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 20. Januar 2015

(F. Gabel)

Contents

1 Introduction **3**

2 Finite Volume Method for Incompressible Flows – Theoretical Basics **4**
2.1 Numerical Grid . . . 4
2.2 Approximation of Integrals and Derivatives . . . 5
2.3 Treatment of Non-Orthogonality of Grid Cells . . . 6
2.3.1 Minimum Correction Approach . . . 6
2.3.2 Orthogonal Correction Approach . . . 6
2.3.3 Over-Relaxed Approach . . . 6
2.3.4 Deferred Non-Orthogonal Correction . . . 6
2.4 Numerical Solution of Non-Linear Systems – Linearization Techniques . . . 7
2.5 Numerical Solution of Linear Systems . . . 7
2.5.1 Stone's SIP Solver . . . 7
2.5.2 Krylov Subspace Methods . . . 7

3 Implicit Finite Volume Method for Incompressible Flows – Segregated Approach **8**
3.1 Calculation of Mass Fluxes and the Pressure-Weighted Interpolation Method . . . 8
3.2 Implicit Pressure Correction and the SIMPLE Algorithm . . . 9
3.3 Discretization of the Momentum Balance . . . 11
3.3.1 Linearization and Discretization of the Convective Term . . . 11
3.3.2 Discretization of the Diffusive Term . . . 12
3.3.3 Discretization of the Source Terms . . . 13
3.4 Discretization of the Pressure Correction Equation . . . 13
3.5 Discretization of the Temperature Equation . . . 13
3.6 Boundary Conditions . . . 14
3.7 Structure of the Assembled Linear Systems . . . 14

4 Conclusion and Outlook **15**

References **16**

List of Figures

1 Comparison of vertex oriented and cell center oriented variable arrangement . . . 4
2 Block structured grid consisting of two blocks . . . 5
3 Minimum correction, orthogonal correction and over-relaxed approach . . . 7

List of Tables

List of Algorithms

1 SIMPLE Algorithm . . . 11

1 Introduction

This thesis is about.

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

2 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.

2.1 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 (2.1). 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 2.1. 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 2.1 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

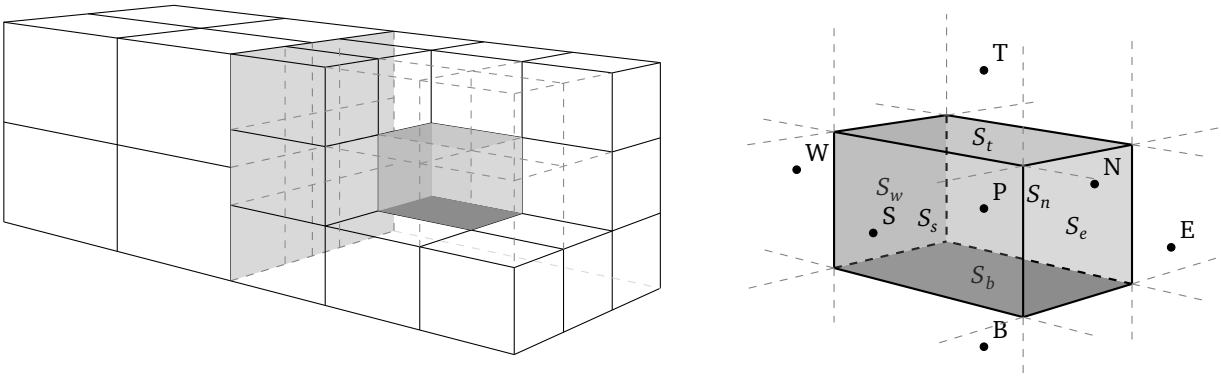

Figure 2: Block structured grid consisting of two blocks

2.2 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

$$(\nabla\phi)_e \cdot \mathbf{n}_e \approx \frac{\phi_P - \phi_E}{||\mathbf{x}_P - \mathbf{x}_E||_2}. \tag{1}$$

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 $\mathbf{n}_e$

$$(\nabla\phi)_e \cdot \mathbf{n}_e \approx \left[\gamma_e(\nabla\phi)_P + (1-\gamma_e)(\nabla\phi)_E\right] \cdot \mathbf{n}_e, \tag{2}$$

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

$$(\nabla\phi)_{i,P} = \left(\frac{\partial\phi}{\partial x_i}\right)_P \approx \frac{\int_V \left(\frac{\partial\phi}{\partial x_i}\right)_P \mathrm{d}V}{|V|}. \tag{3}$$

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.

