit}} + \eta \underbrace{\left[F^{c,h}_{i,f} - F^{c,l}_{i,f}\right]^{(n-1)}}_{\text{explicit}}.$$

Note that the convective fluxes carrying a l or h as exponent, already have been linearized and discretized. The discretization applied to the convective flux in the present work is using the midpoint integration rule and blends the upwind interpolation scheme with the linear interpolation scheme. Applied to above decomposition one can derive the following approximations

$$\begin{aligned} F^{c,l}_{i,f} &= u_{i,F} \min(\dot{m}_f, 0) + u_{i,P} \max(0, \dot{m}_f) \\ F^{c,h}_{i,f} &= u_{i,F} \gamma_e + u_{i,P}(1-\gamma_f), \end{aligned}$$

where the variable values have to be taken from the previous iteration step $(n-1)$ as necessary and the mass flux $\dot{m}_e$ has been used as result of the linearization process. The results can now be summarized by presenting the convective contribution to the matrix coefficients a_{F,u_i} and a_{P,u_i} and the right hand side b_{P,u_i} which are calculated as

$$a^c_{F,u_i} = \min(\dot{m}_f, 0), \quad a^c_{P,u_i} = \sum_{F \in NB(P)} \max(0, \dot{m}_f) \tag{17}$$

$$\begin{aligned} b^c_{P,u_i} = \sum_{F \in NB(P)} & \eta \left(u^{(n-1)}_{i,F} \left(\min(\dot{m}_f, 0) - \gamma_e \right) \right) \\ & + \eta \left(u^{(n-1)}_{i,P} \left(\max(0, \dot{m}_f) - (1-\gamma_e) \right) \right) \end{aligned} \tag{18}$$

4.2.3 Discretization of the Diffusive Term

The diffusive term contains the first partial derivatives of the velocity as result of the material constitutive equation that characterizes the behaviour of Newtonian fluids. As pointed out in section 3.3 directional derivatives can be discretized using central differences on orthogonal grids or in the more general case of non-orthogonal grids using central differences implicitly and a explicit deferred correction comprising the non-orthogonality of the grid. As seen in section **??** the diffusive term of the Navier-Stokes equations can be simplified using the mass balance in the case of an incompressible flow with constant viscosity μ. To sustain the generality of the presented approach this simplification will be omitted.

As before, by using the additivity and furthermore linearity of the Riemann integral, the integration of the diffusive term will be divided into integration over individual boundary faces S_f

$$\iint_S \left(\mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right) n_j \mathrm{d}S \mathrm{d}S = \sum_{f \in \{w,s,b,t,n,e\}} \left[\iint_{S_f} \mu \underline{\frac{\partial u_i}{\partial x_j}} n_j \mathrm{d}S + \iint_{S_f} \mu \frac{\partial u_j}{\partial x_i} n_j \mathrm{d}S \right] = \sum_{f \in \{w,s,b,t,n,e\}} F^d_{i,f},$$

where $F^d_{i,f}$ denotes the diffusive flux through an individual boundary face. Section 3.3 only covered the non-orthogonal corrector for directional derivatives. Since the velocity is a vector field and not a scalar field, the results of section 3.3 may only be applied to the underlined term. The other term will be treated explicitly since it is considerably smaller than the underlined term and does not cause oscillations and thus will not derogate convergence.

Using central differences for the implicit discretization of the directional derivative and furthermore using the *orthogonal correction* approach from 3.3.2

4.2.4 Discretization of the Source Term

4.2.5 Assembly of Linear Systems – Final Form of Equations

Coefficients of matrices for momentum are identical except in case of different factors for under-relaxation (underrelaxation (Andersson))(when does this happen) for the main diagonal coefficient. Small example in code, then show image of assembled system.

4.3 Discretization of the Generic Transport Equation

4.4 The SIMPLE-Algorithm

4.4.1 Pressure Correction Equation

4.4.2 Characteristic Properties of Projection Methods

Under-relaxation, slow convergence, inner iterations outer iterations, relative tolerances, also talk about staggered and collocated variable positioning

4.4.3 Dependence on Under-Relaxation – The Pressure-Weighted Interpolation Method

Present an approach for Under-Relaxation independent converged solution. Conduct the proof to show it really works. Present the results for different under-relaxation factors

4.4.4 Coupling of Temperature Equation

Explicit coupling through source term in momentum balances (Boussinesq-Approximation)

4.5 Boundary Conditions on Domain and Block Boundaries

Introduce chapter by talking about the nature of partial differential equations (Hackbusch). Always start with a simple implementation for the generic transport equation, then specialize to Navier-Stokes equation.

