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

List of Figures

List of Tables

List of Algorithms

1 Introduction

This thesis is about.

4 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 (10). 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.

4.1 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 [5,20,22,33]. A major drawback if SIMPLE-Type algorithms, to be introduced in section 4.2, 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* [25]. 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 [25] 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* [22].

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 4.3. The semi-discrete implicit momentum balances read if one solves for the velocity at node P or Q

$$u_{i,P}^{(n)} = -\frac{\alpha_{\mathbf{u}_P}}{a_{P,u_i}} \left(\sum_{F \in NB(P)} a_{F,u_i} u_{i,F}^{(n)} + b_{P,u_i}^{(n-1)} - V_P \left(\frac{\partial p}{\partial x_i} \right)_P^{(n-1)} \right) + (1-\alpha_{\mathbf{u}}) u_{i,P}^{(n-1)} \tag{15a}$$

$$\text{and} \quad u_{i,Q}^{(n)} = -\frac{\alpha_{\mathbf{u}_Q}}{a_{Q,u_i}} \left(\sum_{F \in NB(Q)} a_{F,u_i} u_{i,F}^{(n)} + b_{Q,u_i}^{(n-1)} - V_Q \left(\frac{\partial p}{\partial x_i} \right)_Q^{(n-1)} \right) + (1-\alpha_{\mathbf{u}}) u_{i,Q}^{(n-1)} \tag{15b}$$

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 [27] 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)

$$u_{i,f}^{(n)} = -\frac{\alpha_{\mathbf{u}_f}}{a_{f,u_i}} \left(\sum_{F \in NB(f)} a_{F,u_i} u_{i,F}^{(n)} + b_{f,u_i}^{(n-1)} - V_f \left(\frac{\partial p}{\partial x_i} \right)_f^{(n-1)} \right) + (1-\alpha_{\mathbf{u}}) u_{i,f}^{(n-1)}. \tag{16}$$

To guarantee convergence of this expression of $u_{i,f}$ under-relaxation is necessary [20]. To eliminate the undefined 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

$$\frac{\alpha_{\mathbf{u}_f}}{a_{f,u_i}} \left(\sum_{F \in NB(f)} a_{F,u_i} u_{i,F}^{(n)} \right) \approx (1-\gamma_f) \frac{\alpha_{\mathbf{u}_P}}{a_{P,u_i}} \left(\sum_{F \in NB(P)} a_{F,u_i} u_{i,F}^{(n)} \right) + \gamma_f \frac{\alpha_{\mathbf{u}_Q}}{a_{Q,u_i}} \left(\sum_{F \in NB(Q)} a_{F,u_i} u_{i,F}^{(n)} \right) \tag{17a}$$

$$\text{and} \quad \frac{\alpha_{\mathbf{u}_f}}{a_{f,u_i}} \approx (1-\gamma_f) \frac{\alpha_{\mathbf{u}_P}}{a_{P,u_i}} + \gamma_f \frac{\alpha_{\mathbf{u}_Q}}{a_{Q,u_i}}, \tag{17b}$$

where γ_f is a geometric interpolation factor.

