Immersed boundary methods

Henry Bandringa

Master Thesis in Applied Mathematics
August 2010
**Immersed boundary methods**

**Summary**

Usually, a body conforming grid is being used for computing the flow around an arbitrary body. This approach requires coordinate transformations and/or complex grid generation. Especially for moving bodies, in every time step a new mesh has to be generated, which requires a lot of computing time. In contrast, an immersed boundary method uses a non-body conforming Cartesian grid. In general a body does not align with the grid, so some of the computational cells will be cut. The cut cells can be treated in several ways. These treatments can be globally categorized into a continuous forcing approach, a discrete forcing approach and cut-cell methods. Many different immersed boundary methods have been developed over the years and one of my tasks was to recommend a few of these methods, which could be implemented in the computational fluid dynamics code ComFLOW. Before computing flows in complex geometries, some simple geometries will be used for testing the methods. One of the testing geometries is a channel placed oblique to the grid.
## Contents

1 Overview of immersed boundary methods ........................................ 1

1 Introduction ...................................................................................... 3
   1.1 General considerations .............................................................. 3

2 Categories of IB methods ................................................................. 5

3 Continuous forcing approach ............................................................ 7
   3.1 Elastic boundaries .................................................................... 7
   3.2 Rigid boundaries ...................................................................... 8
   3.3 Distributed Lagrange multiplier method .................................... 11
   3.4 Immersed interface method ....................................................... 12
   3.5 Solving linear system .................................................................. 12
   3.6 General considerations .............................................................. 12

4 Direct forcing approach (discrete approach) ........................................ 15
   4.1 General idea ............................................................................. 15
   4.2 Direct forcing ........................................................................... 16
      4.2.1 Improvements .................................................................... 17
      4.2.2 Fulfilling conservation laws ............................................... 20
      4.2.3 Conservation of mass ......................................................... 20
      4.2.4 Treatment of the interior of the body ................................... 24
      4.2.5 Mass source/sink approach ............................................... 25
      4.2.6 Hybrid Cartesian/immersed boundary method ..................... 25
      4.2.7 Physical Virtual Model ..................................................... 26
   4.3 Ghost-cell finite-difference approach ......................................... 26
      4.3.1 Basic formulation ............................................................... 27
      4.3.2 Improvements .................................................................... 29
      4.3.3 Difference between ghost cell and ghost fluid method .......... 32
   4.4 General considerations .............................................................. 32
5 Cut-cell finite-volume approach
5.1 Basic formulation 33
5.2 Improvements 35
5.3 LS-STAG 37
5.4 Staggered grid compared to collocated grid 38
5.5 General considerations 38

6 Evaluation of immersed boundary methods 39
6.1 Mass conservation 39
6.2 High Reynolds numbers/Turbulent flows 42
6.3 Large eddy simulations 42
6.4 Compressible flows 43
6.5 Staggered grid 43
6.6 General considerations 43
6.7 Recommended immersed boundary methods 44

II Discretization of LS-STAG 45

7 Basic discretization 47
7.1 LS-STAG mesh 47
7.2 Navier-Stokes equations 48
7.3 Pressure gradient 49
7.4 Navier-Stokes discretized 50
7.4.1 Normal stress discretization 51
7.4.2 Poisson equation 52

8 Discretization of the convective fluxes 55
8.1 Normal fluid cell 56
8.1.1 Control volume for u 56
8.1.2 Control volume for v 57
8.2 South trapezoidal cut-cell 59
8.2.1 Control volume for u 59
8.2.2 Control volume for v 61
8.3 South-east triangular cut-cell 62
8.3.1 Control volume for u 62
8.3.2 Control volume for v 64
8.4 South-east pentagonal cut-cell 65
8.4.1 Control volume for u 65
8.4.2 Control volume for v 68
CONTENTS

9 Discretization of the viscous fluxes 71
  9.1 General considerations .................................. 71
     9.1.1 Discretization of the normal stress fluxes .......... 71
     9.1.2 Discretization of the shear stress fluxes .......... 72
     9.1.3 Five-point structure for viscous term .......... 74
  9.2 Normal fluid cell ........................................ 74
     9.2.1 u-momentum .................................... 74
     9.2.2 v-momentum .................................... 76
  9.3 South trapezoidal cut-cell ................................ 79
     9.3.1 u-momentum .................................... 79
     9.3.2 v-momentum .................................... 80
  9.4 South-east triangular cut-cell ............................ 81
     9.4.1 u-momentum .................................... 81
     9.4.2 v-momentum .................................... 82
  9.5 South-east pentagonal cut-cell ............................ 83
     9.5.1 u-momentum .................................... 83
     9.5.2 v-momentum .................................... 85
  9.6 Integration areas needed for the shear stress .......... 86
     9.6.1 Stress force .................................... 86
     9.6.2 South trapezoidal cut-cell ......................... 90
     9.6.3 South-east triangular cut-cells .................... 91
     9.6.4 South-east pentagonal cut-cell .................... 93

10 Numerical results 95
  10.1 Analytical ............................................ 95
  10.2 Horizontal channel ...................................... 96
     10.2.1 Channel aligned with grid ......................... 96
     10.2.2 Aligned with grid and shifted half a cell .......... 101
     10.2.3 Grid refinement ................................ 101
  10.3 Channel placed oblique to the grid ................. 104
     10.3.1 Channel placed under an angle of $\frac{\pi}{4}$ .... 104
     10.3.2 Channel shifted half a cell upward .......... 107
     10.3.3 Grid refinement ................................ 107
     10.3.4 Channel placed under an angle of $\arctan \frac{1}{2}$ .... 112
     10.3.5 Channel slightly shifted ...................... 112
     10.3.6 Grid refinement ................................ 115

11 Discussion and conclusion 119
  11.1 Discussion ............................................ 119
  11.2 Conclusion and further action ..................... 120
Bibliography 121
Part I

Overview of immersed boundary methods
Chapter 1

Introduction

An (Cartesian grid-based) immersed boundary method is a methodology for dealing with a body which does not necessarily have to fit conform a Cartesian grid, see for example Fig. 1.1. If that’s the case, the solid boundary will cut through this grid. Because the grid does not conform to the solid boundary, imposing the boundary conditions will require modifying the governing equations in the vicinity of the boundary. What these modifications exactly are, will be treated in the subsequent sections. The objective of this thesis is to find an immersed boundary method that can simulate unsteady viscous turbulent flows with high Reynolds numbers and also works in three dimensions. And if possible, this method should also be able to simulate moving boundaries. This method should be implementable on a Cartesian grid with a staggered positioning of the discrete variables.

Figure 1.1: Immersed boundary illustration; Eulerian mesh ($\vec{x}$) and Lagrangian mesh ($\vec{x}_k$) [53].

1.1 General considerations

One of the advantages of using an immersed boundary method is that grid generation is much easier, because a body does not necessarily have to fit conform a Cartesian grid. Another benefit is that grid complexity and quality are not significantly affected by the
complexity of the geometry when carrying out a simulation on a non-boundary conforming Cartesian grid. Also, an immersed boundary method can handle moving boundaries, due to the stationary non-deforming Cartesian grid. As a result of the above remarks, an immersed boundary method uses less memory and CPU compared to the usual method, a body fitted grid and the thereby belonging transformations. In comparison with structured curvilinear body-fitted grids, Cartesian grids reduce the per-grid-point operation count due to absence of additional terms associated with grid transformations. When comparing to unstructured curvilinear body-fitted grids, Cartesian grid-based IB methods are amenable to powerful line-iterative techniques and geometric multigrid methods, leading to a lower per-grid-point operation count. Also multi-phase and multi-material problems, where the interface is between different materials, can be regarded as immersed boundary problems.

A disadvantage is that imposing of the boundary conditions is not straightforward compared to the traditional methods. Also, the ramifications of the boundary treatment on accuracy and conservation properties of numerical schemes are not trivial. Another drawback is the following. Alignment between grid lines and body surface in boundary-conforming grids allows for better control of the grid resolution in the vicinity of the body and this has implications for the increase of grid size with increasing Reynolds numbers. However, a substantial fraction of grid points can be inside the solid body, i.e. where the fluid flow equations need not to be solved.
Chapter 2

Categories of IB methods

The way the boundary conditions are imposed on the immersed boundary determines for the most part how an IB algorithm will look like. It is also what distinguishes one IB method from another.

The simulation of a viscous incompressible flow past a body is described by the Navier-Stokes equations (the governing equations)

\[ \rho \left( \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) + \nabla p - \mu \Delta \mathbf{u} = 0 , \]

\[ \nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega_f \quad \text{and} \]

\[ \mathbf{u} = \mathbf{u}_\Gamma \quad \text{on } \Gamma_b , \]

where \( \mathbf{u}(\mathbf{x}, t) \) is the fluid velocity and \( p(\mathbf{x}, t) \) is the pressure. The coefficients \( \rho \) and \( \mu \) are the constant fluid density and viscosity, respectively. The solid body occupies the domain \( \Omega_b \), with boundary denoted by \( \Gamma_b \), and the surrounding fluid domain denoted by \( \Omega_f \).

In an IB method, equation (2.1) will be discretized on a non-boundary conforming Cartesian grid and the boundary condition will be imposed indirectly through modifications of equation (2.3). In general, the modification takes the form of a forcing function in the governing equations that reproduces the effect of the boundary. There is also another approach available: the so-called cut-cell approach. The first category will be explained first in this section. The second category will be shortly described further on in this section.

Introducing a forcing function, leads to a division of IB methods into two groups \textsuperscript{[42]}, namely continuous forcing and discrete (or direct) forcing. Sometimes these approaches are called diffuse respectively sharp interface methods. In the first approach, the forcing function, denoted here by \( f \), is included into the momentum equation leading to the equations
\begin{equation}
    \rho \left( \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) + \nabla p - \mu \Delta \mathbf{u} = \mathbf{f}, \quad (2.4) \\
    \nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega_f \quad \text{and} \quad (2.5) \\
    \mathbf{u} = \mathbf{u}_\Gamma \quad \text{on } \Gamma_b, \quad (2.6)
\end{equation}

which are then applied to the entire domain \((\Omega_f + \Omega_b)\). Many methods have been developed for choosing the external body force \(\mathbf{f}\). Some of these methods will be described further on in this report. After choosing an appropriate forcing function, the equations are subsequently discretized on a Cartesian grid, and the equations are solved in the entire domain.

In the discrete forcing approach, the governing equations (2.4)-(2.5) are discretized on a Cartesian grid neglecting the immersed boundary, resulting in a set of discretized equations. After that, the discretization in the cells near the IB is adjusted to account for their presence, i.e. the grid points in the vicinity of the immersed boundary will be computed using an interpolation scheme.

In the cut-cell approach the boundary conditions at the immersed boundary are not imposed by a forcing function. Instead, this approach requires truncating the Cartesian cells at the immersed boundary to create new cells which conform to the shape of the surface. This reshaping may result in very small cells, which has a negative impact on the numerical stability. One remedy for this problem is using a cell-merging strategy. In Chapter 5, this method will be described in detail.
Chapter 3

Continuous forcing approach

Several methods make use of a continuous forcing approach. Examples are methods for elastic boundaries, methods for rigid boundaries (i.e. boundaries which are fixed) and the distributed Lagrange multiplier method, see for instance [24] for more methods belonging to this category. These three methods will be discussed in more detail.

3.1 Elastic boundaries

Peskin introduced in 1972 the concept of immersed boundary methods [48, 49], where he used this method to compute flow patterns around heart valves. Peskins method is a mixed Euler-Lagrangian finite-difference method for computing the flow interaction with a flexible immersed boundary. In this method the fluid flow is governed by the incompressible Navier-Stokes equations and these are solved on a stationary Cartesian grid. The IB is represented by a set of massless elastic fibres and the location of these fibres is tracked in a Lagrangian fashion by a collection of massless points that move with the local fluid velocity

\[
\frac{\partial X}{\partial t}(s,t) = u(X(s,t)).
\]  

(3.1)

Here, the boundary configuration is described by the curve \(X(s,t)\), where \(s\) is a parameter chosen in such a way that a given value of \(s\) represents a given physical point of the boundary for all times \(t\).

Fig. 1 shows such a configuration.

Peskin defines the force density, \(f(x,t)\), by a \(\delta\)-function layer that represents the force applied by the immersed boundary to the fluid. The problem of this definition is that the location of the fibres does not generally coincide with the nodal points of the Cartesian grid. Therefore, the forcing is distributed over a band of cells around each Lagrangian point (see Fig. 3.1(a)) and this distributed force will be used in the momentum equations.
of the surrounding nodes. By replacing the sharp $\delta$-function with a smooth distribution function, denoted by $d$, this new forcing function will be more suitable for use on a discrete mesh. Due to the fibres, the forcing at any grid point $x$ is then given by

$$ f(x, t) = \int_{\Gamma_b} F(s, t) \delta(x - X(s, t)) \, ds. \quad (3.2) $$

The boundary force, $F(s, t)$, on the particular segment at time $t$ is determined by the boundary configuration at time $t$

$$ F(s, t) = S(X(s, t), t), \quad (3.3) $$

where the $S$ satisfies a generalized Hooke’s law if the boundary is elastic.

There are more approaches for the distribution function developed over the years and some of them are shown in Fig. 3.1(b).

Some of these authors, Lai et al. [34] and Saiki et al. [50], will be treated next.

### 3.2 Rigid boundaries

The first approach for rigid boundaries is called virtual boundary method, used by Goldstein et al. [21]. The main idea of the virtual boundary method is to treat the body surface as a virtually existent boundary embedded in the fluid. This boundary applies force on the fluid so that the fluid will be at rest on the surface (no-slip condition). Let’s denote the boundary $\Gamma_b$ by $\{X^e(s) : 0 \leq s \leq L_b\}$. The force $F(s, t)$ on the boundary is determined by
3.2. RIGID BOUNDARIES

the requirement that the fluid velocity \( \mathbf{u}(x, t) \) should satisfy the no-slip condition on the boundary

\[
\rho \left( \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) + \nabla p = \mu \Delta \mathbf{u} + \int_{\Gamma} \mathbf{F}(s, t) \delta(x - X^e(s)) \, ds, \tag{3.4}
\]

\[
\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega_f, \tag{3.5}
\]

\[
\mathbf{u} = \mathbf{u}_\Gamma \quad \text{on } \Gamma_b \quad \text{and} \quad (3.6)
\]

\[
0 = \mathbf{u}(X^e(s), t) \equiv \int_{\Omega} \mathbf{u}(x, t) \delta(x - X^e(s)) \, dx. \tag{3.7}
\]

Since the body force is not known \textit{a priori}, it must be calculated in some feedback way in which the velocity on the boundary is used to determine the desired force. In the virtual boundary formulation, the force is expressed as

\[
\mathbf{F}(s, t) = \alpha \int_0^t \mathbf{u}(s, \tau) \, d\tau + \beta \mathbf{u}(s, t), \tag{3.8}
\]

where \( \mathbf{u} \) is the fluid velocity at these surface points. The particular form given in Eq. (3.8), can be seen as a PI controller, where P stands for the proportional part and I for the integrating part. When there is also a differentiating part included, the formula can be described as a PID controller. This construction of Eq. (3.8) seems reasonable given that this formula is a feedback mechanism.

When \( \alpha \) and \( \beta \) are chosen negative and large enough in magnitude, then \( \mathbf{u} \) will stay close to its prescribed value. To avoid interpolating the velocity field from grid points to the boundary points, Goldstein \textit{et al.} let the boundary points coincide with grid points. However, in order to generate a smooth surface rather than a step-like surface, the boundary force is multiplied by a narrow Gaussian distribution so that the nearby grid points can receive a part of the force influences. Although this local smoothing will blur the location of the surface within one grid cell, the method can produce promising results if sufficient spatial resolution is used \cite{34}.

A disadvantage of feedback forcing is that this not only may induce spurious oscillations but also restricts the computational time step associated with numerical stability. Especially for highly unsteady flows, stability problems arise due to considerable stiffness.

Saiki \textit{et al.} \cite{50} extended this feedback forcing approach, such that the spurious oscillations caused by the applied feedback forcing term at the boundary are eliminated. They modified Eq. (3.8) into the area-weighted average function

\[
\mathbf{F}(x_s, \tau) = \alpha \int_0^t \left[ \mathbf{u}(x_s, \tau) - \mathbf{v}(x_s, \tau) \right] \, d\tau + \beta \left[ \mathbf{u}(x_s, t) - \mathbf{v}(x_s, t) \right], \tag{3.9}
\]

where the velocity of the body itself is controlled by specifying \( \mathbf{v} \) at the boundary points. By employing this function a better interpolation of the fluid velocity at the boundary
points is developed. They used fourth-order central difference approximations. The use of finite-differences avoids the appearance of spurious flow oscillations at the boundary. Also an appropriate distribution of nodal boundary forces at these grid points has been made using this formula. If the body moves, i.e. $v \neq 0$, then the position of the boundary points at each time step is computed by integration of $v = \frac{dx}{dt}$. They showed that the feedback-force IB method is capable of handling the solid boundary problems including also moving boundaries.