2.3 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 (1). On the other side the exclusive usage of (3) 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

$$\mathbf{S}_e = \mathbf{\Delta} + \mathbf{k}, \tag{4}$$

where $\mathbf{\Delta}$ 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 $\mathbf{\Delta}$. The resulting vector $\mathbf{k}$ can be calculated by using (4). 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 2.3.3. The last subsection handles the integration of one generic approach into the discretization process.

2.3.1 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 $\mathbf{\Delta}$, which leads to

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

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

2.3.2 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 $||\mathbf{\Delta}||_2 = ||\mathbf{S}_e||_2$ and is thus modelled as

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

2.3.3 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

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

2.3.4 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

$$(\nabla \phi)_e \cdot \mathbf{S}_e \approx ||\Delta||_2 \frac{\phi_P - \phi_E}{||\mathbf{x}_P - \mathbf{x}_E||_2} - (\nabla \phi)_e^{(n-1)} \cdot (\Delta - \mathbf{S}_e).$$

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 an inconsistent discretization of $(\nabla \phi)_e \cdot \mathbf{\Delta}$.

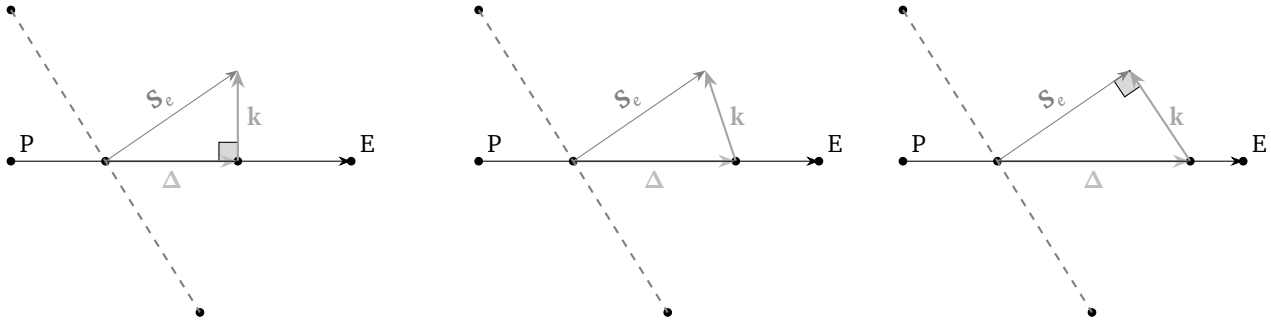

Figure 3: Minimum correction, orthogonal correction and over-relaxed approach

2.4 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.

2.5 Numerical Solution of Linear Systems

2.5.1 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.

2.5.2 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?

3 Implicit Finite Volume Method for Incompressible Flows – Segregated Approach

The purpose of this section is to present the discretization applied to the set of equations (**??**). Since the system of partial differential equations to be solved always exhibits a 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. Methods of calculating mass fluxes and the detailed derivation of all coefficients that result from the discretization process is presented. The discretization of those boundary conditions, that are relevant for the present thesis will be presented in their own subsection.

3.1 Calculation of Mass Fluxes and the Pressure-Weighted Interpolation Method

The advantages of using a cell-centered variable arrangement are evident: The treatment of non-orthogonality is simplified and the conservation property of finite volume methods is retained [5,17,18,27]. A major drawback if SIMPLE-Type algorithms, to be introduced in section 3.2, are used is that pressure field may delink which will then lead to unphysical oscillations in both the pressure and the velocity results. This . If the oscillations are severe enough the solution algorithm might get unstable and diverge. The described decoupling occurs, when the pressure gradient in the momentum balances and the mass fluxes in the continuity equation are discretized using central differences. A common practice to eliminate this behaviour is the use of a momentum interpolation technique, also known as *Rhie-Chow Interpolation* [21]. The original interpolation scheme however doesn't guarantee a unique solution, independent of the amount of under-relaxation. The performance of one of the algorithms that are used in the present thesis heavily relies on the under-relaxation of variables to accomplish stability. Furthermore the original method as proposed by [21] does not account for large body forces which also may lead to unphysical results. This issues will be addressed in this subsection which at the end will present an interpolation method that assures an under-relaxation independent solution, the *pressure-weighted interpolation method* [18].