Using the assumptions made in equation (17) the expression in equation (16) 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{}& (1-\gamma_f)\left(-\frac{\alpha_{\mathbf{u}}}{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}}}{a_{Q,u_i}} \sum_{F\in NB(Q)} a_{F,u_i} u_{i,F}^{(n)}\right) \\
&+ \frac{\alpha_{\mathbf{u}}}{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}{\partial x_i}\right)_f^{(n-1)} + (1-\alpha_{\mathbf{u}}) u_{i,f}^{(n-1)} \\
={}& (1-\gamma_f) u_{i,P}^{(n)} - (1-\gamma_f)\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 + (1-\alpha_{\mathbf{u}}) u_{i,f}^{(n-1)} \\
={}& \left[(1-\gamma_f) u_{i,P}^{(n-1)} + \gamma_f u_{i,Q}^{(n-1)}\right] \\
&- \left[\left((1-\gamma_f)\frac{\alpha_{\mathbf{u}} V_P}{a_{P,u_i}} + \gamma_f \frac{\alpha_{\mathbf{u}} V_Q}{a_{Q,u_i}}\right)\left(\frac{\partial p}{\partial x_i}\right)_f^{(n-1)} - (1-\gamma_f)\frac{\alpha_{\mathbf{u}} V_P}{a_{P,u_i}}\left(\frac{\partial p}{\partial x_i}\right)_P^{(n-1)} - \gamma_f \frac{\alpha_{\mathbf{u}} V_Q}{a_{Q,u_i}}\left(\frac{\partial p}{\partial x_i}\right)_Q^{(n-1)}\right] \\
&+ (1-\alpha)\left[u_{i,f}^{(n-1)} - (1-\gamma_f) u_{i,P}^{(n-1)} - \gamma_f u_{i,Q}^{(n-1)}\right] \\
\approx{}& \left[(1-\gamma_f) u_{i,P}^{(n-1)} + \gamma_f u_{i,Q}^{(n-1)}\right] \\
&- \left((1-\gamma_f)\frac{\alpha_{\mathbf{u}} V_P}{a_{P,u_i}} + \gamma_f \frac{\alpha_{\mathbf{u}} V_Q}{a_{Q,u_i}}\right)\left[\left(\frac{\partial p}{\partial x_i}\right)_f^{(n-1)} - (1-\gamma_f)\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_{\mathbf{u}})\left[u_{i,f}^{(n-1)} - (1-\gamma_f) u_{i,P}^{(n-1)} - \gamma_f u_{i,Q}^{(n-1)}\right].
\end{aligned}
\tag{18}
$$

It should be noted that the argumentation that led to the last expression is that the task of the pressure gradient corrector is to suppress oscillations.

If there are no oscillations this part does not come into consideration. This is however true on equidistant grids, where $\gamma_f = 1/2$ and central differences are used to calculate the gradients. On arbitrary orthogonal grids another modification has to be performed which is based on the special case of the mean value theorem of differential calculus and the following

Proposition. *Let $x_1, x_2 \in \mathbb{R}$ with $x_1 \neq x_2$ and $p(x) = a_0 + a_1 x + a_2 x^2$ a real polynomial function. Then*

$$\frac{dp}{dx}\left(\frac{x_1+x_2}{2}\right) = \frac{p(x_2)-p(x_1)}{x_2-x_1},$$

i.e. the slope of the secant equals the value of the first derivative of p exactly half the way between x_1 and x_2.

Proof. Evaluation of the derivative yields

$$\frac{dp}{dx}\left(\frac{x_1+x_2}{2}\right) = a_1 + 2a_2\frac{x_1+x_2}{2} = a_1 + a_2(x_1+x_2).$$

On the other hand the slope of the secant, using the third binomial rule can be expressed as

$$
\begin{aligned}
\frac{p(x_2)-p(x_1)}{x_2-x_1} &= \frac{a_0 + a_1 x_2 + a_2 x_2^2 - \left(a_0 + a_1 x_1 + a_2 x_1^2\right)}{x_2-x_1} \\
&= \frac{a_1(x_2-x_1) + a_2\left(x_2^2 - x_1^2\right)}{x_2-x_1} \\
&= a_1 + a_2(x_2+x_1).
\end{aligned}
$$