Another approach, used by Lai et al. \[34\], can be viewed as a special version of the virtual boundary method. In order to simulate the flow around a rigid boundary using the immersed boundary method, it is necessary to allow the boundary to move a little bit rather than to be fixed. As long as the immersed boundary $X(s, t)$ stays close to the body surface $X^e(s)$, the equations (3.4)-(3.7) can be rewritten as

$$\rho \left( \frac{\partial u}{\partial t} + u \cdot \nabla u \right) + \nabla p = \mu \Delta u + \int_{\Gamma} F(s, t) \delta (x - X(s, t)) \, ds ,$$  
(3.10)

$$\nabla \cdot u = 0 \text{ in } \Omega_f,$$  
(3.11)

$$u = u_\Gamma \text{ on } \Gamma_b \text{ and}$$  
(3.12)

$$\frac{\partial X(s, t)}{\partial t} = u(x(s, t), t) \equiv \int_{\Omega} u(x, t) \delta (x - X(s, t)) \, dx.$$  
(3.13)

The position of the wall follows from Eq. (3.13). Now an appropriate forcing term $F(s, t)$ is needed to make sure that the boundary points will stay close to the body surface. One straightforward choice is

$$F(s, t) = \kappa (X^e(s) - X(s, t)),$$  
(3.14)

where $\kappa$ is a positive spring constant such that $\kappa \gg 1$. The interpretation of Eq. (3.14) is that the boundary points $X$ are connected to fixed equilibrium points $X^e$ with a very stiff spring whose stiffness constant is $\kappa$. So if the boundary points move away from the desired location, the force on the spring will pull these boundary points back. Thus, as time goes on, we can expect that the boundary points will always be close to the one used for the virtual boundary method in Eq. (3.8). If we choose $\beta = 0$ in eq. (3.8), we can see that Eq. (3.14) is a special version of the previous approach.

Another method, called a penalty method, which has been developed by Khadra et al. \[30\], is the following. The idea is that the entire flow is assumed to occur in a porous medium and is therefore governed by the Navier-Stokes-Brinkman equations. These equations contain an additional term of volume drag, called Darcy drag, with respect to the classical Navier-Stokes equations. This Darcy drag accounts for the action of the porous medium on the flow. The Navier-Stokes-Brinkman equations are given by
3.3. DISTRIBUTED LAGRANGE MULTIPLIER METHOD

\[ \rho \left( \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) + \nabla p - \mu \Delta \mathbf{u} + \mathbf{F}'(t) = \mathbf{f}' , \]  
\[ \nabla \cdot \mathbf{u} = 0 \text{ in } \Omega_f \text{ and } \]  
\[ \mathbf{u} = \mathbf{u}_r \text{ on } \Gamma_b, \]  

(3.15) \hspace{1cm} (3.16) \hspace{1cm} (3.17)

where \( \mathbf{f}' \) is a general body force, for example gravity, and \( \mathbf{F}' \) is defined as

\[ \mathbf{F}'(t) = \frac{\mu}{K} \mathbf{u}(t). \]  

(3.18)

Here \( K \) is the permeability of the medium and is (theoretically) defined as infinity or zero for fluid and solid regions, respectively. The force therefore activates only within the solid, driving the velocity field to zero. Again there is similarity between this formulation and the force in Eq. (3.8). Choosing \( \alpha = 0 \) and \( \beta = \mu/K \) in Eq. (3.8) leads to the forcing approach introduced by Khadra et al..

Based on direct (momentum) forcing on Eulerian grids, Su et al. [54] proposed a new implicit force formulation on the Lagrangian marker to ensure exactly satisfaction of the no-slip boundary condition at the immersed boundary. A mixture of Eulerian and Lagrangian variables is adopted, where the solid boundary is represented by discrete Lagrangian markers embedding in and exerting forces to the Eulerian fluid domain. Interactions between the Lagrangian markers and the fluid variables on the fixed Eulerian grid are linked by a simple discrete delta function. The boundary forces are first computed on the Lagrangian markers and then distributed to the Eulerian grid, using a discrete delta function. Their numerical experiments show that the stability limit is not altered by their proposed formulation. Despite of using second-order accurate Adams-Bashfort and Crank-Nicolson, their numerical scheme is degraded to 1.5 order of accuracy.

A disadvantage of the methods developed for rigid boundaries, is that user-specified parameters in forcing are necessary with associated stability constraints.

3.3 Distributed Lagrange multiplier method

The distributed Lagrange multiplier method (DLM), proposed by Glowinski et al. [20], uses a variational principle (finite element) as framework. The idea is to introduce Lagrange multipliers (i.e. body force) on the immersed rigid body to satisfy the no-slip condition. After that, a finite element approximation is used on the rewritten problem with Lagrange multipliers. In this new formulation, the use of projection (also known as predictor-corrector) is needed. First, the fluid velocity will be approximated using the momentum equation and afterwards the velocity will be corrected by the related unknown pressure using the incompressibility condition.
3.4 Immersed interface method

A method that is developed for elastic membranes, is the immersed interface method (IIM), Lee et al. [36]. IIM uses the same equations as introduced in the section about elastic boundaries, namely (3.1) - (3.3). In the IIM, the boundary force/force strength, $F$ defined as in Eq. (3.2) where the $\delta$-function is used, is decomposed into tangential and normal components. The interface is tracked explicitly in a Lagrangian manner. The tangential component of the force is included in the momentum equation as an explicit term and the explicit normal boundary force is implemented into the governing equations in terms of a pressure jump condition across the interface [55]. Although this method allows discontinuities, it is categorized in the continuous forcing approach, because the concept of IIM shows similarities with the other methods in this category.

3.5 Solving linear system

For the two methods to be treated in this section, the force is computed implicitly by solving a linear system. Implicitly solving the forcing means in this situation, that the forcing will be determined using the prescribed boundary condition at the newest time step. 

Le et al. [35] developed an approach which combines the capability of the original immersed boundary method with an implicit forcing scheme. The forcing term at the boundary is calculated by solving a system of equations derived from the numerical scheme. Once the force is determined at the boundary, the immersed boundary method is employed with a second-order projection (pressure-correction) method to compute the solutions of the Navier-Stokes equations.

Taira et al. [55] presented a new formulation of the immersed boundary with a structure algebraically identical to the traditional fractional step method. To satisfy the no-slip constraint, they applied a boundary force at the immersed surface. Their method can deal with incompressible flow over bodies with prescribed surface motion. Taira et al. constructed the method such that it preserves symmetry and positive definiteness to efficiently solve for the flow field. The boundary force is determined implicitly without any constitutive relations for the rigid boundary formulation, resulting in a relatively high Courant-Friedrichs-Lewy (CFL) number.

3.6 General considerations

An advantage of the continuous forcing approach is that the above described methods are independent of the underlying spatial discretization in contrast to methods that are based on a discrete forcing approach. Therefore, this approach can be implemented into an existing Navier-Stokes solver with relative ease. A disadvantage of these methods is that
the smoothing of the forcing function inherently leads to an inability to provide a sharp representation of the immersed boundary and therefore these methods are not useful for high Reynolds number flows. Another drawback of the continuous approach is that they all require the solution of the governing equations inside immersed body. With increasing Reynolds numbers the proportion of grid points inside the IB also increases.
Chapter 4

Direct forcing approach (discrete approach)

The discrete approach is better suited for higher Reynolds numbers, due to imposing the velocity boundary conditions at the immersed boundary, without introducing or computing any forcing term. The methods that will be discussed here in detail are the direct-forcing method and extensions of it, like the ghost-cell method, and the hybrid Cartesian/immerged boundary method. The governing equations are most of the time discretized as follows. A second-order Adams-Bashforth scheme is employed for the convective terms, while the diffusion terms are discretized using an implicit Crank-Nicolson scheme. This eliminates the viscous stability constraint, which can be quite severe in simulation of viscous flows [69].

4.1 General idea

The incompressible flows governed by the non-dimensional Navier-Stokes equations, including the body force term are given by

\[
\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p - \frac{1}{Re} \Delta \mathbf{u} = \mathbf{f},
\]

\[
\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega_f \quad \text{and}
\]

\[
\mathbf{u} = \mathbf{u}_r \quad \text{on } \Gamma_b.
\]

The forcing term \( \mathbf{f} \) functions as a velocity corrector for the grid points inside the IB. This term is prescribed at each time step to establish the desired boundary moving velocity \( \mathbf{V}_{ib} \).

For a time-stepping scheme, this force can be expressed as

\[
\frac{u_i^{n+1} - u_i^n}{\Delta t} = RHS_i + f_i.
\]
CHAPTER 4. DIRECT FORCING APPROACH (DISCRETE APPROACH)

where the upper index of \( u \) indicates the time and the lower index the space. If \( f_i \) must yield \( u_{i}^{n+1} = \frac{V_{ib}^{n+1} - u_{i}^{n}}{\Delta t} - RHS_i \),

Thus the body force is defined like

\[
    f = \begin{cases} 
        (u \cdot \nabla) u + \nabla p - \frac{1}{Re} \Delta u + \frac{1}{\Delta t} (V_{ib} - u^n), & \text{near } \Gamma_b; \\
        0, & \text{elsewhere.}
    \end{cases}
\]

This approach only holds when the immersed boundary coincide with the grid. In general this is not the case and the question arises: how to prescribe the boundary condition? This algorithm will be explained using the idea of Balaras et al. \[1\].

1. First, compute \( u^* \) in the discretized Navier-Stokes equations and omitting the forcing term \( f^{n+1} \). The resulting \( u^* \) will not satisfy the boundary conditions on the immersed boundary.

2. Then, compute \( f^{n+1} \) from (4.1). The value of the velocity \( V_{ib} \) on the forcing points is computed using an interpolation procedure. These forcing points can be placed outside and inside the body, which is used in a ghost cell method, see §4.3.

3. Compute \( u^* \) from the discretized Navier-Stokes equations with the forcing term. The resulting velocity will satisfy the desired boundary conditions on the immersed boundary.

4. Compute the pressure using the Poisson equation.

5. Update the velocity and pressure.

6. Go to step 1.

4.2 Direct forcing

The (spectral) method of Mohd-Yusof \[43\] uses a forcing term, which is determined by the difference between the interpolated velocities in the boundary points and the desired (physical) boundary velocities. The forcing term, generated in this manner, thus directly compensates the errors between the calculated velocities and the desired velocity profile on the body surface. If the boundary is stationary, a tangentially opposite direction flow field to the external layer flow field is specified in the internal layer inside the immersed boundary. Where the internal/external layer is defined in a layer of grid points immediately inside /outside the immersed boundary. The force is thus determined by pairing the velocity at the internal point to the velocity at the external point with a weighted linear interpolation, to enforce the desired tangential velocity on the boundary, i.e. the method mirrors the velocity field across the immersed boundary. An example of such a pair is illustrated in Fig. 4.1 between Point 1 and Point 2 \[70\].
4.2. DIRECT FORCING

Methods based on mirroring satisfy the velocity boundary condition with the accuracy of the interpolation method. Kim et al. [32] reported that in practice, the accuracy may be lower, because of incompatibility with the continuity equation. Internal treatment is required for this method to alleviate the problem of spurious oscillations near the boundary. Due to the reversed velocity field, problems with the mass conservation arise in the boundary cell [40].

Figure 4.1: Schematic interpolation of Mohd-Yusof method. [70]

4.2.1 Improvements

Fadlun et al. [11] further implemented the discrete-time forcing approach, as suggested by Mohd-Yusof [43] to a three-dimensional finite-difference method on a standard marker-and-cell (MAC) staggered grid and showed that the approach was more efficient than feedback forcing. There is a difference between these approaches. In Fadlun et al. the velocity at the first grid point external to the body (external forcing) is obtained by linearly interpolating the velocity at the second grid point (which is obtained by directly solving the Navier-Stokes equations) and the velocity at the body surface, which conceptually corresponds to applying the momentum forcing inside the flow field. On the other hand, momentum forcing is applied only on the body surface or inside the body in Mohd-Yusof’s method (internal forcing). The interpolation direction (i.e., the direction towards the second grid point) taken by Fadlun et al. is either the streamwise ($x$) or the transverse direction ($y$), but the choice of interpolation direction is arbitrary, which can generate problems in complex configurations [1]. This approach thus does not rely on the mirrored velocity field in the solid region, used by Mohd-Yusof. Therefore, the accuracy of the interpolation remains the same. The linear interpolation formula used here is equivalent to an implicit forcing formulation. Note, this method is sometimes called the standard reconstruction method, i.e., a combination of linear interpolation with the standard mass conservation [27], see Fig. 4.4(a).

Since the velocity boundary condition is enforced with implicit forcing, there is no severe limit on the time step. Another advantage is that the velocity components from the regions across the immersed boundary are decoupled [27]. Fadlun et al. did not include a special constraint on the mass conservation. So the original method for enforcing mass
conservation results in coupling between the solutions across the immersed boundary via
discretized operators, which disobeys the pressure decoupling constraint. Therefore, this
method fails to predict the flow fields correctly. Another drawback is that the velocity
boundary condition is exactly satisfied in the momentum solution step, but a finite error is
introduced during the projection to the divergence-free velocity field. This is because the
velocity equal to the intermediate velocity is not enforced at the immersed boundary\cite{29}.
This method works well for bodies that are aligned with the grid lines. For geometrically
complex immersed boundaries, however, the choice of the reconstruction direction may not
be unique, because often more than one grid line passing through a near-boundary node
may intersect the boundary\cite{7}.

Figure 4.2: Treatment of the interface cells; forcing is applied on the filled circles.\cite{1}.

Balaras\cite{1} proposed a better reconstruction scheme, based on the method of Mohd-
Yusof and Fadlun et al., which performs the reconstruction along the well-defined line
normal to the body. By taking a provisional explicit step (advancing both viscous and
diffusive terms with an Adams-Bashforth scheme for example) an inversion of a large
sparse matrix can be avoided. The algorithm eliminates the ambiguities associated with
interpolation along grid lines, like in Fadlun et al.\cite{11}, see Fig. 4.2. However, this method is
restricted to flows with immersed boundaries that are aligned with one coordinate direction,
for example two-dimensional or axisymmetric shapes\cite{19}.

Based on the ideas of Balaras, Gilmanov et al.\cite{19} developed a new reconstruction
scheme, which is applicable to arbitrarily complex, three-dimensional immersed bound-
daries. The proposed methodology maintains a sharp fluid/body interface by discretizing
the body surface using an unstructured, triangular mesh. The solution in the vicinity
of boundary nodes is reconstructed via linear interpolation along the local normal to the
body. Unfortunately, this method is only applicable to stationary bodies of simple (convex)
shape\cite{18}. 

(a) Fadlun et al.  (b) Balaras.
4.2. DIRECT FORCING

An improved version of Mohd-Yusof’s method has been made by Zhang et al. They improved the method of Mohd-Yosuf by implementing a bilinear interpolation/extrapolation function to interpolate the direct force, which ensures more accurate boundary forcing expressions. In Fig. 4.3 a sketch of the interpolation/extrapolation is shown between immersed boundary points and the grid points near the immersed boundary. This function is the same as the function introduced by Saiki et al. For the method of Mohd-Yusof, only the tangential component of the interpolated velocity was used for specifying the force, while the method of Zhang et al. included both the tangential and the normal velocity components. The advantages are that it is easy for coding and the number of boundary points can be increased independently of the computational grid. Therefore, the accuracy in the vicinity of the boundary can be enhanced without using higher-order schemes, or arranging complicated grids near the boundary. A drawback is that the proposed Saiki et al. interpolation/extrapolation method needs user-defined parameters as input.

Choi et al. made a finite-volume approach, which is based on Fadlun et al. They developed a more general immersed boundary method that is valid at all Reynolds numbers and is suitable for implementation on arbitrary grid topologies. Choi et al. introduced the concept of tangency correction by decomposing the velocity into tangential and normal component along the outward normal direction to the immersed surface. The tangential velocity component is expressed as a power-law function of the wall normal distance. However, the choice for using a power-law function seems somewhat arbitrary.

The well-known pressure Poisson equation approach is applied by Sheu et al. to eliminate the pressure gradient terms from the momentum equations by performing a curl operator to the momentum equations. They propose a quadratic interpolation scheme. First step in their method is to calculate the intermediate velocity, then correct this intermediate velocity using the proposed quadratic interpolation method. After that, they impose the forcing term $F$ in the solid-fluid cells and finally the real velocity will be computed.
4.2.2 Fulfilling conservation laws

The problem of most of the treated methods, is that they do not fulfil (some of) the conservation laws, such as conservation of the wall condition. The treated immersed boundary methods violate the wall condition in the discrete equation system during time-advancement. This problem arises from the inconsistency of the pressure with the velocity interpolated to represent the solid wall. Therefore, Ikeno et al. [25] developed a scheme to maintain the consistency between pressure and velocity for the immersed boundary method to achieve precisely the desired wall condition, like the no-slip wall condition or a non-zero velocity at the wall. They accomplished this by deriving a modified pressure equation based on the interpolated pressure gradient. Ikeno et al. uses a two-point forcing strategy, i.e. in the fluid as well as in the body region forcing is applied. Forcing in the body is employed, because extrapolating the velocity at the grid points nearest to the wall increases the accuracy of the velocity gradient. They demonstrated theoretically the conservation of the wall condition, mass, momentum and energy. Due to satisfaction of all the conservation laws, this method works fine for high Reynolds numbers even in combination with turbulent flows. Ikeno et al. showed that, unlike their method, Fadlun et al. [11], Ye et al. [69], Tseng et al. [58] and Kim et al. [32] do not satisfy the conservation properties of the wall condition.