Starting point of the pressure-weighted interpolation method are the discretized momentum balances at node P and the neighbouring node Q. The discretization for a finite volume methods and details including the incorporation of under-relaxation factors will be handled in subsection 3.3. The semi-discrete implicit momentum balances read if one solves for the velocity at node P or Q

$$u_{i,P}^{(n)} = -\frac{\alpha_{\mathbf{u}_P}}{a_{P,u_i}}\left(\sum_{F\in NB(P)} a_{F,u_i}u_{i,F}^{(n)} + b_{P,u_i}^{(n-1)} - V_P\left(\frac{\partial p^{(n-1)}}{\partial x_i}\right)_P\right) + (1-\alpha_{\mathbf{u}})u_{i,P}^{(n-1)} \tag{5a}$$

$$\text{and}\quad u_{i,Q}^{(n)} = -\frac{\alpha_{\mathbf{u}_Q}}{a_{Q,u_i}}\left(\sum_{F\in NB(Q)} a_{F,u_i}u_{i,F}^{(n)} + b_{Q,u_i}^{(n-1)} - V_Q\left(\frac{\partial p^{(n-1)}}{\partial x_i}\right)_Q\right) + (1-\alpha_{\mathbf{u}})u_{i,Q}^{(n-1)} \tag{5b}$$

where the superscript $(n-1)$ denotes the previous outer iteration number. The reader should note, that the pressure gradient has not been discretized yet. This has the advantage that the selective interpolation technique [**?**] can be applied, which is crucial for the elimination of the mentioned oscillations. In almost the same manner a semi-discrete implicit momentum can be formulated for a virtual control volume located between nodes P and Q (PICTURE)

$$u_{i,f}^{(n)} = -\frac{\alpha_{\mathbf{u}_f}}{a_{f,u_i}}\left(\sum_{F\in NB(f)} a_{F,u_i}u_{i,F}^{(n)} + b_{f,u_i}^{(n-1)} - V_f\left(\frac{\partial p^{(n-1)}}{\partial x_i}\right)_f\right) + (1-\alpha_{\mathbf{u}})u_{i,f}^{(n-1)} \tag{6}$$

To eliminate the artifacts surging form the virtualization of a control volume the following assumptions have to be made to derive a closed expression for the velocity on the boundary face S_f

$$\frac{\alpha_{\mathbf{u}_f}}{a_{f,u_i}}\left(\sum_{F\in NB(f)} a_{F,u_i}u_{i,F}^{(n)}\right) \approx (1-\gamma_f)\frac{\alpha_{\mathbf{u}_P}}{a_{P,u_i}}\left(\sum_{F\in NB(P)} a_{F,u_i}u_{i,F}^{(n)}\right) + \gamma_f\frac{\alpha_{\mathbf{u}_Q}}{a_{Q,u_i}}\left(\sum_{F\in NB(Q)} a_{F,u_i}u_{i,F}^{(n)}\right) \tag{7a}$$

$$\text{and}\frac{\alpha_{\mathbf{u}_f}}{a_{f,u_i}} \approx (1-\gamma_f)\frac{\alpha_{\mathbf{u}_P}}{a_{P,u_i}} + \gamma_f\frac{\alpha_{\mathbf{u}_Q}}{a_{Q,u_i}}, \tag{7b}$$

where γ_f is a geometric interpolation factor.