The comparison of both expressions completes the proof. □

It is desirable for the pressure corrector to vanish independent of the grid spacing if the profile of the pressure is quadratic and hence does not exhibit oscillations. According to the preceding proposition this can be accomplished by modifying equation (18) to average the pressure gradients from node P and Q instead of interpolating linearly

$$\begin{aligned} u_{i,f}^{(n)} &= \left[\left(1-\gamma_f\right)u_{i,P}^{(n-1)}+\gamma_f u_{i,Q}^{(n-1)}\right] \\ &\quad -\left(\left(1-\gamma_f\right)\frac{\alpha_{\mathbf{u}}V_P}{a_{P,u_i}}+\gamma_f\frac{\alpha_{\mathbf{u}}V_Q}{a_{Q,u_i}}\right)\left[\left(\frac{\partial p}{\partial x_i}\right)_f^{(n-1)}-\frac{1}{2}\left(\left(\frac{\partial p}{\partial x_i}\right)_P^{(n-1)}+\left(\frac{\partial p}{\partial x_i}\right)_Q^{(n-1)}\right)\right] \\ &\quad \underline{+\left(1-\alpha_{\mathbf{u}}\right)\left[u_{i,f}^{(n-1)}-\left(1-\gamma_f\right)u_{i,P}^{(n-1)}-\gamma_f\, u_{i,Q}^{(n-1)}\right]}. \end{aligned} \tag{19}$$

Comparing this final expression with the standard interpolation scheme it is evident that the underlined term is not taken into consideration [11] normally. However section **??** shows that neglecting this term creates under-relaxation factor dependent results.

PROOF THAT THIS GUARANTEES INDEPENDENT RESULTS MATHEMATICALLY

4.2 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

$$u_{i,P}^{(n)} = -\frac{\alpha_{\mathbf{u}P}}{a_{P,u_i}}\left(\sum_{F\in NB(P)} a_{F,u_i}u_{i,F}^{(n)}+b_{P,u_i}^{(n-1)}-V_P\left(\frac{\partial p}{\partial x_i}\right)_P^{(n-1)}\right)+\left(1-\alpha_{\mathbf{u}}\right)u_{i,P}^{(n-1)} \tag{20}$$

where the pressure gradient has been discretized only symbolically and b_{P,u_i} denotes the source term. At this stage the equations are still coupled and non-linear. As described in section 3.4 the Picard iteration process has been used to linearize the equations. Every momentum balance equation only depends on the one dominant variable u_i. Furthermore the coupling of the momentum balances through the convective term (u_iu_j) has been resolved in the process of linearization. The decoupled momentum balances can then be solved sequentially for the dominant variable u_i. 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 (6.4.2). This leads to the following linear equation

$$u_{i,P}^{(n*)} = -\frac{\alpha_{\mathbf{u}P}}{a_{P,u_i}}\left(\sum_{F\in NB(P)} a_{F,u_i}u_{i,F}^{(n*)}+b_{P,u_i}^{(n-1)}-V_P\left(\frac{\partial p}{\partial x_i}\right)_P^{(n-1)}\right)+\left(1-\alpha_{\mathbf{u}}\right)u_{i,P}^{(n-1)} \tag{21}$$

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

$$\frac{\delta\left(\rho u_i^{(n)}\right)}{\delta x_i}=0. \quad \text{REVISION NECESSARY} \tag{22}$$

Applying the same procedure as in section 4.1 to equation (**??**) results in the following expression for the face velocities after solving the discretized momentum balances

$$\begin{aligned} u_{i,f}^{(n)} &= \left[\left(1-\gamma_f\right)u_{i,P}^{(n*)}+\gamma_f u_{i,Q}^{(n*)}\right] \\ &\quad -\left(\left(1-\gamma_f\right)\frac{\alpha_{\mathbf{u}}V_P}{a_{P,u_i}}+\gamma_f\frac{\alpha_{\mathbf{u}}V_Q}{a_{Q,u_i}}\right)\left[\left(\frac{\partial p}{\partial x_i}\right)_f^{(n-1)}-\frac{1}{2}\left(\left(\frac{\partial p}{\partial x_i}\right)_P^{(n-1)}+\left(\frac{\partial p}{\partial x_i}\right)_Q^{(n-1)}\right)\right] \\ &\quad +\left(1-\alpha_{\mathbf{u}}\right)\left[u_{i,f}^{(n-1)}-\left(1-\gamma_f\right)u_{i,P}^{(n-1)}-\gamma_f\, u_{i,Q}^{(n-1)}\right]. \end{aligned} \tag{23}$$