Domenichini [8] found out that direct forcing schemes, treated so far, are not able to satisfy the impenetrability condition on the fixed and moving wall. This fact appears to be strictly related to the use of fractional step methods. He suggests that improvements can be obtained with the iterative solution of the irrotational part of the flow (flow with vanishing curl), when spectral methods are used and local modification of the discrete differential operators are difficult to be implemented.

4.2.3 Conservation of mass

In some approaches, like Fadlun et al., mass conservation at the immersed boundary is satisfied by the velocity fields both in the fluid and solid regions, Fig. 4.4(a). The pressure is coupled by this construction. In this case the non-physical velocity field in the solid becomes important because it affects the pressure and velocity distribution through the velocity divergence across the immersed boundary. This issue can become more serious in the reconstruction methods (e.g. [11, 18, 19]), since treatment of the velocities at the first grid points into the solid region is undefined [27].

In general, as the interpolation scheme is defined without regard to the continuity equation, the velocities obtained from the interpolation scheme will not satisfy conservation of mass for the cell. This will cause the magnitude of the pressure in these mass conservation cells to slowly increase without bound, in other words a pressure build-up. However, as this pressure does not appear in any discrete momentum equation (as a result of the two-point pressure gradient stencil and the selection criteria for immersed boundary points),
4.2. DIRECT FORCING

The solution will not diverge. This pressure is therefore, completely decoupled from all other discrete variables \[44\].

![Figure 4.4: Different schemes for defining control volumes for mass conservation near the immersed boundary. \[27\].](image)

To alleviate the problem of incorrect pressure distribution along the immersed boundary, Li et al. \[37\] suggested to impose a zero normal gradient condition of pressure. This approach can be interpreted as an alternative way to enforce continuity since the Poisson equation is derived from the continuity equation.

![Figure 4.5: Implementation of the zero gradient pressure condition on a collocated grid \[37\].](image)

In Fig. 4.5, node A has two fluid neighbour nodes B and D. The zero normal pressure gradient condition on point P can be obtained by setting \(p_A = p_{P_1}\). Thus mass conservation at point P will be satisfied since there will be no flow across the boundary.

Muldoon et al. \[44\] among others, observed the non-physical and unbounded behaviour of the pressure at certain locations near the immersed boundary. They developed two new methods. The first is called PVR (pseudo velocity reduced) for non-moving immersed boundaries. For moving boundaries, the method is called CV (constrained velocity). In both methods, no distinction is made between the inside or outside of an immersed boundary. Muldoon et al. managed to maintain mass conservation in their methods, but due to using a pseudo-velocity the solution will not satisfy the prescribed velocity at the immersed boundary, but satisfy the pseudo-velocity.
The objective of Kang et al. [27] is to assess the accuracy and efficiency of the immersed boundary method to correctly predict the wall-pressure fluctuations in turbulent flows. This will be achieved by introducing additional constraints. Firstly, a compatibility for the interpolated velocity boundary condition related to mass conservation and secondly the formal decoupling of the pressure on this surfaces. Their starting point is the method of Fadlun et al. This approach is referred to as the reconstruction of the interpolation method. The immersed boundary-approximated domain method (IB-ADM) was developed, to satisfy the pressure decoupling constraint. This decoupling process allows discontinuous solutions across the interface and is similar to the jump condition used in the immersed interface method [36], and the ghost fluid method [12]. A schematic sketch of the IB-ADM is shown in Fig. 4.4(c). In the IB-ADM, the velocity equal to the intermediate velocity is enforced at the approximated boundary $\Gamma_a$ instead of $\Gamma_{IB}$. It is very important to satisfy the pressure decoupling constraint, because this assumptions leads to $\frac{\partial(p^k-p^{k-1})}{\partial n} = 0$ at $\Gamma_a$, i.e. (strict) mass conservation and no accumulation of the pressure error. When $\Gamma_A$ is very close to $\Gamma_{IB}$, the original condition is recovered. The linear interpolation is included with the effect of the pressure gradient term (revised linear interpolation method of [28]), resulting in a slightly reduced velocity error. Kang et al. showed that by satisfying the pressure decoupling constraint, the IB-ADM is successfully in handling very thin solid objects.

Kang et al. [28] developed an immersed boundary method that has sufficient accuracy for large eddy simulations (LES) at high Reynolds numbers with minimal increase of computational cost with respect to the simple underlying Cartesian grid solver. This approach is based on a finite difference method on a structured staggered grid. They considered strict and approximate mass conservation of the immersed boundary method. Two novel velocity reconstruction methods based on conservation of momentum are proposed, namely a revised linear interpolation method (RLIM) and a combination of quadratic and momentum interpolation method (QMIM). They use Crank-Nicolson for the diffusion term and a third-order Runge-Kutta method for the convective term. The non-physical behaviour of pressure near the immersed boundary was corrected by solving an additional Poisson equation after the velocity field has been obtained. To satisfy decoupling across the immersed boundary, they linearly interpolated the pseudo pressure and mass fluxes from the variables in the fluid region and the boundary condition. By interpolation both fluxes and applying the discretized divergence equation results in a formal discretization of the Poisson equation. After the pseudo pressure field is obtained, a divergence-free velocity field is computed. This approach is used for strict mass conservation. For approximate mass conservation, the interpolated flux and discretized divergence scheme is applied to only the RHS of the Poisson equation. From their experiments, Kang et al. concluded that strict mass conservation does not appear to be as cost-effective as the approximate mass conservation. There were some more simulations added to the previous paper [28], resulting in a revised version made by Kang et al. [29]. From the results, Kang et al. [28],
concluded that a fine mesh resolution or special treatment is necessary to get an acceptable pressure field when the linear interpolation method (LIM) (of Fadlun et al. [11]) for the velocity is employed. Also Kang et al. found out that the simple LIM has incompatibilities with the time marching scheme, which expressed itself as local error accumulated in the pressure field.

Sadly enough, no obvious improvement can be found in the results of Kang et al. [28] when mass conservation of virtual cells (discarded solid part) is taken into account [23]. Although the effect of the local pressure has been accounted for by using a few additional corrections (like in RLIM), the pressure field is still discontinuous. Therefore, the local error can be accumulated in the pressure field near the immersed boundary. This drawback is especially not suited for moving boundary problems [7]. From simulating a cylinder in a lid-driven cavity, Deng et al. [7] found out that RLIM is first-order accurate in space. So RLIM is not preferable to use.

Two novel implicit second-order accurate methods for simulating flow around three-dimensional, arbitrary stationary bodies is presented by Mark et al. [40], where the arbitrary immersed boundary is triangulated. The first one is the vertex-constraining IB (VCIB) method. A subset of the triangle vertices are used as control points in which the boundary conditions are applied. A trilinear interpolation is applied to ensure the correct velocity at the immersed boundary. The mirroring IB (MIB) method is the second one. This implementation mirrors the interior immersed boundary node along the normal of the triangulated immersed boundary to a fictitious point in the flow domain, such that it becomes exactly defined at the immersed boundary. Trilinear interpolation is employed to interpolate the velocity to the mirrored velocity point outside the immersed boundary. Due to the fictitious velocity field inside the solid body, a mass flux over the immersed boundary is generated, which is non-physical. Therefore, the fictitious velocity field is excluded in the continuity equation, resulting in no mass flux over the immersed boundary. Hence, the presence of the immersed boundary is accounted for both in the pressure correction equation and in the momentum equation. Therefore, the pressure correction equation generates no driving pressure force over the immersed boundary in the momentum equation. Due to this, no regular (Neumann) boundary condition has to be employed at the immersed boundary, which previous methods [37, 58] employ to prevent a pressure force over the immersed boundary. By employing a pressure boundary condition at the immersed boundary unnecessary information is inserted into the pressure correction equations, which could lead to non-physical solutions. Instead the mass conservation over the immersed boundary is solved from a physical point of view, which generates a more physical solution with a faster convergence rate of the solution. They showed that for the mirroring method the resulting coefficients lead to a well-posed and diagonally dominant system which can be efficiently solved with a preconditioned Krylov subspace solver. Mark et al. stated, that the mirroring method is independent of the number of vertices in the triangulation, because it uses control planes, spanned by the closest triangles, instead of control points. Therefore, the mirroring method can track more general immersed boundaries and the size of the tri-
angles of the triangulation does not play a role. From simulations, they concluded that the mirroring method is more stable and has faster convergence than the VCIB method. The convergence rate is increased compared to earlier immersed boundary methods, because there exists no mass flux over the immersed boundary.

Concerning satisfying the mass continuity, Shinn et al. [52] uses a similar approach as Mark et al. For ghost pressure, two practices are possible. In the first practice, as used in [19, 58], the ghost pressures are extrapolated from inside by the mirror reflection procedure using a Neumann condition. However, this practice leads to mass fluxes across the solid boundaries and mass errors in the ghost cells, also it leads eventually to oscillatory flow field or even numerical divergence. They observed that even when compensating these errors for by mass sources/sinks, oscillations occur in the velocity field. Shinn et al. suggested another approach, which is to directly satisfy the continuity equation for the ghost cells also and determine the pressure the in usual way through the Poisson equation. However, the mass errors should not be evaluated using the ghost velocities, because they are not solutions of the momentum equations. Instead, the boundary velocities must be directly substituted and the ghost velocities (outside the boundary) must be used only for the momentum equations. This approach preserves global continuity and avoids mass source/sinks in ghost cells.

Ikeno et al. [25] and Taira et al. [55] modified the projection step so that the interpolation formula is satisfied after the projection, i.e the velocity boundary condition is exactly satisfied and at the same time the conservation of mass is guaranteed.

### 4.2.4 Treatment of the interior of the body

Fadlun et al. [11] suggested that there are three possible ways of treatment for the flow interior of the body with stationary immersed boundaries. The first is to apply the forcing at every point inside the body without any smoothing, as suggested by Saiki et al. [50]. This is equivalent to imposing the velocity distribution inside the body with the pressure that adjusts accordingly. The second is to leave the interior of the body free to develop a flow without imposing anything. This is to reverse the velocity at the first point inside the body in such a way that results in the desired velocity on the boundary. Extensive testing of these procedures has been performed to check the influence of the internal treatment of the body on the accuracy and the efficiency of the scheme. Fadlun et al. [11] and Iaccarino et al. [24] found that, when using the direct forcing, there is essentially no influence. Therefore, depending on the particular flow, the easiest treatment can be used.

Previous studies indicated that for stationary boundary problems, different treatments inside the solid body did not affect the external flow. However, the relationship between the internal treatment of the solid body and external flow for moving boundary problems was not studied extensively and therefore Liao et al. [38] investigated this. Their approach is based on direct momentum forcing on a Cartesian grid utilizing a combination of interpolation at fluid nodes adjacent to the solid body [11] and solid body forcing at nodes,
which had been part of the solid region in current time step. They showed that it is impor-
tant to use solid body forcing in computing flows with moving objects. Significant lower
amplitude oscillations in computed lift and drag coefficients are obtained compared with
those without solid body forcing strategy. A note hereby however is that Liao et al. only
used experiments at low Reynolds numbers.

4.2.5 Mass source/sink approach

Kim et al. [32] used a second-order linear or bi-linear interpolation schemes within a finite-
volume context, using a staggered mesh. Their method is actually an explicit variant of
the direct forcing method of Mohd-Yusof [43]. Flows over immersed complex geometries
were simulated. They introduced the use of a mass source/sink in the continuity equation
and a momentum forcing to guarantee the no-slip boundary condition on the immersed
boundary, and to satisfy the continuity for the cell containing the immersed boundary.
When the mass source/sink term was employed, Kim et al. found that the quality of
the solution was improved with respect to Mohd-Yusof’s method. Also the non-physical
solution was corrected. However, this approach is formulated in a stepwise approximate
manner, i.e. the grid points fall on the immersed boundary. Huang et al. [23] showed that
this approximation may degrade the quality of the solution. However, because adding the
force density term should depend on the velocity at the time step being solved for, and
not on the pseudo velocity, the resulting velocity will not satisfy the desired velocity at
the boundary. Another issue is that due to the added mass source term, the velocity obtained
by using the pressure determined from the pressure-Poisson equation will not be divergence
free [44].

Huang et al. [23] improved the concept of mass forcing and applied the method. They
derived a more accurate formulation of the mass source/sink by considering mass conserva-
tion of the virtual cells (discarded solid part) in the fluid crossed by the immersed boundary.
By introducing face-centred velocities of the virtual cells, this will be accomplished. If the
momentum-forcings are calculated implicitly, like in Fadlun et al. [11], a large sparse ma-
trix is created for a complicate interpolation scheme. This results in a significant increase
of computing cost. Therefore, Huang et al. use instead a prediction step by the forward
Euler explicit scheme.

4.2.6 Hybrid Cartesian/immersed boundary method

Gilmanov et al. [18] designed a method, that is applicable to arbitrarily complex moving
bodies, using a second-order hybrid non-staggered/staggered grid approach. For the inter-
polation scheme a quadratic interpolation near the body is used. The reconstruction of
the solution near the boundary is carried out by interpolation along the normal to the body
(based on [19]). The external boundary velocities are set by a Dirichlet boundary condi-
tion such that an interpolation of the velocity along the normal of the immersed boundary
fulfils the no-slip boundary condition. The Neumann pressure boundary condition is set in a similar way. The boundary condition for the pressure is applied explicitly, and the boundary condition is discretized with first-order accuracy, directly in the exterior grid points.

4.2.7 Physical Virtual Model

Another approach that can be categorized into direct forcing is the Physical Virtual Model (PVM), proposed by Silva et al. The name for this model comes from the fact that it is based only upon the laws of conservation. The PVM model is based on the evaluation of the various terms (such as acceleration force and pressure force) in the momentum equations at the rigid boundary. All these components are computed at the rigid boundary using a second-order Lagrange polynomial approximation and based on the solutions at the previous time level. Once the momentum forcing term has been computed at the boundary, the original IB method is applied to compute the velocity and pressure fields. There are no ad hoc constants in this model and a special algorithm to capture the neighbouring grid points of the immersed interface is not required. However, this method is first-order accurate. These forcing formulations are simple to implement but require a small time-step to maintain the stability. While the approach of Silva et al. is ideally simple, the calculations of momentum forcing at the boundary points are quite complex.

Deng et al. developed a method for simulating flows over complex immersed, moving boundaries. The method is based on a second-order central difference scheme on a staggered mesh together with a two-step fractional method. A momentum forcing is added at the body boundaries and also inside the body to satisfy the no-slip boundary condition. The immersed boundary is represented by a set of discrete control points (Lagrangian points). The Lagrangian forcing is calculated, using the method in PVM, over these control points, and then scaled to the grid points nearby through a linear interpolation. Deng et al. found out that the solver is second-order accurate, by testing it on a circular cylinder immersed in a lid-driven cavity.

4.3 Ghost-cell finite-difference approach

In this subsection the ghost cell method of Tseng et al. will be treated. Then some improvements to this method will be described. The ghost cell method for compressible flows developed by Ghias et al., won’t be treated, because it used the same principles as the method of Tseng et al.. And finally for clarification, the difference between the ghost cell method and ghost fluid method will be discussed.
4.3. GHOST-CELL FINITE-DIFFERENCE APPROACH

4.3.1 Basic formulation

Tseng et al. [58] extend the idea of Fadlun et al. [11] and Verzicco et al. [67] via a ghost cell approach (introduced in [39]). Their approach attempts to achieve a higher-order representation of the boundary using a ghost zone inside the body. Ghost cells are defined as cells in the solid that have at least one neighbour in the fluid. For each ghost cell, an interpolation scheme has to be made that implicitly incorporates the boundary condition on the IB. There are a number of options available for constructing the interpolation scheme. A simple choice will be a bilinear (trilinear in 3D) interpolation. However, at high Reynolds numbers when the resolution is marginal, linear reconstruction could lead to erroneous predictions [42]. A more sophisticated option would be an interpolation scheme that is linear in the tangential direction and quadratic in the normal direction [39].

Figure 4.6: Ghost cell, adapted from [58].

The simplest approach in 2D is to construct a triangle with the ghost node and the two nearest fluid nodes as the vertices. This choice minimizes the probability of numerical instability. In Fig. 4.6(b) G is the ghost node, X₁ and X₂ are the two nearest fluid nodes and O is the node at which the boundary condition is to be satisfied. A simple interpolation formula in 2D is

\[ \phi = a_0 + a_1 x + a_2 y, \]  

(4.4)

where \( \phi \) is a local flow variable. The ghost cell value is a weighted combination of the values at the nodes (X₁, X₂ and O), given by

\[ \phi_G = w_1 \phi_1 + w_2 \phi_2 + w_O \phi_O. \]  

(4.5)
The coefficients can be expressed in terms of the nodal values

\[ [w_1, w_2, w_3]^T = B^{-1}[x_G, y_G, 1]^T, \]  

(4.6)

where, for linear interpolation, \( B \) is a 3 x 3 matrix, whose elements can be computed from the coordinates of the three boundary points of the interpolation space. This linear relationship is used to extrapolate the ghost-point value \( G \), because \( G \) lies outside the interpolation space, i.e. \( G \) lies outside the region covered by the points \((X_1, X_2, O)\). The major drawback with this extrapolation is that large negative weighting coefficients are encountered when the boundary point is close to one of the fluid nodes used in the extrapolation. Although this is algebraically correct, this can lead to numerical instability, i.e. the absolute value at the ghost point may be greater than the nearby point values and the solution may not converge.