Using the assumptions made equation (7) the expression in equation (6) can be closed in a way that it only depends on the variable values in node P and Q.

$$
\begin{aligned}
u_{i,f}^{(n)} \approx \left(1-\gamma_f\right)&\left(-\frac{\alpha_{\mathbf{u}_P}}{a_{P,u_i}}\sum_{F\in NB(P)} a_{F,u_i}u_{i,F}^{(n)}\right)+\gamma_f\left(-\frac{\alpha_{\mathbf{u}_Q}}{a_{Q,u_i}}\sum_{F\in NB(Q)} a_{F,u_i}u_{i,F}^{(n)}\right)\\
&+\frac{\alpha_{\mathbf{u}_f}}{a_{f,u_i}}b_{f,u_i}^{(n-1)}-\frac{\alpha_{\mathbf{u}_f}}{a_{f,u_i}}V_f\left(\frac{\partial p^{(n-1)}}{\partial x_i}\right)_f+\left(1-\alpha_{\mathbf{u}}\right)u_{i,f}^{(n-1)}\\
=\left(1-\gamma_f\right)&u_{i,P}^{(n)}-\left(1-\gamma_f\right)\left(b_{Q,u_i}^{(n-1)}-V_Q\left(\frac{\partial p}{\partial x_i}\right)_Q^{(n-1)}\right)\\
&+\gamma_f u_{i,Q}^{(n)}-\gamma_f\left(b_{Q,u_i}^{(n-1)}-V_Q\left(\frac{\partial p}{\partial x_i}\right)_Q^{(n-1)}\right)\\
&+\frac{\alpha_{\mathbf{u}_f}}{a_{f,u_i}}b_{f,u_i}^{(n-1)}-\frac{\alpha_{\mathbf{u}_f}}{a_{f,u_i}}V_f\left(\frac{\partial p^{(n-1)}}{\partial x_i}\right)_f+\left(1-\alpha_{\mathbf{u}}\right)u_{i,f}^{(n-1)}\\
=\left[\left(1-\gamma_f\right)\right.&\left.u_{i,P}^{(n-1)}+\gamma_f u_{i,Q}^{(n-1)}\right]\\
&-\frac{\alpha_{\mathbf{u}}}{a_{f,u_i}}\left[\left(\frac{\partial p}{\partial x_i}\right)_f^{(n-1)}-\left(1-\gamma_f\right)\left(\frac{\partial p}{\partial x_i}\right)_P^{(n-1)}-\gamma_f\left(\frac{\partial p}{\partial x_i}\right)_Q^{(n-1)}\right]\\
&+(1-\alpha)\left[u_{i,f}^{(n-1)}-\left(1-\gamma_f\right)u_{i,P}^{(n-1)}-\gamma_f\, u_{i,Q}^{(n-1)}\right]
\end{aligned}
\tag{8}
$$

3.2 Implicit Pressure Correction and the SIMPLE Algorithm

The goal of finite volume methods is to deduce a system of linear algebraic equations from a partial differential equation. In the case of the momentum balances the general structure of this linear equations is

$$
a_{P,u_i}u_{i,P}^{(n)}+\sum_{F\in NB(P)} a_{F,u_i}u_{i,F}^{(n)}=b_{P,u_i}^{(n)}-\left(\frac{\delta p^{(n)}}{\delta x_i}\right)_P, \tag{9}
$$

where the pressure gradient has been discretized only symbolically and b_{P,u_i} denotes the source term. The symbolic discretization of the pressure gradient not only includes the discretization of the differential operator but, in the case a finite volume method is used, also the approximation of the integral. At this stage the equations are still coupled and non-linear. As described in section 2.4 the Picard iteration process will be used to linearize the equations. After this, every momentum balance equation will only depend on the one dominant variable u_i. Furthermore the coupling of the momentum balances through the convective term $(u_i u_j)$ is resolved in the process of linearization. The decoupled momentum balances will then be solved sequentially for the dominant variable. All coefficients $a_{\{P,F\},u_i}$, the source term and the pressure gradient will be evaluated explicitly by using results of the preceding outer iteration $(n-1)$ until the non-linear equations are fulfilled up to a certain tolerance (**??**). This leads to the following linear equation

$$
a_{P,u_i}u_{i,P}^{(n*)}+\sum_{F\in NB(P)} a_{F,u_i}u_{i,F}^{(n*)}=b_{P,u_i}^{(n-1)}-\left(\frac{\delta p^{(n-1)}}{\delta x_i}\right)_P. \tag{10}
$$