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' as in

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

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

$$u_{i,P}^{(n)} = -\frac{\alpha_{\mathbf{u}_P}}{a_{P,u_i}} \left(\sum_{F\in NB(P)} a_{F,u_i} u_{i,F}^{(n)} + b_{P,u_i}^{(n-1)} - V_P \left(\frac{\partial p}{\partial x_i} \right)_P^{(n)} \right) + (1-\alpha_{\mathbf{u}}) u_{i,P}^{(n-1)}. \tag{24}$$

It should be noted that the only difference to the equation that will be solved in the next outer iteration is that the source term b_{P,u_i} has not been updated yet. The same applies to the equation for the face velocity $u_{i,f}$

$$\begin{aligned} u_{i,f}^{(n)} = & \left[\left(1-\gamma_f\right) u_{i,P}^{(n)} + \gamma_f u_{i,Q}^{(n)} \right] \\ & - \left(\left(1-\gamma_f\right) \frac{\alpha_{\mathbf{u}} V_P}{a_{P,u_i}} + \gamma_f \frac{\alpha_{\mathbf{u}} V_Q}{a_{Q,u_i}} \right) \left[\left(\frac{\partial p}{\partial x_i} \right)_f^{(n)} - \frac{1}{2} \left(\left(\frac{\partial p}{\partial x_i} \right)_P^{(n)} + \left(\frac{\partial p}{\partial x_i} \right)_Q^{(n)} \right) \right] \\ & + (1-\alpha_{\mathbf{u}}) \left[u_{i,f}^{(n-1)} - \left(1-\gamma_f\right) u_{i,P}^{(n-1)} - \gamma_f \, u_{i,Q}^{(n-1)} \right]. \end{aligned} \tag{25}$$

To couple velocity and pressure correctors one can subtract equations (21) from (24), (23) from (25) to get

$$u'_{i,P} = -\frac{\alpha_{\mathbf{u}_P}}{a_{P,u_i}} \left(\underline{\sum_{F\in NB(P)} a_{F,u_i} u'_{i,F}} - V_P \left(\frac{\partial p'}{\partial x_i} \right)_P^{(n)} \right) \tag{26}$$

$$u'_{i,f} = \left[\left(1-\gamma_f\right) u'_{i,P} + \gamma_f u'_{i,Q} \right] - \left(\left(1-\gamma_f\right) \frac{\alpha_{\mathbf{u}} V_P}{a_{P,u_i}} + \gamma_f \frac{\alpha_{\mathbf{u}} V_Q}{a_{Q,u_i}} \right) \left[\left(\frac{\partial p}{\partial x_i} \right)_f' - \frac{1}{2} \left(\left(\frac{\partial p}{\partial x_i} \right)_P' + \left(\frac{\partial p}{\partial x_i} \right)_Q' \right) \right], \tag{27}$$

The majority of the class of pressure correction algorithms has this equations as a 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. The method used in the present work is the SIMPLE Algorithm (Semi-Implicit Method for Pressure-Linked Equations [**?**]). 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 equation 26 accordingly. 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 solely in terms of the pressure correction. This can be accomplished by substituting equation (26) in to equation (27). This gives

$$u'_{i,f} = -\left(\left(1-\gamma_f\right) \frac{\alpha_{\mathbf{u}} V_P}{a_{P,u_i}} + \gamma_f \frac{\alpha_{\mathbf{u}} V_Q}{a_{Q,u_i}} \right) \left[\left(\frac{\partial p}{\partial x_i} \right)_f' \right] \tag{28}$$

which is then inserted into the discretized continuity equation (22) to obtain

$$\left[\frac{\delta \left(\rho u_i^{(n*)} \right)}{\delta x_i} \right]_P - \frac{\delta}{\delta x_i} \left[\frac{\sum_{F\in NB(P)} a_{F,u_i} u'_{i,F}}{a_{P,u_i}} \right]_P - \frac{\delta}{\delta x_i} \left[\frac{1}{a_{P,u_i}} \left(\frac{\delta p'}{\delta x_i} \right)_P \right]_P = \frac{\delta \left(\rho u_i^{(n*)} \right)}{\delta x_i} + \frac{\delta \left(\rho u'_i \right)}{\delta x_i} = 0. \tag{29}$$