4.5.1 Dirichlet Boundary Condition

Only talk about dirichlet for velocities not for pressure.
Problematics of outlet boundary conditions

4.5.2 Wall Boundary Condition

Note that there are different approaches. Explain which approach is used and why (memory efficiency)

4.5.3 Block Boundary Condition

Relevant for block structured grids as for the validity of the domain composition.

6 CAFFA Framework

6.1 PETSc Framework

Keep in mind not to copy the manual

6.1.1 About PETSc

Bell Prize, MPI Programming

6.1.2 Basic Data Types

Vec,Mat (Different Matrix Types and Their effect on complex methods)

6.1.3 KSP and PC Objects and Their Usage

Singularities

6.1.4 Profiling

PETSc Log

6.1.5 Common Errors

Optimization, Interfaces, (ROWMAJOR,COLUMNMAJOR), Compiler Errors not helpful, Preallocation vs. Mallocs

6.2 Grid Generation and Conversion

Generation of block structured locally refined grids with non-matching block interfaces, neighbouring relations are represented by a special type of boundary conditions; Random number generator to move grid points within a epsilon neighbourhood while maintaining the grid intact. Show in a graph how preallocation impacts on runtime.

6.3 Preprocessing

Matching algorithm – the idea behind clipper and the used projection technique; alt.: Opencascade. Efficient calculation of values for discretization. Important for dynamic mesh refinement, arbitrary polygon matching, parallelizable due to easier interface

6.4 Implementation Details of CAFFA

6.4.1 MPI Programming Model

Basic idea of distributed memory programming model, emphasize the differences to shared memory model. Have a diagram at hand that shows how CAFFA sequentially works (schedule) and point out the locations where and of which type (global reduce, etc.) communication is, or when synchronization is necessary.

- after each solve
- pressure reference
- error calculation
- gradient calculation

Point out that one should try to minimize the number of this points such that parallel performance stays high. Better to calculate Velocity and Pressure Gradients at once not by seperately calling this routine.

6.4.2 Convergence Control

Explain how the criterion for convergence is met

6.4.3 Modi of Calculation

there are different modi of calculation, (NS segregated, then scalar; NS and Scalar Segregated; NS coupled and Scalar segregated; Fully coupled (wath out with fully coupled, this term seems to have already another meaning)). Note that for comparison of solvers it is crucial to develop programs on the same basis. This establishes comparability.

6.4.4 Indexing of Variables and Treatment of Boundary Values

Describe MatZeroValues and how it is used to simplify the code. Also loose a word on PCREDISTRIBUTE its advantages and downsides. Compliance of PETSc zero based indexing and CAFFA indexing which considers boundary values. Problems with boundary entries

6.4.5 Field Interlacing

Realization through special arrangement of variables and the use of index sets (subvector objects) and/or preprocessor directives. Advantages (there was a paper I cited in my thesis). Note that not all variables are interlaced (Velocities are, but their gradients are not). Great impact on Matrix structure.

6.4.6 Domain Decomposition, Exchange of Ghost Values and Parallel Matrix Assembly

- Ghost values are stored in local representations of the global vector (state the mapping for those entries).
- Matrix coefficients are calculated on one processor and sent to the neighbour.
- Preallocation as crucial aspect for program performance. For the coupled system the matrix is assembled in a 2-3 step process to save memory for coefficients.
- Present a simple method for balancing the matrix related load by letting PETSc take care of matrix distribution.
- Use Spy function of Matlab to visualize the sparse matrices. Point out advantages of calculating coefficients for the neighbouring cells locally (no need to update mass fluxes, geometric data doesn't need to be shared, small communication overhead since processors assemble matrix parts that don't belong to them (visualize)).
- Paradigm: Each time new information is available perform global updates. Advantages of using matrices: Show structure of matrix when using arbitrary matching vs. higher memory requirements vs. better convergence

6.5 Postprocessing

Visualization of Results with Paraview and Tecplot Export matrices as binaries and visualize them using matlab scripts.

7 Verification of CAFFA

Different parts, describe incremental approach, only present final results. Refer to next section for Validation of CAFFA

7.1 Theoretical Discretization Error

present the Taylor-Series Expansion

7.2 Method of Manufactured Solutions

basically sum up the important points of salari's technical report, symmetry of solution/domain/grid is bad point out that mms is not able to detect errors in the physical model Also loose a word or two about discontinuous manufactured solutions

7.3 Exact and Manufactured Solutions for the Navier-Stokes Equations and the Energy Equation

Not always there is an exact solution. Divergence free approach. Presentation of the used manufactured solution. What if solution is not divergence free? Derivation of equations and modifications to continuity equation. analyze the problem of too complicated manufactured solutions. also use temperature dependent density function. Explain why global mass conservation in a discrete sense is important and how it can be achived. Special domain, vanishing manufactured solution or symmetric manufactured solutions if a higher approximation of boundary fluxes is not feasible.