Here $(*)$ indicates that the solution of this equation still needs to be corrected to also fulfill the discretized mass balance

$$
\frac{\delta\left(\rho u_i^{(n)}\right)}{\delta x_i}=0. \tag{11}
$$

The lack of an equation with the pressure as dominant variable leads to the necessity to alter the mass balance as the only equation left. Methods of this type are called projection methods. A common class of algorithms of this family of

methods uses an equation for the additive pressure correction p' instead of the pressure itself and enforces continuity by correcting the velocities with an additive corrector u_i':

$$u_{i,P}^{(n)} = u_{i,P}^{(n*)} + u_{i,P}', \quad p_P^{(n)} = p_P^{(n-1)} + p_P'.$$

It is now possible to formulate the discretized momentum balance for the corrected velocities and the corrected pressure as

$$a_{P,u_i} u_{i,P}^{(n)} + \sum_{F \in NB(P)} a_{F,u_i} u_{i,F}^{(n)} = b_{P,u_i}^{(n-1)} - \left(\frac{\delta p^{(n)}}{\delta x_i} \right)_P . \tag{12}$$

It should be noted that the only difference between (12) and (9) is that the source term b_{P,u_i} has not been updated yet. To couple velocity and pressure correctors one can subtract equations (9) and (10) and consider the linearity of the discretization operator $\frac{\delta}{\delta x_i}$

$$u_{i,P}' = -\underline{\frac{\sum_{F \in NB(P)} a_{F,u_i} u_{i,F}'}{a_{P,u_i}}} - \frac{1}{a_{P,u_i}} \left(\frac{\delta p'}{\delta x_i} \right)_P . \tag{13}$$

Since the global purpose of the presented method is to enforce continuity by implicitly calculating a pressure correction, the velocity correction has to be expressed in terms of the pressure correction and inserted into the discretized continuity equation (11) as follows

$$\left[\frac{\delta \left(\rho u_i^{(n*)} \right)}{\delta x_i} \right]_P - \frac{\delta}{\delta x_i} \left[\frac{\sum_{F \in NB(P)} a_{F,u_i} u_{i,F}'}{a_{P,u_i}} \right]_P - \frac{\delta}{\delta x_i} \left[\frac{1}{a_{P,u_i}} \left(\frac{\delta p'}{\delta x_i} \right)_P \right]_P = \frac{\delta \left(\rho u_i^{(n*)} \right)}{\delta x_i} + \frac{\delta \left(\rho u_i' \right)}{\delta x_i} = 0. \tag{14}$$

In order to separate the pressure correction p', the dominant variable from the velocities the semi-discrete form of the pressure correction equation can be stated as

$$\frac{\delta}{\delta x_i} \left[\frac{1}{a_{P,u_i}} \left(\frac{\delta p'}{\delta x_i} \right)_P \right]_P = \left[\frac{\delta \left(\rho u_i^{(n*)} \right)}{\delta x_i} \right]_P - \underline{\frac{\delta}{\delta x_i} \left[\frac{\sum_{F \in NB(P)} a_{F,u_i} u_{i,F}'}{a_{P,u_i}} \right]_P} . \tag{15}$$

The majority of the class of pressure correction algorithms has this equation as common basis. Each algorithm then introduces special distinguishable approximations of the velocity corrections that are, at the moment of solving the pressure equation, still unknown and will be calculated by means of equation (13), after a solution to (15) has been obtained. The method used in the present work is the SIMPLE Algorithm (Semi-Implicit Method for Pressure-Linked Equations REFERENCE). The approximation this algorithm performs is severe since the term containing the unknown velocity corrections is dropped entirely. The respective term has been underlined in (15) and accordingly in (13) for the velocity correction.

The approximation performed in the SIMPLE algorithm affects convergence in a way that the pressure correction has to be under-relaxed with a parameter $\alpha_p \in [0, 1]$

$$p_P^{(n)} = p_P^{(n-1)} + \alpha_{\mathbf{p}} p_P' \tag{16}$$

The velocities will be under-relaxed as well for stability reasons. This will change the discrete momentum balance to

$$\frac{a_{P,u_i}}{\alpha_{\mathbf{u}}} u_{i,P}^{(n)} + \sum_{F \in NB(P)} a_{F,u_i} u_{i,F}^{(n)} = b_{P,u_i}^{(n)} - \left(\frac{\delta p^{(n)}}{\delta x_i} \right)_P + \frac{(1 - \alpha_{\mathbf{u}}}{\alpha_{\mathbf{u}}} u_{i,P}^{(n-1)},$$