Two approaches are used to remedy the difficulty. The first is to use the image of the ghost node inside the flow domain to ensure positive weighting coefficients [39]. The point \( I \) is the image of the ghost node \( G \) through the boundary as shown in Fig. 4.7(a). The flow variable is evaluated at the image point using the interpolation scheme. The value at the ghost node is then \( \phi_G = 2\phi_O - \phi_I \), for the Dirichlet boundary condition.

![Schematic of a ghost cell using the image method.](image1)

![Schematic of adding an additional ghost cell \( G' \) if the boundary is close to the fluid points.](image2)

Figure 4.7: Special treatment to minimize numerical instability. \(--\) is the linear piecewise approximation to the boundary, \(\cdots\) is the boundary approximated by two piecewise segments, adapted from [58].

The other approach is to modify the piecewise linear boundary [17] (ghost fluid method). This holds only for the case when Dirichlet boundary conditions are imposed. When the boundary is close to a fluid node and far from the ghost node as in Fig. 4.7(b), the boundary point will be moved to the fluid node closest to the boundary. Using the fact that the boundary is approximated as piecewise linear, the accuracy is hardly affected when
the boundary segment is divided into two pieces, see Fig. 4.7(b). This ensures that large negative weights will not occur.

This method does not require any internal treatment of the body except the ghost cells since a fractional step method is used and the forcing is only on the boundary. It can handle Dirichlet, Neumann types and mixed forms of these types (Robin boundary condition). A disadvantage is that in the method of Tseng et al., the fictitious velocity field inside the immersed boundary is included in the continuity equation, resulting in a slower convergence rate and flux over the immersed boundary [40].

In the staggered grid arrangement all three velocity components and the pressure are computed on different grids. This arrangement increases the required storage. However, the increase is not significant since the boundary is lower dimensional than the domain [58].

### 4.3.2 Improvements

The ghost cell approach is also used for example by Mittal et al. [41]. They constructed the interpolation operators in a direction normal to the immersed boundary, which simplifies the implementation of the Neumann boundary conditions on the immersed boundary. They also use the concept of the image point. When using an image point, a boundary intercept point is needed. As shown in Fig. 4.8, there is not always a unique boundary intercept point or even a real boundary intercept point. Correct identification of this point is crucial, since incorrect boundary intercept points can lead to an excessively large interpolation stencil for the ghost-cell and can severely deteriorate the iterative convergence of the governing equations.

![Figure 4.8](image)

Figure 4.8: Two degenerate situations encountering in identification body-intercept point; ⋅ is Fluid-Cell, ♂ is Ghost-Cell and ◇ is Solid-Cell [41].

To avoid this problem, Mittal et al. have adopted an approach whereby they first determine the surface element vertex that is closest to the ghost-cell. Next, they identify the set of surface elements that share this vertex and search for a normal-intercept among
these elements. When there are multiple normal-intercepts found, the body-intercept point is chosen to be the normal-intercept point that has the shortest intercept. For cases where no normal-intercepts are found on the surface, the first thing to do is to repeat the search over a larger region of the surface surrounding the closest vertex. If the search is still unsuccessful, revert back to the first set of surrounding elements and search for the point in this set of elements that is closest to the ghost-cell. Keep in mind that this closest point could even be on the edge or vertex of an element.

Their method does fulfil the divergence-free condition, i.e. conservation of mass. Using a central-difference spatial scheme coupled with a collocated mesh guarantees good discrete kinetic energy conservation properties [13]. Felten et al. [13] showed that pressure errors do not have a visible impact on the results, provided that the simulations are run at a sufficient high mesh resolution and small time steps. And for the interpolation error, employing a first-order centred interpolation is necessary. Furthermore, the method includes moving boundaries and works in 3D.

Berthelsen et al. [2] present an immersed boundary method which is capable of solving the incompressible Navier-Stokes equations in the presence of highly irregular boundaries. The main idea is to use a local directional ghost cell which is obtained by one-dimensional extrapolation along the same direction as the discretization it will be used for. This construction allows for highly irregular boundaries (e.g. sharp corners and thin plates) to be treated accurately. Each irregular grid cell has its own set of local ghost cells, because of the topological differences. The ghost cells are updated such that the velocity field satisfies the immersed boundary conditions as well as the incompressibility constraint at the end of each time step. The time stepping is done explicitly using a second-order Runge-Kutta method. The spatial derivatives are approximated by finite difference methods on a staggered, Cartesian grid with local block structured grid refinements near the immersed boundary. They demonstrated that the spatial accuracy of their numerical method is second-order. Berthelsen et al. made some suggestions about extending this work to moving boundaries. Although their method can handle highly irregular boundaries very accurately, the stencil belonging to the interpolation for each direction consists of 7 points which is not preferable.

Based on the ghost cell method of Tseng et al. [58], Pan et al. [46] made a method for solving the incompressible Navier-Stokes equations by an implicit pressure correction upwind finite volume method on a Cartesian mesh. Multigrid methods have been developed to solve the discretized equations for both velocity and pressure correction. The remedies introduced by Tseng et al. for dealing with fluid nodes which are too close to the immersed boundary, require local modifications of the reconstruction stencil. Therefore, using the concept of image points, Pan et al. construct a simple and stable reconstruction scheme. This is done by using the location of the image point a fixed distance away from the body surface. There is no ambiguity in choosing the stencil points and the extrapolation to the ghost cell will not introduce numerical instability. Pan et al. show that their method is second-order accurate in space. Also this method is capable of handling moving bodies.
Pan et al. suggested that their method can be extended to three dimensions.

Using the ghost point treatment as starting point, Gao et al. [15] improved the method of Tseng et al. [58]. When the boundary is close to a fluid node, the inversions for different matrices may tend to be ill-conditioned and even become more serious for higher-order reconstruction polynomials. To overcome this difficulty, they use a second-order Taylor series expansion rather than polynomials to obtain the values at the ghost points so that the matrix inversions are avoided. An inverse distance weighting (IDW) interpolation method by Franke [14] is employed to interpolate the values due to its properties of preserving the local extrema and smooth reconstructions. The IDW interpolation also provides a smooth and flexible boundary treatment, leading to accelerating the convergence. For moving boundaries, they use a non-inertial reference frame fixed to the moving rigid body [31].

To improve the accuracy at the boundaries, Shinn et al. [52] implemented the immersed boundary method using the ghost cell approach, whereby the incompressible flows are solved on a staggered grid. Their primary concern is the satisfaction of local mass continuity for ghost pressure cells, rather than extrapolating the pressures from within the flow domain. The method of Shinn et al. preserves local continuity in each cell and also global continuity. As a result, no explicit mass sources or sinks are needed.

For each ghost cell, a mirror image point is determined by reflecting the ghost point across the boundary into the interior flow domain, like in Fig. 4.9.

![Figure 4.9: Illustration of a ghost point along the normal to a mirror point M](image)

To determine the value of the mirror point $\Phi_M$, the points $A, C, G$ and $G_1$ are used. Use of $\Phi_G$ to subsequently determine $\Phi_G$ from $\Phi_M$ has been observed by Shinn et al. to lead to numerical divergence if $G$ is close to the boundary. Therefore, they rewrite

$$\Phi_M = f_1 \Phi_A + f_2 \Phi_C + f_3 \Phi_G + f_4 \Phi_{G_1}, \tag{4.7}$$

where $f_1, f_2, f_3$ and $f_4$ are bilinear interpolation coefficients. Substituting the value of $\Phi_G$ from the mirror condition, $\frac{\Phi_G + \Phi_M}{2} = \Phi_B$, into Eq. (4.7) gives

$$\Phi_M = f_1 \Phi_A + f_2 \Phi_C + f_3 (2\Phi_B - \Phi_M) + f_4 \Phi_{G_1}$$

That is,
\[ \Phi_M (1 + f_3) = f_1 \Phi_A + f_2 \Phi_C + 2f_3 \Phi_B + f_4 \Phi_G. \]  

(4.8)

Then the ghost values can be determined on which the forcing will be applied.

### 4.3.3 Difference between ghost cell and ghost fluid method

We have to make a distinction between the ghost cell method and the ghost fluid method of Fedkiw et al. [12], because there can be some confusion between them. A ghost fluid method is commonly used in fluid-fluid structures, such as surface flows. The ghost fluid method implicitly captures the boundary conditions at the interface by the construction of a ghost fluid i.e. an artificial fluid. Ghost cells are defined at every point in the computational domain so that each point contains the mass, momentum and energy for the real fluid that exists at that point and a ghost mass, momentum and energy for the other fluid that does not really exists at that grid point. In the ghost cell method, not every point in the computational domain has a ghost point. Also, the way the value of the ghost points are defined is different. For this comparison, the method of Gibou et al. [17] will be used, because it is second-order accurate. This ghost fluid method [17] computes these values through one-dimensional extrapolation along the Cartesian grid lines whereas the method of Tseng et al. [58] constructs the interpolation along the boundary normal direction. Furthermore, the interpolation used in [17] is along the principal direction (instead of the normal to the immersed boundary in [58]) which would tend to complicate the imposition of Neumann boundary conditions.

### 4.4 General considerations

An advantage of the methods treated in this section is that they all can make a sharp representation of the immersed boundary, which is necessary for high Reynolds numbers. They do not introduce any extra stability constraints in the representation of solid bodies, due to absence of user-specified parameters in the forcing and the elimination of associated stability constraints. The methods decouple the equations for fluid nodes from solid grid points. A disadvantage of these methods is that they all strongly depend on the discretization method in contrast to the continuous forcing approach. However, this allows direct control over the numerical accuracy, stability and discrete conservation properties of the solver. Another drawback is that these methods are not straightforward for implementation due to first discretization and then introducing a forcing term. The methods which are categorized into the continuous forcing section do not suffer from this difficulty. Also inclusion of boundary motion can be more difficult.
Chapter 5

Cut-cell finite-volume approach

The basic formulation of the cut-cell method will first be described. Then some improvements for this approach will be treated. And finally, the problem of using a staggered grid instead of a collocated grid for the cut-cell method will be discussed.

5.1 Basic formulation

Ye et al. [69] proposed a different approach for simulating convection-dominated flows on a collocated (non-staggered) grid called a cut-cell method (in the past also named Cartesian grid method), which does not use the concept of momentum forcing. They used a central-difference interpolation scheme near the immersed boundary that gives second-order spatial accuracy. In this method cells in the Cartesian grid that are cut by the IB are identified, and the intersection of the boundary with the sides of these cut-cells is determined. Next, cells cut by the IB, whose cell center lies in the fluid, are reshaped by discarding the portion of these cells that lies in the solid. Pieces of cut-cells whose centres lie in the solid are usually absorbed by neighbouring cells to prevent stability problems. This results in the formation of control volumes, which are trapezoidal in shape, as shown in Fig. 5.1. Details of this reshaping procedure can be found in Udaykumar et al. [61,64].

The approach proposed by Ye et al. is to express a given flow variable \( \phi \) in terms of a two-dimensional polynomial interpolating function in an appropriate region and evaluate the fluxes \( f \) based on this interpolating function. For instance, in order to approximate the flux on the southwest face, \( f_{sw} \), \( \phi \) (in the shaded trapezoidal region shown in Fig. 5.2(b)) is expressed in terms of a function that is linear in \( x \) and quadratic in \( y \)

\[
\phi = c_1 xy^2 + c_2 y^2 + c_3 xy + c_4 y + c_5 x + c_6,
\]

where \( c_1 \) to \( c_6 \) are six unknown coefficients. Eq. (5.1) represents the most compact function that allows at least a second-order accurate evaluation of \( \phi \) at the \( sw \) location.
The presence of immersed boundaries alters the conditioning of the linear operators and this can slow down the iterative solution of the these equations. Therefore a preconditioned Conjugate Gradient method is used for accelerating the convergence. Since the inside of the immersed boundary is also gridded, this method also has the capability to solve a different set of equations inside the immersed boundary. For instance, equations of heat conduction could be solved inside the body.

While Ye et al. show that their interpolation scheme is itself second-order accurate, in
their test for the accuracy of the overall scheme they use simulations of Wannier flow in which there is no contribution from the advection (convection) terms \cite{33}. A disadvantage of cell-merging is that it generally entails a considerable increase in complexity as fluxes between diagonally adjacent cells must also be calculated and the computational molecule for merged boundary cells becomes different to that used for the standard cells. There are also significant problems associated with the formulation of a systematic merging algorithm in three dimensions \cite{33}, because of complex polyhedral cells.

The discretization fulfills the conservation laws due to the finite-volume approach. Mittal et al. \cite{41} stated that successful implementation of the cut-cell method (with cell merging) to three-dimension geometries has not yet been accomplished, due to the above described difficulties. Kirkpatrick et al. \cite{33} noted that the matrix condition number increases significantly when the size of the reshaped control volume is very small.

Udaykumar et al. \cite{63} extended the method of Ye et al. in order to allow for motion of the immersed boundary. The moving boundary is represented as a sharp interface using an Eulerian-Lagrangian approach and the interface tracking procedure from Udaykumar et al. \cite{64}. The curved immersed boundary is represented using marker particles which are connected by piecewise quadratic curves parametrized with respect to the arclength. They use an implicit treatment, i.e. the boundary and flow are advanced in time simultaneously in a fully coupled manner instead of a sequentially manner which is the case for an explicit treatment. The primary advantage of this approach instead of the explicit treatment, is that it removes any stability constraints associated with the boundary motion. A multigrid method is used for accelerating the convergence of the pressure Poisson equation. The finite-volume discretization in a given cell can be written in the general form

\[
\sum_{k=1}^{M} a_k \phi_k = b
\]  

(5.2)

where \(a_k\) (for \(k = 1, \ldots, M\)) denote the coefficients accompanying the nodal values \(\phi_k\) within a stencil consisting of \(M\) neighbours and \(b\) is the source term that contains the explicit terms as well as the terms involving boundary conditions. For cells away from the interface, \(M = 5\) corresponds with the central-difference spatial discretization.

Udaykumar et al. \cite{62} demonstrated the versatility of the numerical method \cite{63} by applying it to some challenging physical problems in solidification and fluid-structure interactions.

### 5.2 Improvements

Kirkpatrick et al. \cite{33} developed a method for representing curved boundaries for the solution of the Navier-Stokes equations on a non-uniform, staggered, three-dimensional Cartesian grid. Solid boundaries are defined as quadratic surfaces in a Cartesian coordinate system. Kirkpatrick et al. circumvent the problems related to cell-merging by introducing a
novel cell-linking method. This method overcomes problems associated with the creation of small cells while avoiding the complexities involved with other merging approaches. Rather than merging two cells to form a single cell, the two cells are linked as a "master/slave" pair in which the two nodes are coincident while each cell remains a distinct entity. A master cell and slave cell is shown in Fig. 5.3. As a result, the fluxes, wall shear stress, volumetric and surface information are calculated in exactly the same way for the master and slave cells as they are for the standard boundary cells.

There are some problems when using a staggered grid in combination with a cut-cell approach, as pointed out by Kirkpatrick et al. in §5.4. Using cell-linking, is one of the solutions for some of these problems.

Figure 5.3: A master and slave cell are shown for the u component of velocity. The slave cell velocity has only one pressure node associated with it. It is moved to the same position as the master cell node [33].

However, the application of Kirkpatrick et al. method to moving objects is not so easy because of the presence of extremely small-size cells [56]. Also, the way the problem of linking a master and a slave velocity node has been handled is not elegant. The slave node is placed a small distance from the master node. Then the diffusion flux between the two nodes automatically becomes extremely high and forces the two velocities to take the same value. This situation is not ideal for the case of high Reynolds number flows.

Dröge et al. [9] introduced a different kind of cut-cell discretization. The cut-cells are not merged or linked as in the previous cases, there are just left alone. Their cut-cell discretization preserves the symmetry of convection and diffusion. Convection is discretized by a skew-symmetric operator and diffusion is approximated by a symmetric, positive-definite coefficient matrix. This formulation for discretization conserves the kinetic energy, if the dissipation is turned off, and is stable on any grid. This scheme is second-order accurate, when the dissipation is turned off.

The cut-cell approach of Chung [5] simulates two-dimensional unsteady viscous incompressible flows with (moving) rigid bodies of arbitrary shape on a collocated Cartesian grid.
The pressure-free projection method is used as the incompressible flow solver. This solves the coupling problem between the divergence-free constraint and the pressure gradient in the momentum equation. In the work of Ye et al. [69], the value of variables or fluxes at the center of the merged cell face or the merged cutting surface of the merged cell is directly interpolated from the neighbouring positive centers or centers of the merged boundary segment. However, in Chung’s method, the merged cell face is regarded as a combination of a full cell face and a cut-cell face. Likewise, the merged cutting surface is a combination of two boundary segments. This approach has particular benefit in three-dimensional problems, due to easier and more systematic making of both the location of the cut face center and the construction of the interpolation stencil. Further, the merging of boundary segments in Ye et al. will reduce the shape resolution of the immersed boundary, more than the method of Chung. See for example Fig. 5.4 for such a situation.