- `http://scicomp.stackexchange.com/questions/6943/manufactured-solutions-for-incompressible-navier-stokes`

- http://link.springer.com/article/10.1007/BF00948290
- http://physics.stackexchange.com/questions/60476/exact-solutions-to-the-navier-stokes-equations
- http://www.annualreviews.org/doi/pdf/10.1146/annurev.fl.23.010191.001111

7.4 Measurement of Error and Calculation of Order

Different error measures (L2-Norm,completeness of function space, consistency etc.)

7.4.1 Testcase on Single Processor on Orthogonal Locally Refined Grid

7.4.2 Testcase on Multiple Processors on Non-Orthogonal Locally Refined Grid

Give a measure of the grid quality.

8 Comparison of Solver Concepts

8.1 Convergence Behaviour on Locally Refined Block Structured Grids with Different Degrees of Coupling

Show how the implicit treatment of block boundaries maintains (high) convergence rates. Plot Residual over number of iterations.

8.2 Parallel Performance

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

- Networking
- Mem Section and processes in between islands (calculating across islands)
- Versioning information (PETSc,INTEL COMPILERS,CLIPPER,MPI IMPLEMENTATION,BLAS/LAPACK)
- Software not designed to perform well on desktop PCs.

8.2.2 Measures of Performance

- Maße definieren
- Nochmal Hager,Wellein studieren
- Guidelines for measuring performance (bias through system processes or user interaction), only measure calculation time do not consider I/O in the beginning and the end
- Cite Schäfer and Peric with their different indicators for parallel efficiency, load balancing and numerical efficiency

8.2.3 Preliminary Upper Bounds on Performance – The STREAM Benchmark

Pinning of processes (picture), preliminary constraints by hardware and operating systems, identification of bottlenecks and explain possible workarounds, history and results of STREAM. Bandwidth as Bottleneck, how to calculate a Speedup estimate based on the measured bandwidth. PETSc Implementation of STREAM

8.2.4 Discussion of Results for Parallel Efficiency

8.2.5 Speedup Measurement for Analytic Test Cases

8.3 Test Cases with Varying Degree of Non-Linearity

As Peric says I want to prove that the higher the non-linearity of NS, the better relative convergence rates can be achieved with a coupled solver. Fi

8.3.1 Transport of a Passive Scalar – Forced Convection

8.3.2 Buoyancy Driven Flow – Natural Convection

8.3.3 Flow with Temperature Dependent Density – A Highly Non-Linear Test Case

Maybe I could consider two test cases, one with oscillating density and one with a quadratic polynomial. Interesting would be also to consider the dependence of convergence on another scalar transport equation

8.4 Realistic Testing Scenarios – Benchmarking

Also consider simple load balancing by distributing matrix rows equally

8.4.1 Flow Around a Cylinder 3D – Stationary

Describe Testing Setup (Boundary conditions and grid). Present results and compare them with literature.

8.4.2 Flow Around a Cylinder 3D – Instationary

- `http://www.featflow.de/en/benchmarks/cfdbenchmarking/flow/dfg_flow3d/dfg_flow3d_configuration.html`

Describe Testing Setup (Boundary conditions and grid). Present results and compare them with literature.

8.4.3 Heat-Driven Cavity Flow

- `http://www.featflow.de/en/benchmarks/cfdbenchmarking/mit_benchmark.html`

Describe Testing Setup (Boundary conditions and grid). Present results and compare them with literature.

8.5 Realistic Testing Scenario – Complex Geometry

9 Conclusion and Outlook

Turbulence (turbulent viscosity has to be updated in each iteration), Multiphase (what about discontinuities), GPU-Accelerators, Load-Balancing, dynamic mesh refinement, Counjugate Heat Transfer with other requirements for the numerical grid, grid movement, list some papers here) Identify the optimal regimes / conditions for maximizing performance. Each solver concept has its strengths and weaknesses. Try other variants of Projection Methods like SIMPLEC, SIMPLER, PISO or PIMPLE (OpenFOAM)

References

[1] 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.

[2] 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.

[3] 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.

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

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

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

[12] 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.

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

[14] 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.

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

[16] 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.

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

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

[19] 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.

[20] 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.

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

[22] 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.

[23] 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.

[24] 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.

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