A more detailed integration of the under-relaxation into the momentum balance will be found at the end of subsection (REFERENCE). The under-relaxation leads to an undesirable property of the SIMPLE algorithm, if the mass fluxes, to be introduced in subsection (REFERENCE), are calculated by the standard Rhie-Chow momentum interpolation scheme, which is the common practice. In this case the interpolation method will generate results that depend on the under-relaxation factor of the velocity $alpha_{\mathbf{u}}$. Since the present work aims at a detailed comparison of the results of different solver algorithms an superior interpolation technique will be presented, namely the pressure weighted interpolation method.

Generally the SIMPLE algorithm can be represented by the following iterative procedure as in Algorithm 3.2.

Algorithm 1 SIMPLE Algorithm

```
Solve linearized momentum balances
i = 1                                                                    ▷ *
while i ≤ x.n and k > x.key[i] do          ▷ localize key                ▷ *
    i = i + 1
end while
if i ≤ x.n then                            ▷ avoid array out of bounds   ▷ *
    if k == x.key[i] then                                                ▷ check if key is in node
        inNode = true
    end if
end if
```

3.3 Discretization of the Momentum Balance

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

$$\underbrace{\iint_S (\rho u_i u_j) n_j \mathrm{d}S}_{\text{convective term}} - \underbrace{\iint_S \left(\mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right) n_j \mathrm{d}S}_{\text{diffusive term}} = \underbrace{-\iiint_V \frac{\partial p}{\partial x_i}\mathrm{d}V}_{\text{sourceterm pressure}} \underbrace{- \iiint_V \rho\beta\,(T - T_0)\,\mathrm{d}V}_{\text{sourceterm temperature}} \tag{17}$$

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 and due to deferred corrections in an explicit manner whereas the terms on the right will be treated exclusively in an explicit manner.

3.3.1 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 (2.4), the non linearity will be dealt with by means of an iterative process, the Picard iteration. The part dependent on 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

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

where $F_{i,f}^c := \iint_{S_f} \rho u_i^{(n)} u_j^{(n-1)} n_j \mathrm{d}S$ 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

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

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

$$F_{i,f}^{c,l} = u_{i,F}\min(\dot{m}_f, 0) + u_{i,P}\max(0, \dot{m}_f)$$
$$F_{i,f}^{c,h} = u_{i,F}\,\gamma_f + u_{i,P}\,(1-\gamma_f),$$

where the variable values have to be taken from the previous iteration step $(n-1)$ as necessary and the mass flux $\dot{m}_f$ 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

$$a_{F,u_i}^c = \min(\dot{m}_f, 0), \qquad a_{P,u_i}^c = \sum_{F\in NB(P)} \max(0, \dot{m}_f) \tag{18a}$$

$$b_{P,u_i}^c = \sum_{F\in NB(P)} \eta\left(u_{i,F}^{(n-1)}\left(\min(\dot{m}_f, 0) - \gamma_f\right)\right) + \eta\left(u_{i,P}^{(n-1)}\left(\max(0, \dot{m}_f) - \left(1-\gamma_f\right)\right)\right) \tag{18b}$$

3.3.2 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 2.3 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 equation (**??**) 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

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

where $F_{i,f}^d$ denotes the diffusive flux through an individual boundary face. Section 2.3 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 2.3 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. To begin with all present integrals will be approximated using the midpoint rule of integration. The diffusive flux $F_{i,f}^d$ for a generic face S_f then reads

$$F_{i,f}^d \approx \underline{\mu\left(\frac{\partial u_i}{\partial x_j}\right)_f n_j S_f} + \mu\left(\frac{\partial u_j}{\partial x_i}\right)_f n_j S_f.$$