![Figure 5.4: Difference between Chung and Ye.](image)

The observed order of accuracy in the spatial discretization is super-linear, though not quadratic.

### 5.3 LS-STAG

Cheny et al. [3] developed a new immersed boundary method, which is based on the MAC method for staggered Cartesian grids and where the irregular boundary is sharply represented by its level-set function (signed distance function), hence it is called LS-STAG method. The flow variables are computed in the cut-cells and not interpolated like in e.g. [11]. Furthermore, the fluxes in Cartesian and cut-cells are discretized in a consistent and unified fashion, so \textit{ad hoc} treatments for the cut-cells are no longer needed. The LS-STAG is based on the symmetry preserving finite-volume method by Verstappen et al. [66], which has the ability to preserve on non-uniform staggered Cartesian grids the conservation of total mass, momentum and kinetic energy of the original MAC method [22]. The discretization in the LS-STAG method preserves the five-point Cartesian structure of
the stencil, resulting in a highly computationally efficient method. The LS-STAG method can also handle moving immersed boundaries, but the first attempt of Cheny et al. is a simple procedure, which does not guarantee the discrete conservation of momentum or kinetic energy. Therefore, the developers will in the future improve their procedure by using in the freshly cleared cells (fluid cells which were inside the solid at the previous time step) a fully implicit semi-Lagrangian time-stepping. By performing an accuracy test, they discovered that their method is super-linear accurate in space. Cheny et al. stated that by using a level-set function, the immersed boundary can be sharply represented. However, Berthelsen et al. are of the opinion that the level-set formulation fails to describe sharp corners and infinitely thin plates.

5.4 Staggered grid compared to collocated grid

With a staggered grid, the pressure cell and the cells associated with each of the three velocity components are at a different location and will generally have a different shape when they are cut by an embedded boundary. A cut-cell scheme for a staggered grid must deal with this extra complexity in a consistent manner. Collocated grids do not suffer from this problem.

The pressure correction procedure by which mass conservation is enforced is also different for staggered and collocated arrangements. This leads to a number of complications in the staggered case which do not occur when a collocated grid is used. Firstly, on a staggered grid the placement of the velocity nodes is dictated by its role in the pressure correction equation. It must be placed at the centroid of the cut face of the pressure cell to maintain the advantages of using a staggered grid. Secondly, for calculation of the advective fluxes, different interpolations are required for the velocity components parallel and perpendicular to the flux direction.

5.5 General considerations

An advantage of a cut-cell method is that it is possible to accurately impose the boundary conditions on the body. Another benefit is that the cut-cell method is based on finite volume, so strict conservation of mass and momentum is guaranteed even in the vicinity of the immersed boundary. Furthermore, the application of adaptively refined grids does not complicate the implementation of this technique.

One of the drawbacks of this approach is the following. Implementing the boundary conditions in irregular cells requires a large number of "special treatments", which could result in complex coding logistics. When using cut-cells, these cells should not become too small. Otherwise this could not only lead to stability problems, but also lead to slow convergence of the Poisson solver. Finally, an iterative solution procedure may be required due to the irregular stencil near the immersed boundary.
Chapter 6

Evaluation of immersed boundary methods

The purpose of this section is to recommend a few immersed boundary methods, that fulfil (most of) the requirements stated in the section introduction (and more). To compare all of the methods treated so far, we put them in a table (see Table 6.1 and Table 6.2), and judge them by their strong and weak points. From this we get some insight in all treated immersed boundary methods.

6.1 Mass conservation

Fulfilling the conversation laws (mass, momentum and energy) is one of the most important issues. First of all, we consider the conservation of mass. This means that (exactly) satisfying the divergence free condition, i.e. conservation of mass for incompressible flow, is required. Otherwise, some non-physical situations can manifest, such as pressure build-up. However, only considering the divergence free condition does not guarantee that the pressure will not oscillate near the immersed boundary. Some authors, like [19, 41, 58], try to solve this problem by introducing a Neumann boundary condition for the pressure. This practice, however, leads to mass fluxes across the solid boundaries and mass errors in the ghost cells and eventually to oscillatory flow field or even numerical divergence. Kang et al. [27, 28] introduce a different kind of interpolation method. They included the effect of the pressure gradient in the linear interpolation. However, this form of interpolation is more complicated, because the interpolation of the pressure is also required. This construction shows similarities with introducing a Neumann boundary condition to deal with pressure oscillations. Kang et al. [27] use the above interpolation method and tested their method using a laminar flow around a circular cylinder to verify the accuracy of the wall pressure. By zooming in the results of the time-averaged wall pressure coefficients, Kang et al. showed that the wiggles in the wall pressure with the standard reconstruction method are reduced.
## CHAPTER 6. EVALUATION OF IMMERSED BOUNDARY METHODS

### Ease of Implementation

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Peskin</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Goldstein</td>
<td>+</td>
<td>-</td>
<td>±</td>
<td>+</td>
<td>-</td>
<td>expl</td>
<td>impl</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Saiki</td>
<td>+</td>
<td>-</td>
<td>±</td>
<td>+</td>
<td>-</td>
<td>expl</td>
<td>impl</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lai</td>
<td>±</td>
<td>-</td>
<td>±</td>
<td>-</td>
<td>-</td>
<td>expl</td>
<td>impl</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Khadra</td>
<td>-</td>
<td>-</td>
<td>±</td>
<td>+</td>
<td>-</td>
<td>expl</td>
<td>impl</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Su</td>
<td>±</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>expl</td>
<td>impl</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DLM</td>
<td>±</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>expl</td>
<td>impl</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IIM</td>
<td>Le</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>expl</td>
<td>impl</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Linear system</td>
<td>Taira</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>expl</td>
<td>impl</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mohd-Yusof</td>
<td>-</td>
<td>-</td>
<td>±</td>
<td>±</td>
<td>+</td>
<td>expl</td>
<td>impl</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fadlun</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>expl</td>
<td>impl</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Balaras</td>
<td>-</td>
<td>-</td>
<td>±</td>
<td>+</td>
<td>-</td>
<td>expl</td>
<td>expl</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gilmanov</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>expl</td>
<td>impl</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zhang</td>
<td>±</td>
<td>-</td>
<td>±</td>
<td>+</td>
<td>±</td>
<td>expl</td>
<td>impl</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Choi</td>
<td>-</td>
<td>±</td>
<td>+</td>
<td>+</td>
<td>±</td>
<td>expl</td>
<td>impl</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ikeno</td>
<td>-</td>
<td>±</td>
<td>+</td>
<td>+</td>
<td>±</td>
<td>expl</td>
<td>impl</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Muldoon</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>expl</td>
<td>impl</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sheu</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>expl</td>
<td>expl</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kang</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>±</td>
<td>+</td>
<td>expl</td>
<td>impl</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kang</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>expl impl</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mark</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>expl</td>
<td>impl</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Liao</td>
<td>-</td>
<td>±</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>expl</td>
<td>impl</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6.1: Overview characteristics IB methods.
### 6.1. MASS CONSERVATION

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ease of implementation</td>
<td>(local) mass conservation</td>
<td>works with high Reynolds numbers</td>
<td>Turbulence/LES</td>
<td>staggered grid</td>
<td>3D</td>
<td>moving boundaries</td>
<td>no severe time step restriction</td>
<td>second-order in space</td>
<td>discretization convection</td>
<td>discretization diffusion</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mass source/sink</td>
<td>expl</td>
<td>impl</td>
<td>expl</td>
<td>expl</td>
<td>expl</td>
<td>expl</td>
<td>expl</td>
<td>expl</td>
<td>expl</td>
<td>expl</td>
<td>expl</td>
<td>expl</td>
<td>expl</td>
<td>expl</td>
<td>expl</td>
<td>expl</td>
</tr>
</tbody>
</table>

**Table 6.2:** Overview characteristics IB methods.
by their method. But their results suggested that there is still some mass flux across the immersed boundary. Choi et al. use a power-law based interpolation, but this interpolation method does not guarantee that there exists no mass flux over the immersed boundary. Ikono et al. tried to show that the divergence free condition will be satisfied, but it is still unclear if they really can manage such a condition by using some algebra. Because, when using points inside a solid body, there is a great chance that there exists some mass flux over the immersed boundary. The only remedy, see the papers \cite{40,52}, is to exclude the "ghost" points from the continuity equation, which is not done by Ikono et al..

From this discussion it is clear that not all immersed boundary methods satisfy the mass conservation. So the immersed boundary methods, which do not satisfy the mass conservation drop out. From Tables 6.1 and 6.2 we can conclude that this requirement leads to the fact that all of the continuous forcing methods will drop out. Also the immersed interface method, distributed Lagrange multiplier method and the methods in which the force is computed implicitly by solving a linear system will drop out. This leave us with only eight methods that satisfy the (local) mass conservation condition. The numerical methods concerned are the following: Mark et al. \cite{40}, Huang et al. \cite{23}, Silva et al. \cite{53}, Shinn et al. \cite{52}, Udaykumar et al. \cite{62,63}, Kirkpatrick et al. \cite{33}, Chung et al. \cite{5} and Cheny et al. \cite{3}. The first four of the eight methods can be categorized as direct-forcing methods and the other four fall into the category of cut-cell methods.

6.2 High Reynolds numbers/Turbulent flows

For most applications we have in mind it is important that an immersed boundary method can deal with high Reynolds number wall bounded flows. This computation is challenging as it requires the consideration of thin turbulent boundary layers, i.e. near wall regions with large gradients of the flow field variables. For such flows, the representation of the wall boundary has a large impact on the accuracy of the computation. It is also critical for the robustness and convergence of the flow solver \cite{26}. Considering the requirement that an immersed boundary method can work with high Reynolds number flows, the methods of Huang et al. and Silva et al. are cancelled out.

6.3 Large eddy simulations

Some papers treat explicitly a combination between large eddy simulation and immersed boundary methods, see for example Iaccarino et al. \cite{24} and Tyagi et al. \cite{60}. The starting point is building a framework for large eddy simulation, and for the boundary an immersed boundary method is used. Cristallo et al. \cite{6} prefer to work with wall functions instead of immersed boundary methods. The reason is that as the Reynolds number increases it becomes difficult to fulfil the near-wall resolution requirements when using an immersed boundary method. A suggestion for a wall model is the one made by Tessicini et al. \cite{57}. 
They considered a two-layer wall modelling approach instead of the linear interpolation \cite{11}, in order to extend the applicability of the immersed boundary method to higher Reynolds number flows. Kalitzin et al. \cite{26} and Peller et al. \cite{47} among others, are trying to improve the accuracy in the vicinity of the wall for the benefit of large eddy simulations. The interpolation proposed by Kalitzin et al. \cite{26} is carried out in two steps. First, for each neighbouring cell a (linear or quadratic) interpolation is carried out normal to the wall to an intermediate location that has the same distance to the wall as the interface cell center (cells that have their cell center outside or on the surface). This approach is based on the interpolation method of Fadlun et al. \cite{11}, see \S4.2.1. The second interpolation is at a surface that is parallel to the wall. For this, they used inverse distance weighted method \cite{14}, like Gao et al. \cite{15}, see \S4.3.2. Peller et al. \cite{47} used a third-order least squares interpolation for the interface cells. For this interpolation a polynomial is chosen so that the sum of the squares of the distance from the polynomial to the values at the fluid points is minimal. Their method prevents instabilities which are present in high-order Lagrange interpolation schemes which they also have been investigating. Although papers have been published about combining large eddy simulations with a direct-forcing method, the combination of cut-cell methods with large eddy simulations has still to be found.

6.4 Compressible flows

De Palma et al. \cite{45}, De Tullio et al. \cite{59}, Yang et al. \cite{68} and Ghias et al. \cite{16} developed methods for dealing with compressible flows over (moving) bodies. Due to a different set of Navier-Stokes equations, compressible flows require some additional considerations compared to incompressible. This will manifest in different spatial discretization schemes. That is mainly the biggest difference between compressible and incompressible flows. The concept of how the treated immersed boundary methods and cut-cell methods are constructed for incompressible flows remains the same for compressible flows.

6.5 Staggered grid

Another requirement for the immersed boundary method we have in mind is that it must be implemented on a staggered grid. Although it may be possible to transform an immersed method based on a collocated grid to a staggered grid, the methods of Udaykumar et al. \cite{63,63} and Chung et al. \cite{5} drops out of the shortlist from \S6.1.

6.6 General considerations

By collecting the pros and the cons of the remaining methods, the best methods will roll out. In this case, we consider the following best methods for some further investigation: Mark et al. \cite{40}, Shinn et al. \cite{52}, Kirkpatrick et al. \cite{33} and Cheny et al. \cite{3}.
One of the reasons why the method of Kirkpatrick [33] is not extended to moving boundaries is that it is in general difficult to apply moving boundaries to cut-cell methods, due to the presence of small cells. Also there is some extremely high diffusion flux between a master and a slave velocity node, which is not ideal for the case of high Reynolds number flows. Cheny et al. [3] stated that despite of all the efforts for improving the accuracy and consistency of the direct-forcing methods, these methods are still not able to satisfy strict conservation of quantities such as mass, momentum or kinetic energy near the irregular boundary. However, Cheny et al. do not mention anything about the method of Mark et al. [40] or Shinn et al. [52]. Also Cheny et al. has taken a lot of effort in the discretization of the viscous terms of the momentum equation. As mentioned by them, the discretization of the viscous fluxes is much more intricate than for the convective fluxes. The method of Cheny et al. together with Taira et al. [55] and Dröge et al. [9], are the only methods which use symmetry preservation. The method of Cheny et al. makes use of a combination between cut-cell and level-set function. The problem of using a level-set function is that the level-set formulation fails to describe sharp corners and infinitely thin plates [2]. Also Cheny et al. did not say anything about their method if it works with high Reynolds numbers and also if it can handle turbulent flows. The proposed methods by Mark et al. and Shinn et al. do fulfil the mass conservation, but both methods were not tested for high Reynolds number flows.

### 6.7 Recommended immersed boundary methods

Concluding from the previous section, the recommended immersed boundary methods to investigate further are the cut-cell method of Cheny et al. [3] (LS-STAG method) and the direct forcing methods of Mark et al. [40] and Shinn et al. [52]. The original plan was to implement a cut-cell method and a direct forcing method and then compare these. However due to lack of time, we had to make a decision between those two types of methods. The most important argument to choose the LS-STAG method of Cheny et al. for further investigating, was that the existing ComFLOW code has already a cut-cell method implemented, so only the discretization has to be modified, which saves a lot of time. The direct forcing method has a different structure, which means that we have to do more work for implementing such a method. Another argument is that in ComFLOW the diffusion is discretized in a very simple way (as if no cut-cells are present). So by implementing the discretization used in the LS-STAG method, the ComFLOW method can be improved. Our experience with the LS-STAG method will be described in Part II of this report.
Part II

Discretization of LS-STAG
Chapter 7

Basic discretization

7.1 LS-STAG mesh

To clarify the notation that will be used in the subsequently chapters, an example of a cut-cell is plotted, see Fig. 7.1.

\[
\theta_{u,i,j} \quad \theta_{v,i,j} \quad \Delta x_i \quad \Delta y_j
\]

\[
u_{i,j}^b \quad u(x_i, y_{i,j}^b)
\]

\[
u_{i,j}^a \quad v_{i,j}^a
\]

\[
u_{i,j+1} \quad v_{i,j+1}
\]

\[
u_{i,j-1} \quad v_{i,j-1}
\]

Figure 7.1: A south-east triangular cut-cell.

The apertures or cell-face fraction ratios that are open for fluid are notated as \(\theta_{u,i,j}\) and \(\theta_{v,i,j}\), which represents the fluid portion of the east and north faces, respectively. The volume of a cut-cell is notated as \(V_{i,j}\). When considering a south-east triangular cut-cell [7.1], the corresponding volume \(V_{i,j} = \frac{1}{2} \theta_{u,i,j} \theta_{v,i,j-1} \Delta x_i \Delta y_j\). This cut-cell is one of the 12, which are used in the LS-STAG method. This methods assumes that every cut-cell must be of the following form
CHAPTER 7. BASIC DISCRETIZATION

(a) Triangular shaped cut-cells.

(b) Trapezoidal shaped cut-cells.

(c) Pentagonal shaped cut-cells.

Figure 7.2: Overview of all possible cut-cells.

This means that curved boundaries are approximated by linear line segments.

7.2 Navier-Stokes equations

The finite-volume discretization of the momentum equation in the $x$ and $y$ directions, respectively

$$
\frac{d}{dt} \int_{\Omega} u dV + \int_{\Gamma} (\mathbf{v} \cdot \mathbf{n}) u dS + \int_{\Gamma} \mathbf{p}_x \cdot \mathbf{n} dS - \int_{\Gamma} \nu \nabla u \cdot \mathbf{n} dS = 0,
$$

(7.1)

$$
\frac{d}{dt} \int_{\Omega} v dV + \int_{\Gamma} (\mathbf{v} \cdot \mathbf{n}) v dS + \int_{\Gamma} \mathbf{p}_y \cdot \mathbf{n} dS - \int_{\Gamma} \nu \nabla v \cdot \mathbf{n} dS = 0,
$$

(7.2)

where $p$ is the pressure and $\nu$ is the kinematic viscosity.