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

$$\frac{\delta}{\delta x_i} \left[\frac{1}{a_{P,u_i}} \left(\frac{\delta p'}{\delta x_i} \right)_P \right]_P = \left[\frac{\delta \left(\rho u_i^{(n*)} \right)}{\delta x_i} \right]_P \underline{- \frac{\delta}{\delta x_i} \left[\frac{\sum_{F\in NB(P)} a_{F,u_i} u'_{i,F}}{a_{P,u_i}} \right]_P}. \tag{30}$$

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 (**??**), after a solution to (30) 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 (30) and accordingly in (**??**) 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 $\alpha_p \in [0,1]$

$$p_P^{(n)} = p_P^{(n-1)} + \alpha_{\mathbf{p}} p'_P \tag{31}$$

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 4.2.

Algorithm 1 SIMPLE Algorithm

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

4.3 Discretization of the Momentum Balance

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

$$\underbrace{\iint\limits_S (\rho u_i u_j) n_j \mathrm{d}S}_{\text{convective term}} - \underbrace{\iint\limits_S \left(\mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right) n_j \mathrm{d}S}_{\text{diffusive term}} = - \underbrace{\iiint\limits_V \frac{\partial p}{\partial x_i} \mathrm{d}V}_{\text{sourceterm pressure}} - \underbrace{\iiint\limits_V \rho \beta \left(T - T_0 \right) \mathrm{d}V}_{\text{sourceterm temperature}} \tag{32}$$

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.

4.3.1 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 (3.4), 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 $\dot{m}_f = \iint\limits_{S_f} \rho u_j^{(n-1)} n_j \mathrm{d}S$. 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

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

where $F^c_{i,f} := \iint\limits_{S_f} \rho u_i^{(n)} u_j^{(n-1)} n_j \mathrm{d}S$ 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

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

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

$$a^c_{F,u_i} = \min(\dot{m}_f, 0), \qquad a^c_{P,u_i} = \sum_{F \in NB(P)} \max(0, \dot{m}_f) \tag{33a}$$

$$\begin{aligned} b^c_{P,u_i} = \sum_{F \in NB(P)} & \eta \left(u_{i,F}^{(n-1)} \left(\min(\dot{m}_f, 0) - \gamma_f \right) \right) \\ & + \eta \left(u_{i,P}^{(n-1)} \left(\max(0, \dot{m}_f) - \left(1 - \gamma_f \right) \right) \right) \end{aligned} \tag{33b}$$

4.3.2 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 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 equation (7) 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

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

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. To begin with all present integrals will be approximated using the midpoint rule of integration. The diffusive flux $F^d_{i,f}$ for a generic face S_f then reads

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

Using central differences for the implicit discretization of the directional derivative and furthermore using the *orthogonal correction* approach from 3.3.2 the approximation can be derived as

$$\begin{aligned} F^d_{i,f} &\approx \mu \left(\underline{||\boldsymbol{\Delta}_f||_2 \frac{u_{P_i} - u_{F_i}}{||\mathbf{x}_P - \mathbf{x}_F||_2} - (\nabla u_i)_f^{(n-1)} \cdot \left(\boldsymbol{\Delta}_f - \mathbf{S}_f \right)} \right) + \mu \left(\frac{\partial u_j}{\partial x_i} \right)_f^{(n-1)} n_{f_i} \\ &= \mu \left(\underline{S_f \frac{u_{P_i} - u_{F_i}}{||\mathbf{x}_P - \mathbf{x}_F||_2} - \left(\frac{\partial u_i}{\partial x_j} \right)_f^{(n-1)} \left(\xi_{f_i} - n_{f_i} \right) S_f} \right) + \mu \left(\frac{\partial u_j}{\partial x_i} \right)_f^{(n-1)} n_{f_i} , \end{aligned}$$