Using central differences for the implicit discretization of the directional derivative and furthermore using the *orthogonal correction* approach from 2.3.2 the approximation can be derived as

$$F_{i,f}^d \approx \mu\left(\underline{||\boldsymbol{\Delta}_f||_2 \frac{u_{P_i} - u_{F_i}}{||\mathbf{x}_P - \mathbf{x}_F||_2} - (\nabla u_i)_f^{(n-1)} \cdot \left(\boldsymbol{\Delta}_f - \mathbf{S}_f\right)}\right) + \mu\left(\frac{\partial u_j}{\partial x_i}\right)_f^{(n-1)} n_{f_i}$$

$$= \mu\left(\underline{S_f \frac{u_{P_i} - u_{F_i}}{||\mathbf{x}_P - \mathbf{x}_F||_2} - \left(\frac{\partial u_i}{\partial x_j}\right)_f^{(n-1)} \left(\xi_{f_i} - n_{f_i}\right) S_f}\right) + \mu\left(\frac{\partial u_j}{\partial x_i}\right)_f^{(n-1)} n_{f_i},$$

where the unit vector pointing in direction of the straight line connecting control volume P and control volume F is denoted as

$$\xi_f = \frac{\mathbf{x}_P - \mathbf{x}_F}{||\mathbf{x}_p - \mathbf{x}_F||_2}.$$

The interpolation of the cell center gradients to the boundary faces is performed as in (2). Now the contribution of the diffusive part to the matrix coefficients and the right hand side can be calculated as

$$a^d_{F,u_i} = -\frac{\mu S_f}{||\mathbf{x}_P - \mathbf{x}_F||_2}, \qquad a^d_{P,u_i} = \sum_{F \in NB(P)} \frac{\mu S_f}{||\mathbf{x}_P - \mathbf{x}_F||} \tag{19a}$$

$$\begin{aligned} b^d_{F,u_i} &= \sum_{F \in NB(P)} \left(\frac{\partial u_i}{\partial x_j}\right)^{(n-1)}_f \left(\xi_{f_i} - n_{f_i}\right) S_f - \mu \left(\frac{\partial u_j}{\partial x_i}\right)^{(n-1)}_f n_{f_i} S_f \\ &= \left(\frac{\partial u_i}{\partial x_j}\right)^{(n-1)}_f \xi_{f_i} S_f - \mu \left(\left(\frac{\partial u_i}{\partial x_j}\right)^{(n-1)}_f - \left(\frac{\partial u_j}{\partial x_i}\right)^{(n-1)}_f \right) n_{f_i} S_f . \end{aligned} \tag{19b}$$

3.3.3 Discretization of the Source Terms

Since in the segregated solution approach in every equation all other variables but the dominant one are treated as constants and furthermore the source terms in equation (17) do not depend on the dominant variable the discretization is straightforward. The source terms of the momentum balance are discretized using the midpoint rule of integration, which leads to the source term

$$-\iiint_V \frac{\partial p}{\partial x_i} \mathrm{d}V - \iiint_V \rho \beta \left(T - T_0\right) \mathrm{d}V \approx -\left(\frac{\partial p}{\partial x_i}\right)^{(n-1)}_P V_P - \rho \beta \left(T_P^{(n-1)} - T_0\right) V_P = b^{sc}_{P,u_i} \tag{20}$$

3.4 Discretization of the Pressure Correction Equation

3.5 Discretization of the Temperature Equation

The discretization of the temperature equation is performed by the same means as for the momentum balance. The only difference is a simpler diffusion term. The integral form of the temperature equation after applying the Gauss' theorem of integration is

$$\underbrace{\iint_S \rho u_j T n_j \mathrm{d}S}_{\text{advective term}} - \underbrace{\iint_S \kappa \frac{\partial T}{\partial x_j} n_j \mathrm{d}V}_{\text{diffusive term}} = \underbrace{\iiint_V q_T \mathrm{d}V}_{\text{source term}} .$$

Proceeding as in the previous subsections one can now discretize the advective, the diffusive term and the source term. Since this process does not provide further insight, just the final results will be presented. The discretization yields the matrix coefficients as

$$a_{F,T} = \min(\dot{m}_f, 0) + \frac{\kappa S_f}{||\mathbf{x}_P - \mathbf{x}_F||_2} \tag{21a}$$

$$a_{P,T} = \sum_{F \in NB(P)} \max(0, \dot{m}_f) - \frac{\kappa S_f}{||\mathbf{x}_P - \mathbf{x}_F||_2} \tag{21b}$$

$$\begin{aligned} b_{P,T} = &\sum_{F \in NB(P)} \eta \left(T_F^{(n-1)} \left(\min(\dot{m}_f, 0) - \gamma_f \right) \right) \\ &\quad + \eta \left(T_P^{(n-1)} \left(\max(0, \dot{m}_f) - \left(1 - \gamma_f\right) \right) \right) \\ &+ \sum_{F \in NB(P)} \left(\frac{\partial T}{\partial x_j}\right)^{(n-1)}_f \left(\xi_{f_j} - n_{f_j}\right) S_f \\ &+ q_{T_P} V_P . \end{aligned} \tag{21c}$$