Considering the discretization of the momentum equation, the corresponding semi-discrete matrix representation is given by

$$
\frac{d}{dt} \mathcal{M} \mathbf{U} + \mathcal{C}[\mathbf{U}] \mathbf{U} + \mathcal{G}P - \nu \mathcal{K} \mathbf{U} + S^{ib,c} - \nu S^{ib,v} = 0,
$$

(7.3)
where the diagonal mass matrix $\mathcal{M}$ is built from the volume of the fluid cells, the matrix $C[U]$ represents the discretization of the convective fluxes, $G$ is the discrete pressure gradient, $K$ represents the diffusive term, $S^{ib,c}$ and $\nu S^{ib,v}$ are source terms arising from the boundary conditions of the convective and viscous terms, respectively.

The finite volume discretization of the continuity equation leads to

$$
\int_{\Gamma} \mathbf{v} \cdot \mathbf{n} dS = 0 \tag{7.4}
$$

$$
\hat{m}_{i,j} \equiv -\bar{u}_{i-1,j} + \bar{u}_{i,j} - \bar{v}_{i,j-1} + \bar{v}_{i,j} + \mathcal{U}^{ib}_{i,j} = 0, \tag{7.5}
$$

where $\mathbf{v} = (u, v)$ is the velocity and

$$
\bar{u}_{i,j} \equiv \int_{\Gamma^e_{i,j}} \mathbf{v} \cdot \mathbf{e}_x dS \approx \theta^u_{i,j} \Delta y_j u_{i,j} \tag{7.6}
$$

$$
\bar{v}_{i,j} \equiv \int_{\Gamma^e_{i,j}} \mathbf{v} \cdot \mathbf{e}_y dS \approx \theta^v_{i,j} \Delta x_i v_{i,j} \tag{7.7}
$$

$$
\mathcal{U}^{ib}_{i,j} \equiv \int_{\Gamma^b_{i,j}} \mathbf{v}^{ib} \cdot \mathbf{n}^{ib}_{i,j} dS \tag{7.8}
$$

$$
\approx u^{ib}_{i,j} [n_x \Delta S]^{ib}_{i,j} + v^{ib}_{i,j} [n_y \Delta S]^{ib}_{i,j} \tag{7.9}
$$

$$
= u^{ib}_{i,j} (\theta^u_{i-1,j} - \theta^u_{i,j}) \Delta y_j + v^{ib}_{i,j} (\theta^v_{i,j-1} - \theta^v_{i,j}) \Delta x_i \tag{7.10}
$$

where $\mathcal{U}^{ib}_{i,j}$ denotes the mass flux through the solid part of the cell boundary, and $v^{ib}_{i,j}$ is the velocity prescribed at the immersed boundary.

Thus the discrete continuity equation is given by

$$
\hat{m}_{i,j} \equiv \Delta y_j (\theta^u_{i,j} u_{i,j} - \theta^u_{i-1,j} u_{i-1,j}) + \Delta x_i (\theta^v_{i,j} v_{i,j} - \theta^v_{i,j-1} v_{i,j-1}) + \mathcal{U}^{ib}_{i,j} = 0. \tag{7.11}
$$

### 7.3 Pressure gradient

The pressure gradient must be dual to the divergence operator

$$
\mathcal{G} = -\nabla^T. \tag{7.12}
$$
In the situation shown in Fig. 7.3 part of the $x$-contribution to the discrete divergence is given by

$$D^x = \begin{pmatrix} \theta^u_{i-1,j} \Delta y_j & 0 & 0 \\ 0 & \theta^u_{i,j} \Delta y_j & 0 \\ 0 & 0 & \theta^u_{i+1,j} \Delta y_j \end{pmatrix}$$

This implies that using (7.12), the $x$-component of the discrete gradient becomes

$$G^x = \begin{pmatrix} \theta^u_{i-1,j} \Delta y_j & 0 \\ 0 & \theta^u_{i,j} \Delta y_j & 0 \\ 0 & 0 & \theta^u_{i+1,j} \Delta y_j \end{pmatrix}$$

Note how the apertures are positioned. Therefore, the discrete pressure gradient in control volumes $\Omega_{i,j}^u$ and $\Omega_{i,j}^v$ is described as

$$\int_{\Gamma_{i,j}^u} p e_x \cdot n dS \approx [G^u P]_{i,j} = \theta^u_{i,j} \Delta y_j (p_{i+1,j} - p_{i,j})$$

$$\int_{\Gamma_{i,j}^v} p e_y \cdot n dS \approx [G^v P]_{i,j} = \theta^v_{i,j} \Delta x_i (p_{i,j+1} - p_{i,j})$$

For clarification of the discretization of the pressure gradient, see Fig. 7.3.

### 7.4 Navier-Stokes discretized

Convection and diffusion are both discretized explicitly as
7.4. NAVIER-STOKES DISCRETIZED

\[ \mathcal{D}^{n+1}U^{n+1} = -\mathcal{D}_T^{n+1}U^{ib,n+1} \quad (7.15) \]
\[ \mathcal{M} \frac{U^{n+1} - U^n}{\Delta t} = -\mathcal{C}[U]U^n - \mathcal{G}P^{n+1} + \nu KU^n \quad (7.16) \]
\[ \mathcal{D} = -\mathcal{G}^{T} \quad (7.17) \]
\[ U^{n+1} = U^n + \Delta t M^{-1} \left[ -\mathcal{C}[U]U^n - \mathcal{G}P^{n+1} + \nu KU^n \right] \quad (7.18) \]

where \( \mathcal{M}_x(i, j) = \frac{1}{2} V_{i,j} + \frac{1}{2} V_{i+1,j} \) for the \( x \)-direction and \( \mathcal{M}_y(i, j) = \frac{1}{2} V_{i,j} + \frac{1}{2} V_{i,j+1} \) for the \( y \)-direction.

Introducing an auxiliary variable

\[ \tilde{U}^n = U^n + \Delta t M^{-1} \left[ -\mathcal{C}[U]U^n + \nu KU^n \right] \quad (7.19) \]

and putting this equation into above formula results in

\[ U^{n+1} = \tilde{U}^n - \Delta t M^{-1} \mathcal{G}P^{n+1}. \quad (7.20) \]

Taking the divergence of both sides of the equation gives

\[ \mathcal{D}^{n+1}U^{n+1} = \mathcal{D}^{n+1} \left[ \tilde{U}^n - \Delta t M^{-1} \mathcal{G}P^{n+1} \right] = -\mathcal{D}_T^{n+1}U^{ib,n+1}. \quad (7.21) \]

This results in the Poisson equation for the pressure

\[ \mathcal{D}^{n+1} \mathcal{M}^{-1} \mathcal{G}P^{n+1} = \frac{1}{\Delta t} \left[ \mathcal{D}^{n+1} \tilde{U}^n + \mathcal{D}_T^{n+1}U^{ib,n+1} \right]. \quad (7.22) \]

For the immersed boundary, we use the no-slip boundary condition, i.e. the velocity at the immersed boundary is zero, i.e. \( \mathcal{D}_T^{n+1}U^{ib,n+1} = 0. \) We end up with

\[ (\mathcal{D}^{n+1} \mathcal{M}^{-1} \mathcal{G}P^{n+1}) = \frac{\mathcal{D}^{n+1} \tilde{U}^n}{\Delta t}. \quad (7.23) \]

This Poisson equation will be solved as described in the following.

### 7.4.1 Normal stress discretization

For the discretization of the normal stress, Cheny et al. use the fact that the discrete normal stress should be consistent with the discrete pressure, as both are contributing to the diagonal part of the Cauchy stress tensor. Therefore, the normal stress flux shall be discretized in a similar fashion as the pressure gradient \((7.13), (7.14)\):
\[
\int_{\Gamma_{i,j}} \frac{\partial u}{\partial x} \mathbf{e}_x \cdot \mathbf{n} \, dS \cong \theta_{i,j}^u \Delta y_j \left( \frac{\partial u}{\partial x} \bigg|_{i+1,j} - \frac{\partial u}{\partial x} \bigg|_{i,j} \right),
\]
(7.24)

\[
\int_{\Gamma_{i,j}} \frac{\partial v}{\partial y} \mathbf{e}_y \cdot \mathbf{n} \, dS \cong \theta_{i,j}^v \Delta x_i \left( \frac{\partial v}{\partial y} \bigg|_{i+1,j} - \frac{\partial v}{\partial y} \bigg|_{i,j} \right),
\]
(7.25)

Using Gauss theorem (Divergence theorem)
\[
\int_{\Omega_{i,j}} \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) \, dV = \int_{\Gamma_{i,j}} \mathbf{v} \cdot \mathbf{n} \, dS,
\]
(7.26)

which is valid at the discrete level in a cut-cell, since it is trivially verified by the MAC method in a Cartesian cell.

\[
\int_{\Omega_{i,j}} \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) \, dV \cong \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) \bigg|_{i,j} V_{i,j}
\]
(7.27)

\[
\int_{\Gamma_{i,j}} \mathbf{v} \cdot \mathbf{n} \, dS = \Delta y_j \left( \theta_{i,j}^u u_{i,j} - \theta_{i-1,j}^u u_{i-1,j} \right) + \Delta x_i \left( \theta_{i,j}^v v_{i,j} - \theta_{i,j-1}^v v_{i,j-1} \right) + U_{i,j}^{ib} \\
= \Delta y_j \left( \theta_{i,j}^u u_{i,j} - \theta_{i-1,j}^u u_{i-1,j} \right) + \Delta x_i \left( \theta_{i,j}^v v_{i,j} - \theta_{i,j-1}^v v_{i,j-1} \right) \\
+ u_{i,j}^{ib} \left[ \theta_{i-1,j}^u - \theta_{i,j}^u \right] \Delta y_j + v_{i,j}^{ib} \left[ \theta_{i,j-1}^v - \theta_{i,j}^v \right] \Delta x_i
\]
(7.28)

Combining the equations leads to

\[
\frac{\partial u}{\partial x} \bigg|_{i,j} \cong \frac{\theta_{i,j}^u u_{i,j} - \theta_{i-1,j}^u u_{i-1,j} + u_{i,j}^{ib} \left( \theta_{i-1,j}^u - \theta_{i,j}^u \right)}{V_{i,j}/\Delta y_j},
\]
(7.29)

\[
\frac{\partial v}{\partial y} \bigg|_{i,j} \cong \frac{\theta_{i,j}^v v_{i,j} - \theta_{i,j-1}^v v_{i,j-1} + v_{i,j}^{ib} \left( \theta_{i,j-1}^v - \theta_{i,j}^v \right)}{V_{i,j}/\Delta x_i},
\]
(7.30)

which are the discretization of the normal stress fluxes.

### 7.4.2 Poisson equation

In the Poisson equation, the term \( \mathcal{D}^{\alpha+1} \mathcal{M}^{-1} \mathcal{G} P^{\alpha+1} \) is required. Since we have not derived this term yet, it will be derived next.

Substituting the pressure gradient \( \mathcal{D} \) for the control volume \( \Omega_{i,j}^u \)

\[
[\mathcal{M}^{-1} \mathcal{G}^z P]_{i,j} = \frac{\Delta y_j (p_{i+1,j} - p_{i,j})}{\frac{1}{2} V_{i,j} + \frac{1}{2} V_{i+1,j}}
\]
(7.31)
into the first term of the divergence operator (7.11)

\[
\frac{\partial u}{\partial x}_{i,j} \approx \frac{\theta_{i,j}^u u_{i,j} - \theta_{i-1,j}^u u_{i-1,j}}{V_{i,j}/\Delta y_j}
\]  

(7.32)

leads to

\[
D^{n+1}M^{-1}G^{n+1}|_{i,j} = \frac{\theta_{i,j}^u [\theta_{i,j}^u \Delta y_j (p_{i+1,j} - p_{i,j})]}{(V_{i,j}/\Delta y_j) (\frac{1}{2}V_{i,j} + \frac{1}{2}V_{i+1,j})} - \frac{\theta_{i-1,j}^u [\theta_{i-1,j}^u \Delta y_j (p_{i,j} - p_{i-1,j})]}{(V_{i,j}/\Delta y_j) (\frac{1}{2}V_{i-1,j} + \frac{1}{2}V_{i,j})}.
\]  

(7.33)

So for the x-direction, the \(D^{n+1}M^{-1}G^{n+1}\) is given by

\[
\frac{p_{i-1,j} [\theta_{i-1,j}^u (\Delta y_j)^2]}{V_{i,j} (\frac{1}{2}V_{i-1,j} + \frac{1}{2}V_{i,j})} - \frac{p_{i,j} [\theta_{i,j}^u (\Delta y_j)^2]}{V_{i,j} (\frac{1}{2}V_{i-1,j} + \frac{1}{2}V_{i,j})} + \frac{p_{i,j} [\theta_{i-1,j}^u (\Delta y_j)^2]}{V_{i,j} (\frac{1}{2}V_{i,j} + \frac{1}{2}V_{i+1,j})} - \frac{p_{i,j} [\theta_{i,j}^u (\Delta y_j)^2]}{V_{i,j} (\frac{1}{2}V_{i,j} + \frac{1}{2}V_{i+1,j})}.
\]  

(7.34)

Analogous for \(D^{n+1}M^{-1}G^{n+1}|_{i,j}^y\), which becomes

\[
\frac{p_{i,j-1} [\theta_{i,j-1}^u (\Delta x_i)^2]}{V_{i,j} (\frac{1}{2}V_{i,j-1} + \frac{1}{2}V_{i,j})} - \frac{p_{i,j} [\theta_{i,j}^u (\Delta x_i)^2]}{V_{i,j} (\frac{1}{2}V_{i,j-1} + \frac{1}{2}V_{i,j})} - \frac{p_{i,j} [\theta_{i,j}^u (\Delta x_i)^2]}{V_{i,j} (\frac{1}{2}V_{i,j} + \frac{1}{2}V_{i,j+1})} + \frac{p_{i,j} [\theta_{i,j}^u (\Delta x_i)^2]}{V_{i,j} (\frac{1}{2}V_{i,j} + \frac{1}{2}V_{i,j+1})}.
\]  

(7.35)

Together this results in the following Poisson equation

\[
\frac{p_{i-1,j} [\theta_{i-1,j}^u (\Delta y_j)^2]}{V_{i,j} (\frac{1}{2}V_{i-1,j} + \frac{1}{2}V_{i,j})} - \frac{p_{i,j} [\theta_{i,j}^u (\Delta y_j)^2]}{V_{i,j} (\frac{1}{2}V_{i-1,j} + \frac{1}{2}V_{i,j})} + \frac{p_{i+1,j} [\theta_{i+1,j}^u (\Delta y_j)^2]}{V_{i,j} (\frac{1}{2}V_{i,j} + \frac{1}{2}V_{i+1,j})} - \frac{p_{i,j} [\theta_{i,j}^u (\Delta y_j)^2]}{V_{i,j} (\frac{1}{2}V_{i,j} + \frac{1}{2}V_{i+1,j})}.
\]  

(7.36)
Chapter 8

Discretization of the convective fluxes

For the convective fluxes a 5-point scheme is assumed:

\[
\int_{\Gamma_{i,j}} (\mathbf{v} \cdot \mathbf{n}) udS \approx C[U]^W_{i,j}u_{i-1,j} + C[U]^E_{i,j}u_{i+1,j} + C[U]^P_{i,j}u_{i,j} + C[U]^S_{i,j}u_{i,j-1} + C[U]^N_{i,j}u_{i,j+1}
\]

\[
\int_{\Gamma_{i,j}} (\mathbf{v} \cdot \mathbf{n}) vdS \approx C[U]^W_{i,j}v_{i-1,j} + C[U]^E_{i,j}v_{i+1,j} + C[U]^P_{i,j}v_{i,j} + C[U]^S_{i,j}v_{i,j-1} + C[U]^N_{i,j}v_{i,j+1}.
\]  

(8.1)  

(8.2)

To impose that the discretization of the convective terms leads to a skew-symmetric matrix, the following conditions must be fulfilled

\[ C[U] = -C[U]^T. \]  

(8.3)

Thus concretely, this means

\[ C[U]^P_{i,j} = 0 \]  

(8.4)

\[ C[U]^E_{i,j} = -C[U]^W_{i+1,j} \]  

(8.5)

\[ C[U]^N_{i,j} = -C[U]^S_{i,j+1}. \]  

(8.6)
8.1 Normal fluid cell

8.1.1 Control volume for $u$

For a fluid cell, the control volume corresponding with the velocity $u_{i,j}$ is shown in Fig. 8.1.