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 (12). Now the contribution of the diffusive part to the matrix coefficients and the right hand side can be calculated as

$$a^d_{F,u_i} = -\frac{\mu S_f}{||\mathbf{x}_P - \mathbf{x}_F||_2}, \qquad a^d_{P,u_i} = \sum_{F \in NB(P)} \frac{\mu S_f}{||\mathbf{x}_P - \mathbf{x}_F||} \tag{34a}$$

$$\begin{aligned} b^d_{F,u_i} &= \sum_{F \in NB(P)} \left(\frac{\partial u_i}{\partial x_j} \right)_f^{(n-1)} \left(\xi_{f_i} - n_{f_i} \right) S_f - \mu \left(\frac{\partial u_j}{\partial x_i} \right)_f^{(n-1)} n_{f_i} S_f \\ &= \left(\frac{\partial u_i}{\partial x_j} \right)_f^{(n-1)} \xi_{f_i} S_f - \mu \left(\left(\frac{\partial u_i}{\partial x_j} \right)_f^{(n-1)} - \left(\frac{\partial u_j}{\partial x_i} \right)_f^{(n-1)} \right) n_{f_i} S_f . \end{aligned} \tag{34b}$$

4.3.3 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 (32) 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

$$-\iiint_V \frac{\partial p}{\partial x_i} \mathrm{d}V - \iiint_V \rho \beta \left(T - T_0 \right) \mathrm{d}V \approx -\left(\frac{\partial p}{\partial x_i} \right)_P^{(n-1)} V_P - \rho \beta \left(T_P^{(n-1)} - T_0 \right) V_P = b^{sc}_{P,u_i} \tag{35}$$

4.4 Discretization of the Pressure Correction Equation

4.5 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

$$a_{F,T} = \min(\dot{m}_f, 0) + \frac{\kappa S_f}{||\mathbf{x}_P - \mathbf{x}_F||_2} \tag{36a}$$

$$a_{P,T} = \sum_{F\in NB(P)} \max(0, \dot{m}_f) - \frac{\kappa S_f}{||\mathbf{x}_P - \mathbf{x}_F||_2} \tag{36b}$$

$$\begin{aligned} b_{P,T} = \sum_{F\in NB(P)} &\eta\left(T_F^{(n-1)}\left(\min(\dot{m}_f, 0) - \gamma_f\right)\right) \\ &+ \eta\left(T_P^{(n-1)}\left(\max(0, \dot{m}_f) - \left(1 - \gamma_f\right)\right)\right) \\ + &\sum_{F\in NB(P)} \left(\frac{\partial T}{\partial x_j}\right)_f^{(n-1)} \left(\xi_{f_j} - n_{f_j}\right) S_f \\ + &q_{T_P} V_P. \end{aligned} \tag{36c}$$

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.

4.6 Boundary Conditions

4.7 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

$$a_{P,u_i} = a^c_{P,u_i} - a^d_{P,u_i} \tag{37}$$

$$a_{F,u_i} = a^c_{F,u_i} - a^d_{F,u_i} \tag{38}$$

$$b_{P,u_i} = b^c_{P,u_i} - b^d_{P,u_i} + b^{sc}_{P,u_i}. \tag{39}$$

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

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

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

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

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

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

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.

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

[9] 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.

[10] 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.

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

[12] 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.

[13] 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.

[14] 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.

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

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

[17] 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.

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

[19] 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.

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

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

[22] 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.

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

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

[25] 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.

[26] 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.

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

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

[29] 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.

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

[31] 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.

[32] 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.

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