Again it is possible though not always necessary, as in the case of the velocities, to under-relax the solution of the resulting linear system with a factor α_T. This can be accomplished as shown in the previous sections.

3.6 Boundary Conditions

3.7 Structure of the Assembled Linear Systems

The objective of a finite volume method is to create a set of linear algebraic equations by discretizing partial differential equations. For the momentum balance all necessary components have been calculated. Taking all contributions together leads to the following linear algebraic equation for each control volume P

$$a_{P,u_i} u_{P_i} + \sum_{F \in NB(P)} a_F u_{F_i} = b_{P,u_i},$$

where the coefficients are composed as

$$a_{P,u_i} = a^c_{P,u_i} - a^d_{P,u_i} \tag{22}$$

$$a_{F,u_i} = a^c_{F,u_i} - a^d_{F,u_i} \tag{23}$$

$$b_{P,u_i} = b^c_{P,u_i} - b^d_{P,u_i} + b^{sc}_{P,u_i}. \tag{24}$$

In the case of control volumes located at boundaries some of the coefficients will be calculated in a different manner. This aspect will be addressed in a later section. For the decoupled iterative solution process of the Navier-Stokes equations it is necessary to reduce the change of each dependent variable in each iteration. Normally this is done by a *under-relaxation* technique, a convex combination of the solution of the linear system present iteration (n) and from the previous iteration $(n-1)$ with the under-relaxation parameter α_{u_i}. Generally speaking this parameter can be chosen individually for each equation. Since there are no rules for choosing this parameters in a general setting the under-relaxation parameter for the velocities is chosen to be equal for all three velocities, $\alpha_{u_i} = \alpha_{\mathbf{u}}$. This has the further advantage that, in case the boundary conditions are implemented with the same intention, the linear system for each of the velocities remains unchanged except for the right hand side. This helps to increase memory efficiency.

Let the solution for the linear system without under-relaxation be denoted as

$$\tilde{u}^{(n)}_{P_i} := \frac{b_{P,u_i} - \sum_{F \in NB(P)} a_F u_{F_i}}{a_{P,u_i}},$$

Which is only a formal expression. A convex combination as described yields

$$\begin{aligned} u^{(n)}_{P_i} &:= \alpha_{\mathbf{u}} \tilde{u}^{(n)}_{P_i} + (1-\alpha_{\mathbf{u}}) u^{(n-1)}_{P_i} \\ &= \alpha_{\mathbf{u}} \frac{b_{P,u_i} - \sum_{F \in NB(P)} a_F u_{F_i}}{a_{P,u_i}} + (1-\alpha_{\mathbf{u}}) u^{(n-1)}_{P_i}, \end{aligned}$$

an expression that can be modified to derive a linear system whose solution is the under-relaxed velocity

$$\frac{a_{P,u_i}}{\alpha_{\mathbf{u}}} u_{i,P} + \sum_{F \in NB(P)} a_{F,u_i} u_{i,F} = b_{P,u_i} + \frac{(1-\alpha_{\mathbf{u}}) a_{P,u_i}}{\alpha_{\mathbf{u}}} u^{(n-1)}_{i,P}.$$

It must be noted that under-relaxation has not been accounted for in the derivation of the Rhie-Chow interpolation method (REFERENCE). As it has been shown the results depend on the choice of the under-relaxation factor. However under-relaxation is necessary for overall convergence, so in order to be able to compare the results of the different solver algorithms this dependency has to be eliminated. Subsection (REFERENCE) will present a common approach to resolve this dependency.

4 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] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

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

[14] 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.

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

[16] 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.

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

[18] 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.

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

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

[21] 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.

[22] 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.

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

[24] 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.

[25] 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.

[26] 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.

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