The boundary integrals for the control volume $\Omega_{u_{i,j}}^u$ are

$$\int_{\Gamma_{u_{i,j}}^u} (v \cdot n) u dS = - \int_{\Gamma_{u_{i,j}}^w} (v \cdot e_x) u dy + \int_{\Gamma_{u_{i,j}}^e} (v \cdot e_x) u dy$$

$$- \int_{\Gamma_{u_{i,j}}^{s,e} \cup \Gamma_{i+1,j}^{s,w}} (v \cdot e_y) u dx + \int_{\Gamma_{u_{i,j}}^{n,e} \cup \Gamma_{i+1,j}^{n,w}} (v \cdot e_y) u dx.$$  \hspace{1cm} (8.7)

Next, for each of the boundary integrals the convective fluxes are calculated as

$$\int_{\Gamma_{u_{i,j}}^{u,w}} (v \cdot n) u dS = - \frac{u_{i-1,j} + u_{i,j}}{2} + \frac{u_{i,j}}{2}, \hspace{1cm} (8.8)$$

$$\int_{\Gamma_{u_{i,j}}^{s,e}} (v \cdot n) u dS = - \frac{1}{2} \frac{u_{i,j-1} + u_{i,j}}{2}, \hspace{1cm} (8.9)$$

$$\int_{\Gamma_{u_{i,j}}^{n,e}} (v \cdot n) u dS = \frac{1}{2} \frac{u_{i,j} + u_{i,j+1}}{2}, \hspace{1cm} (8.10)$$

$$\int_{\Gamma_{u_{i,j}}^{s,w}} (v \cdot n) u dS = \frac{u_{i,j} + u_{i+1,j}}{2} - \frac{u_{i,j}}{2}, \hspace{1cm} (8.11)$$

$$\int_{\Gamma_{u_{i,j}}^{s,w}} (v \cdot n) u dS = - \frac{1}{2} \frac{u_{i,j-1} + u_{i+1,j-1}}{2}, \hspace{1cm} (8.12)$$

$$\int_{\Gamma_{u_{i,j}}^{n,w}} (v \cdot n) u dS = \frac{1}{2} \frac{u_{i,j} + u_{i,j+1}}{2}. \hspace{1cm} (8.13)$$
From this, we can deduce the corresponding components for the 5-points stencil of the convective fluxes, which are given as

\[
C[U]_W(u, i, j) = -\frac{1}{4} (\bar{u}_{i-1,j} + \bar{u}_{i,j}),
\]
\[
C[U]_E(u, i, j) = \frac{1}{4} (\bar{u}_{i,j} + \bar{u}_{i+1,j}),
\]
\[
C[U]_P(u, i, j) = \frac{1}{4} (-\bar{u}_{i-1,j} + \bar{u}_{i,j} - \bar{v}_{i,j} + \bar{v}_{i+1,j} - \bar{v}_{i+1,j-1} + \bar{v}_{i,j}),
\]
\[
= \frac{1}{4} (\dot{m}_{i,j} + \dot{m}_{i+1,j}) = 0,
\]
\[
C[U]_N(u, i, j) = \frac{1}{4} (\bar{v}_{i,j} + \bar{v}_{i+1,j}),
\]
\[
C[U]_S(u, i, j) = -\frac{1}{4} (\bar{v}_{i,j-1} + \bar{v}_{i+1,j-1}),
\]

which verifies the anti-symmetric conditions (8.4), (8.5) and (8.6) for the convective fluxes.

### 8.1.2 Control volume for \( v \)

After formulating the control volume \( \Omega_{i,j}^u \), it is the turn for the control volume for \( v \), \( \Omega_{i,j}^v \)

![Figure 8.2](image)

Figure 8.2: Control volume \( \Omega_{i,j}^v \).

The boundary integrals for the control volume \( \Omega_{i,j}^v \) are
\[
\int_{\Gamma_{i,j}} (v \cdot n) dvS = -\int_{\Gamma_{i,j}} (v \cdot e_y) dxdy + \int_{\Gamma_{i,j}} (v \cdot e_y) dxdy - \int_{\Gamma_{i,j} \cup \Gamma_{i,j+1}} (v \cdot e_x) dydS
\]
\[(8.14)\]

Then for each boundary integral the convective fluxes are described as
\[
\int_{\Gamma_{i,j}} (v \cdot n) dvS = \frac{1}{2} \overline{u}_{i,j},
\]
\[(8.15)\]
\[
\int_{\Gamma_{i,j}} (v \cdot n) dvS = -\frac{1}{2} \overline{v}_{i,j-1} + \overline{v}_{i,j} \frac{v_{i,j-1} + v_{i,j}}{2},
\]
\[(8.16)\]
\[
\int_{\Gamma_{i,j}} (v \cdot n) dvS = -\frac{1}{2} \overline{u}_{i-1,j} \frac{v_{i,j} + v_{i-1,j}}{2},
\]
\[(8.17)\]
\[
\int_{\Gamma_{i,j}} (v \cdot n) dvS = \frac{1}{2} \overline{u}_{i,j+1},
\]
\[(8.18)\]
\[
\int_{\Gamma_{i,j}} (v \cdot n) dvS = \frac{1}{2} \overline{v}_{i,j+1} + \overline{v}_{i,j} \frac{v_{i,j} + v_{i,j+1}}{2},
\]
\[(8.19)\]
\[
\int_{\Gamma_{i,j}} (v \cdot n) dvS = -\frac{1}{2} \overline{u}_{i-1,j+1} \frac{v_{i,j} + v_{i-1,j}}{2}.
\]
\[(8.20)\]

From the boundary integrals, the coefficients for the convective fluxes can be derived
\[
C[U]_W(i,j) = -\frac{1}{4} (\overline{u}_{i-1,j} + \overline{u}_{i-1,j+1}),
\]
\[
C[U]_E(i,j) = \frac{1}{4} (\overline{u}_{i,j} + \overline{u}_{i,j+1}),
\]
\[
C[U]_P(i,j) = \frac{1}{4} (\overline{v}_{i,j} - \overline{v}_{i,j-1} - \overline{v}_{i,j} - \overline{v}_{i-1,j-1} + \overline{v}_{i,j+1} + \overline{v}_{i,j+1} + \overline{v}_{i,j+1} - \overline{v}_{i-1,j+1}),
\]
\[
= \frac{1}{4} (\overline{m}_{i,j} + \overline{m}_{i,j+1}) = 0,
\]
\[
C[U]_N(i,j) = \frac{1}{4} (\overline{v}_{i,j} + \overline{v}_{i,j+1}),
\]
\[
C[U]_S(i,j) = -\frac{1}{4} (\overline{v}_{i,j-1} + \overline{v}_{i,j}).
\]

If we check above components with the demands for the skew-symmetric convective matrix ( (8.4), (8.5) and (8.6) ), it turns out that these requirements are fulfilled.
8.2 South trapezoidal cut-cell

In the following, the control volumes are divided into a left half of control volume and a right half of control volume $\Omega^u_{i,j}$. The reason is that for determining the control volume $\Omega^u_{i,j}$, the neighbour of the current cut-cell is needed. In 2D there are 12 cut-cells and 1 fluid cell, which means that there can be up to 13 combinations with the current cut-cell leading to an inefficient implementation. Therefore, we only look at the current cut-cell and store the left and right half of control volume separately. By doing so, it does not matter what type of cut-cell the neighbouring cell is, because every valid combination can be constructed. This means that for example $C[\overline{U}]^u_{NL}(i,j) + C[\overline{U}]^u_{NR}(i,j) = C[\overline{U}]^u_N(i,j)$.

8.2.1 Control volume for $u$

![Control volume Ω^u_{i,j}.](image)

**Left half of control volume**

We begin with the construction of the left half of control volume. The boundary integrals for the left half of control volume $\Omega^u_{i,j}$ are

$$\int_{\Gamma_{i,j}} (\mathbf{v} \cdot \mathbf{n}) dS = \int_{\Gamma^e_{i,j}} (\mathbf{v} \cdot \mathbf{n}) dS + \int_{\Gamma^w_{i,j}} (\mathbf{v} \cdot \mathbf{n}) dS + \int_{\Gamma^{ib,e}_{i,j}} (\mathbf{v} \cdot \mathbf{n}) dS.$$  \hspace{1cm} (8.21)

The situation depicted in Fig. 8.3(b) requires a different formulation for the boundaries of the control volume. Instead of $\Gamma^e_{i,j}$, $\Gamma^{ib,e}_{i,j}$ is used and the contribution, which is similar to (8.10), is computed as
\[
\int_{\Gamma_{i,j}^{ib,e}} (\mathbf{v} \cdot \mathbf{n}) \, dS = \frac{1}{2} U_{i,j}^{ib} \left( \frac{1}{2} u_{i,j} + \frac{1}{2} u(x_i, y_{i,j}^{ib}) \right),
\]

\[
= \frac{1}{2} \left( u_{i,j}^{ib} [n_x \Delta S]_{i,j}^{ib} + v_{i,j}^{ib} [n_y \Delta S]_{i,j}^{ib} \right) \left( \frac{1}{2} u_{i,j} + \frac{1}{2} u(x_i, y_{i,j}^{ib}) \right),
\]

\[
= \frac{1}{2} \left( u_{i,j}^{ib} [\theta_{i-1,j}^{u} - \theta_{i,j}^{u}] \Delta y_j + v_{i,j}^{ib} [\theta_{i,j-1}^{v} - \theta_{i,j}^{v}] \Delta x_i \right) \left( \frac{1}{2} u_{i,j} + \frac{1}{2} u(x_i, y_{i,j}^{ib}) \right).
\]

The corresponding components for the 5-points stencil of the convective fluxes are given as

\[
C[\mathbf{U}]_{W_L}^u(i, j) = -\frac{1}{4} (\overline{u}_{i-1,j} + \overline{u}_{i,j}), \tag{8.22}
\]

\[
C[\mathbf{U}]_{E_L}^u(i, j) = 0, \tag{8.23}
\]

\[
C[\mathbf{U}]_{P_L}^u(i, j) = -\frac{1}{4} \left( \overline{u}_{i-1,j} + \overline{u}_{i,j} + \overline{u}_{i,j-1} + \overline{U}_{i,j}^{ib} \right), \tag{8.24}
\]

\[
C[\mathbf{U}]_{N_L}^u(i, j) = 0, \tag{8.25}
\]

\[
C[\mathbf{U}]_{S_L}^u(i, j) = -\frac{1}{4} \overline{u}_{i,j-1}, \tag{8.26}
\]

\[
S_{i,j,L}^{ib,cu} = \frac{1}{4} U_{i,j}^{ib} u(x_i, y_{i,j}^{ib}). \tag{8.27}
\]

**Right half of control volume**

The construction for the right half of control volume will be discussed next. We start with the boundary integrals for the right half of control volume \(\Omega_{i,j}^u\)

\[
\int_{\Gamma_{i,j}^u} (\mathbf{v} \cdot \mathbf{n}) \, dS = \int_{\Gamma_{i,j}^{i+1,w}} (\mathbf{v} \cdot \mathbf{n}) \, dS + \int_{\Gamma_{i,j}^{u,e}} (\mathbf{v} \cdot \mathbf{n}) \, dS + \int_{\Gamma_{i+1,j}^{ib,w}} (\mathbf{v} \cdot \mathbf{n}) \, dS. \tag{8.28}
\]

Fig. 8.3(b) shows that for the boundary integral \(\Gamma_{i,j}^{ib,w}\) a different formulation has to be formulated. This boundary is computed as

\[
\int_{\Gamma_{i+1,j}^{ib,w}} (\mathbf{v} \cdot \mathbf{n}) \, dS = \frac{1}{2} U_{i+1,j}^{ib} \left( \frac{1}{2} u_{i,j} + \frac{1}{2} u(x_i, y_{i,j}^{ib}) \right),
\]

\[
= \frac{1}{2} \left( u_{i+1,j}^{ib} [\theta_{i,j}^{u} - \theta_{i+1,j}^{u}] \Delta y_j + v_{i+1,j}^{ib} [\theta_{i+1,j-1}^{v} - \theta_{i+1,j}^{v}] \Delta x_{i+1} \right),
\]

\[
= \left( \frac{1}{2} u_{i,j} + \frac{1}{2} u(x_i, y_{i,j}^{ib}) \right).
\]
8.2. SOUTH TRAPEZOIDAL CUT-CELL

The remaining components for the 5-points structure are stated below

\[ C[U]_{WR}^{u}(i,j) = 0, \quad (8.29) \]
\[ C[U]_{ER}^{u}(i,j) = \frac{1}{4}(\overline{u}_{i,j} + \overline{u}_{i+1,j}), \quad (8.30) \]
\[ C[U]_{PR}^{u}(i,j) = \frac{1}{4}\left(\overline{u}_{i,j} + \overline{u}_{i+1,j} - \overline{v}_{i+1,j-1} + \overline{U}_{i+1,j}^{ib}\right), \quad (8.31) \]
\[ C[U]_{NR}^{u}(i,j) = 0, \quad (8.32) \]
\[ C[U]_{SR}^{u}(i,j) = -\frac{1}{4}\overline{v}_{i+1,j-1}, \quad (8.33) \]
\[ S_{i,j}^{ib,v} = \frac{1}{4}\overline{U}_{i+1,j}^{ib}u(x_{i},y_{i,j}). \quad (8.34) \]

8.2.2 Control volume for v

The construction for the control volume \( \Omega_{i,j}^{v} \) is analogous to the configuration of the control volume \( \Omega_{i,j}^{u} \). The \( \Omega_{i,j}^{v} \) will be divided in a lower half and an upper half of control volume \( \Omega_{i,j}^{v} \). From Fig. 8.4 it is clear that the lower half of control volume \( \Omega_{i,j}^{v} \) is a 'normal' half of \( \Omega_{i,j}^{v} \). Therefore, we only describe the construction of the upper half of control volume \( \Omega_{i,j}^{v} \).

Upper half of control volume

The boundary integrals for the upper half of control volume \( \Omega_{i,j}^{v} \) are defined as

\[ \Gamma_{i,j}^{v,n} \]
\[ \Gamma_{i,j}^{v,s} \]
\[ \Gamma_{i,j}^{v,e} \]
\[ \Gamma_{i,j}^{v,w} \]

Figure 8.4: Control volume \( \Omega_{i,j}^{v} \).
\[ \int_{\Gamma_{i,j}}^{U} (\mathbf{v} \cdot \mathbf{n}) v dS = \int_{\Gamma_{i,j}^{e \rightarrow i,j+1}}^{e} (\mathbf{v} \cdot \mathbf{n}) v dS + \int_{\Gamma_{i,j}^{b,n}}^{b} (\mathbf{v} \cdot \mathbf{n}) v dS + \int_{\Gamma_{i,j}^{w \rightarrow i,j+1}}^{w} (\mathbf{v} \cdot \mathbf{n}) v dS. \]  

(8.35)

The situation depicted in Fig. 8.4 requires a different formulation for the boundary integral \( \Gamma_{i,j}^{v,n} \) compared to the case of a normal fluid cell. Instead \( \Gamma_{i,j}^{b,n} \) has been introduced and can be calculated as

\[ \int_{\Gamma_{i,j}^{b,n}} (\mathbf{v} \cdot \mathbf{n}) v dS = \frac{1}{2} \left( \overline{U}_{i,j}^{ib} + \overline{v}_{i,j} \right) \left( \frac{1}{2} v_{i,j} + \frac{1}{2} v_{i,j}^{ib} \right), \]

\[ = \frac{1}{2} \left( v_{i,j}^{ib} [\theta_{i-1,j}^{u} - \theta_{i,j}^{u}] \Delta y_{j} + v_{i,j}^{ib} [\theta_{i,j-1}^{v} - \theta_{i,j}^{v}] \Delta x_{i} + \overline{v}_{i,j} \right) \left( \frac{1}{2} v_{i,j} + \frac{1}{2} v_{i,j}^{ib} \right). \]

The coefficients can be deduced and are formulated as

\[ \mathcal{C}[\mathcal{U}]_{WU}(i,j) = -\frac{1}{4} \overline{u}_{i-1,j+1}, \]  

(8.36)

\[ \mathcal{C}[\mathcal{U}]_{E_U}(i,j) = \frac{1}{4} \overline{u}_{i,j+1}, \]  

(8.37)

\[ \mathcal{C}[\mathcal{U}]_{P_i}(i,j) = \frac{1}{4} \left( \overline{u}_{i,j+1} + \overline{v}_{i,j} - \overline{u}_{i-1,j+1} + \overline{U}_{i,j}^{ib} \right), \]  

(8.38)

\[ \mathcal{C}[\mathcal{U}]_{N_U}(i,j) = 0, \]  

(8.39)

\[ \mathcal{C}[\mathcal{U}]_{S_i}(i,j) = 0, \]  

(8.40)

\[ S_{i,jU}^{ib} = \frac{1}{4} \left( \overline{U}_{i,j+1}^{ib} + \overline{v}_{i,j} \right) v_{i,j}^{ib}. \]  

(8.41)

8.3 South-east triangular cut-cell

8.3.1 Control volume for \( u \)

When taking a look at Fig. 8.5 it is immediately clear that in case of a south-east triangular cut-cell it is not possible to create a right half of control volume \( \Omega_{i-1,j}^{u} \). Therefore, the construction of the left half of control volume \( \Omega_{i,j}^{u} \) will be only formulated

Left half of control volume

The boundary integrals for the left half of control volume \( \Omega_{i,j}^{u} \) are

\[ \int_{\Gamma_{i,j}^{L}}^{L} (\mathbf{v} \cdot \mathbf{n}) u dS = \int_{\Gamma_{i,j}^{e \rightarrow i,j}}^{e} (\mathbf{v} \cdot \mathbf{n}) u dS + \int_{\Gamma_{i,j}^{w \rightarrow i,j}}^{w} (\mathbf{v} \cdot \mathbf{n}) u dS + \int_{\Gamma_{i,j}^{b,e \rightarrow i,j}}^{b,e} (\mathbf{v} \cdot \mathbf{n}) u dS. \]  

(8.42)
The geometry shown in Fig. 8.5 requires a different formulation for two of the three boundaries of the control volume. First, the boundary integral $\Gamma_{i,j}^{n,e}$ has been replaced by $\Gamma_{i,j}^{ib,e}$ and also the boundary integral $\Gamma_{i,j}^{u,w}$ will be modified. These introduced boundary integrals are defined as

$$\int_{\Gamma_{i,j}^{n,e}} (\mathbf{v} \cdot \mathbf{n}) u dS = \frac{1}{2} \left( u_{i,j}^{ib} \left[ n_x \Delta S_{i,j}^{ib} - \bar{u}_{i,j} \right] \left( \frac{1}{2} u_{i,j} + \frac{1}{2} u_{i,j}^{ib} \right) \right),$$

$$= \frac{1}{2} \left( u_{i,j}^{ib} \left[ \theta_{i-1,j}^{u} - \theta_{i,j}^{u} \right] \Delta y_{j} - \bar{u}_{i,j} \right) \left( \frac{1}{2} u_{i,j} + \frac{1}{2} u_{i,j}^{ib} \right), \quad (8.43)$$

$$\int_{\Gamma_{i,j}^{ib,e}} (\mathbf{v} \cdot \mathbf{n}) u dS = \frac{1}{2} v_{i,j}^{ib} \left[ n_y \Delta S_{i,j}^{ib} \right] \left( \frac{1}{2} u_{i,j} + \frac{1}{2} u(x_i, y_{i,j}^{ib}) \right),$$

$$= \frac{1}{2} v_{i,j}^{ib} \left[ \theta_{i,j}^{v} - \theta_{i,j}^{v} \right] \Delta x_{i} \left( \frac{1}{2} u_{i,j} + \frac{1}{2} u(x_i, y_{i,j}^{ib}) \right). \quad (8.44)$$

Compared to a south trapezoidal cut-cell the term $U_{i,j}^{ib}$ is in this situation split up into two terms. The second term $v_{i,j}^{ib} \left[ \theta_{i,j}^{v} - \theta_{i,j}^{v} \right] \Delta x_{i}$ contributes to the boundary $\Gamma_{i,j}^{ib,e}$. The reason is that with this construction there exists a boundary integral at the north-east of the cut-cell. So at least three boundaries of the half of control volume can be constructed. The first term $u_{i,j}^{ib} \left[ \theta_{i-1,j}^{u} - \theta_{i,j}^{u} \right] \Delta y_{j}$ represents the boundary $\Gamma_{i,j}^{u,w}$ together with $\bar{u}_{i,j}$, which has a negative sign, due to the inflow.

The components for the five-point stencil of the convective fluxes are formulated as

$$C[\mathbf{U}]_{Wi}(i,j) = 0, \quad (8.45)$$

$$C[\mathbf{U}]_{Ei}(i,j) = 0, \quad (8.46)$$

$$C[\mathbf{U}]_{Pi}(i,j) = -\frac{1}{4} \left( \bar{u}_{i,j} + \bar{u}_{i,j-1} - \left( u_{i,j}^{ib} \left[ n_x \Delta S_{i,j}^{ib} + v_{i,j}^{ib} \left[ n_y \Delta S_{i,j}^{ib} \right] \right) \right), \quad (8.47)$$
\( C[\bar{U}]_{N_L}^u(i,j) = 0 \),
\( C[\bar{U}]_{S_L}^u(i,j) = -\frac{1}{4} \overline{v}_{i,j-1} \),
\( S_{i,j,L}^{ib,cu} = \frac{1}{4} \left( u_{i,j}^{ib} [\theta_{i-1,j}^{u} - \theta_{i,j}^{u}] \Delta y_j - \overline{u}_{i,j} \right) u_{i,j}^{ib}, 
+ \frac{1}{4} v_{i,j}^{ib} [\theta_{i,j-1}^{v} - \theta_{i,j}^{v}] \Delta x_i u(x_i, y_{i,j}^{ib}). \) 

(8.50)

8.3.2 Control volume for \( v \)

In the philosophy of Cheny et al., the \( \Omega_{i,j}^v \) will be divided in a lower half and an upper half of control volume \( \Omega_{i,j}^v \), the latter is not a problem. Therefore only the upper half will be treated.

Upper half of control volume

The construction of the upper half of the control volume \( \Omega_{i,j}^v \) as shown in Fig. 8.6 will be discussed now. The boundary integrals for the upper half of control volume \( \Omega_{i,j}^v \) are formulated as

\[
\int_{\Gamma_{i,j}^v} (\mathbf{v} \cdot \mathbf{n}) \, dS = \int_{\Gamma_{i,j+1}^{w,n}} (\mathbf{v} \cdot \mathbf{n}) \, dS + \int_{\Gamma_{i,j}^{w,n}} (\mathbf{v} \cdot \mathbf{n}) \, dS + \int_{\Gamma_{i,j}^{ib,s}} (\mathbf{v} \cdot \mathbf{n}) \, dS.
\]

(8.51)
\[ 8.4. \ \textit{SOUTH-EAST PENTAGONAL CUT-CELL} \]

\[ \Gamma_{i,j}^{\text{ib},s} \] and also the boundary integral \( \Gamma_{i,j}^{\text{ib},n} \) has been modified. These modified boundary integrals are computed as

\[
\int_{\Gamma_{i,j}^{\text{ib},s}} (\mathbf{v} \cdot \mathbf{n}) d\mathbf{S} = \frac{1}{2} \left( v_{i,j+1}^{\text{ib}} [n_y \Delta S]_{i,j+1}^{\text{ib}} + v_{i,j} \right) \left( \frac{1}{2} v_{i,j} + \frac{1}{2} v_{i,j+1}^{\text{ib}} \right),
\]

\[
= \frac{1}{2} \left( v_{i,j+1}^{\text{ib}} [\theta_{i,j}^{\nu} - \theta_{i,j+1}^{\nu}] \Delta x_i + v_{i,j} \right) \left( \frac{1}{2} v_{i,j} + \frac{1}{2} v_{i,j+1}^{\text{ib}} \right), \quad (8.52)
\]

\[
\int_{\Gamma_{i,j+1}^{\text{ib},s}} (\mathbf{v} \cdot \mathbf{n}) d\mathbf{S} = \frac{1}{2} u_{i,j+1}^{\text{ib}} [n_x \Delta S]_{i,j+1}^{\text{ib}} \left( \frac{1}{2} v_{i,j} + \frac{1}{2} v(x_{i,j}, y_j) \right),
\]

\[
= \frac{1}{2} u_{i,j+1}^{\text{ib}} [\theta_{i,j}^{\mu} - \theta_{i,j+1}^{\mu}] \Delta y_{j+1} \left( \frac{1}{2} v_{i,j} + \frac{1}{2} v(x_{i,j}, y_j) \right). \quad (8.53)
\]

The coefficients for the 5-point stencil are formulated as

\[
C[\mathcal{U}]_{W_U}^{\nu} (i, j) = 0, \quad (8.54)
\]

\[
C[\mathcal{U}]_{E_U}^{\nu} (i, j) = \frac{1}{4} v_{i,j+1}, \quad (8.55)
\]

\[
C[\mathcal{U}]_{\beta_U}^{\nu} (i, j) = \frac{1}{4} (v_{i,j} + v_{i,j+1} + (v_{i,j+1}^{\text{ib}} [n_y \Delta S]_{i,j+1}^{\text{ib}} + u_{i,j+1}^{\text{ib}} [n_x \Delta S]_{i,j+1}^{\text{ib}})), \quad (8.56)
\]

\[
C[\mathcal{U}]_{N_U}^{\nu} (i, j) = 0, \quad (8.57)
\]

\[
C[\mathcal{U}]_{S_U}^{\nu} (i, j) = 0, \quad (8.58)
\]

\[
S_{i,j+1}^{\nu} = \frac{1}{4} \left( v_{i,j+1}^{\text{ib}} [n_y \Delta S]_{i,j+1}^{\text{ib}} + v_{i,j} \right) \cdot v_{i,j+1},
\]

\[
+ \frac{1}{4} u_{i,j+1}^{\text{ib}} [n_x \Delta S]_{i,j+1}^{\text{ib}} \cdot v(x_{i,j}, y_j). \quad (8.59)
\]

### 8.4 South-east pentagonal cut-cell

#### 8.4.1 Control volume for \( u \)

**Left half of control volume**

We begin with the construction of the left half of control volume \( \Omega_{i,j}^{\nu} \) in Fig. 8.7(a). The boundary integrals for the left half of control volume \( \Omega_{i,j}^{\nu} \), for situation 1, are

\[
\int_{\Gamma_{i,j}^{\nu}} (\mathbf{v} \cdot \mathbf{n}) d\mathbf{S} = \int_{\Gamma_{i,j}^{\text{se}}} (\mathbf{v} \cdot \mathbf{n}) d\mathbf{S} + \int_{\Gamma_{i,j}^{\text{sw}}} (\mathbf{v} \cdot \mathbf{n}) d\mathbf{S} + \int_{\Gamma_{i,j}^{\text{ib}}} (\mathbf{v} \cdot \mathbf{n}) d\mathbf{S} + \int_{\Gamma_{i,j}^{\text{ne}}} (\mathbf{v} \cdot \mathbf{n}) d\mathbf{S}. \quad (8.60)
\]

In this situation an extra boundary integral is introduced, named \( \Gamma_{i,j}^{\text{ib},e} \), for reasons that will be explained further on in this section. The introduced boundary \( \Gamma_{i,j}^{\text{ib},e} \), is defined as
\[ \int_{\Gamma_{ib}} (\mathbf{v} \cdot \mathbf{n}) \, u \, dS = \frac{1}{2} U_{i,j}^{ib} \left( \frac{1}{2} u_{i,j} + \frac{1}{2} u(x_{i,j}, y_j) \right), \]

\[ = \frac{1}{2} \left( u_{i,j}^{ib} [\theta_{i-1,j} - \theta_{i,j}] \Delta y_j + v_{i,j}^{ib} [\theta_{i,j-1} - \theta_{i,j}] \Delta x_i \right) \left( \frac{1}{2} u_{i,j} + \frac{1}{2} u(x_{i,j}, y_j) \right). \]

The term \( u_{i,j}^{ib} [\theta_{i-1,j} - \theta_{i,j}] \Delta y_j \) may not be added by the boundary \( \Gamma_{i,j}^{u,w} \), because then the skew-symmetric convection relation

\[ C[\mathcal{U}]_E(i, j) = -C[\mathcal{U}]_W(i, j + 1), \quad (8.61) \]

does not hold. The same argument holds for the term \( v_{i,j}^{ib} [\theta_{i,j-1} - \theta_{i,j}] \Delta x_i \). The corresponding components for the five point stencil are defined as

\[ C[\mathcal{U}]_{W_L}(i, j) = -\frac{1}{4} (\bar{u}_{i-1,j} + \bar{u}_{i,j}), \quad (8.62) \]

\[ C[\mathcal{U}]_{E_L}(i, j) = 0, \quad (8.63) \]

\[ C[\mathcal{U}]_{P_L}(i, j) = -\frac{1}{4} \left( \bar{u}_{i-1,j} + \bar{u}_{i,j} + \bar{v}_{i,j-1} - \bar{v}_{i,j} - U_{i,j}^{ib} \right), \quad (8.64) \]

\[ C[\mathcal{U}]_{N_L}(i, j) = \frac{1}{4} \bar{v}_{i,j}, \quad (8.65) \]

\[ C[\mathcal{U}]_{S_L}(i, j) = -\frac{1}{4} \bar{v}_{i,j-1}, \quad (8.66) \]

\[ S_{i,j,L}^{ib,c} = \frac{1}{4} U_{i,j}^{ib} \cdot u(x_{i,j}, y_j). \quad (8.67) \]
8.4. SOUTH-EAST PENTAGONAL CUT-CELL

Right half of control volume

The construction for the right half of control volume will be discussed next. The boundary integrals for the right half of control volume $\Omega_{i,j}$ in Fig. 8.7(b) are

$$\int_{\Gamma_{i,j}} (\mathbf{v} \cdot \mathbf{n}) dS = \int_{\Gamma_{i+1,j}} (\mathbf{v} \cdot \mathbf{n}) dS + \int_{\Gamma_{i,j}} (\mathbf{v} \cdot \mathbf{n}) dS + \int_{\Gamma_{i+1,j}} (\mathbf{v} \cdot \mathbf{n}) dS. \quad (8.68)$$

In the configuration pointed out in situation 2 leads to a different formulation for the boundary integral $\Gamma_{i+1,j}^{n,e}$. The boundary integral $\Gamma_{i+1,j}^{ib,w}$, which replace $\Gamma_{i+1,j}^{n,e}$ is formulated as

$$\int_{\Gamma_{i+1,j}^{ib,w}} (\mathbf{v} \cdot \mathbf{n}) dS = \frac{1}{2} \left( \bar{U}_{i+1,j} + \bar{v}_{i+1,j} \right) \left( \frac{1}{2} u_{i,j} + \frac{1}{2} u_{i+1,j} \right),$$

$$= \frac{1}{2} \left( u_{i+1,j} \left[ \theta_{i,j}^u - \theta_{i+1,j}^u \right] \Delta y_j + \bar{v}_{i+1,j} \left[ \theta_{i,j}^v - \theta_{i+1,j}^v \right] \Delta x_i \right),$$

$$\left( \frac{1}{2} u_{i+1,j} + \frac{1}{2} u_{i+1,j} \right) \left( \frac{1}{2} v_{i+1,j} + \frac{1}{2} v_{i+1,j} \right).$$

The construction of the boundary integrals for the convective fluxes differs from the boundary integrals corresponding to trapezoidal cells and triangular cells. The reason behind this will be explained now. The term $\bar{v}_{i+1,j}$ is included with the boundary $\Gamma_{i+1,j}^{ib,w}$. If we introduce the integral boundary $\Gamma_{i+1,j}^{n,e}$ where $\bar{v}_{i+1,j}$ is included, the following condition has to hold

$$C\bar{U}_N(i,j) = -C\bar{U}_S(i,j + 1). \quad (8.69)$$

However, the velocity $u_{i+1,j+1}$ does not exist following the LS-STAG philosophy. If $u_{i+1,j+1}$ is present, this results in a not unique determination of $\frac{\partial u}{\partial y}$, namely formulae (9.8) and (9.10) can be both used. It could be allowed that $u_{i+1,j+1}$ exist in this situation, but then the formulae (9.10) must be used and not formulae (9.8). The only cases which are valid are depicted in Fig. 8.8.

Thus by including the term $\bar{v}_{i,j}$ with $\Gamma_{i,j}^{ib,w}$ above skew-symmetric condition does not have to be fulfilled and can be omitted therefore. After defining the boundary integrals, the coefficients for the 5-point stencil for convective fluxes can be deduced and are given by
CHAPTER 8. DISCRETIZATION OF THE CONVECTIVE FLUXES

8.4.2 Control volume for $v$

Again, following the idea of Cheny et al., the control volume $\Omega_{i,j}^v$ will be divided in a lower half and an upper half of control volume $\Omega_{i,j}^u$. First, we describe the construction of the former in situation 1.

Lower half of control volume

The boundary integrals for the lower half of control volume $\Omega_{i,j}^v$, shown in Fig. 8.9(a), are

\begin{align*}
C[\overline{U}]_{WR}^v(i,j) &= 0, \quad (8.70) \\
C[\overline{U}]_{ER}^w(i,j) &= \frac{1}{4}(\overline{u}_{i,j} + \overline{u}_{i+1,j}), \quad (8.71) \\
C[\overline{U}]_{FR}^w(i,j) &= \frac{1}{4}(\overline{u}_{i,j} + \overline{u}_{i+1,j} - \overline{v}_{i+1,j} - 1 + \overline{v}_{i+1,j} + \overline{U}_{ib}^{ib}), \quad (8.72) \\
C[\overline{U}]_{NR}^w(i,j) &= 0, \quad (8.73) \\
C[\overline{U}]_{SR}^w(i,j) &= -\frac{1}{4}\overline{v}_{i+1,j}, \quad (8.74) \\
\mathcal{S}_{ib,c}^{ib,c} &= \frac{1}{4}(\overline{U}_{ib}^{ib} + \overline{U}_{ib}^{ib}) \cdot \overline{u}(x_i, y_{i,j}). \quad (8.75)
\end{align*}
\[ \int_{\Gamma_{i,j}}^L (\mathbf{v} \cdot \mathbf{n}) dS = \int_{\Gamma_{i,j}^{v,n}}^L (\mathbf{v} \cdot \mathbf{n}) dS + \int_{\Gamma_{i,j}^{e,n}}^L (\mathbf{v} \cdot \mathbf{n}) dS + \int_{\Gamma_{i,j}^{v,s}}^L (\mathbf{v} \cdot \mathbf{n}) dS + \int_{\Gamma_{i,j}^{w,n}}^L (\mathbf{v} \cdot \mathbf{n}) dS. \] (8.76)

The configuration in this case leads to a different formulation for the boundary integral \( \Gamma_{i,j}^{w,n} \). This term will be replaced by \( \Gamma_{i,j}^{ib,n} \) which is defined as

\[ \int_{\Gamma_{i,j}^{ib,n}} (\mathbf{v} \cdot \mathbf{n}) dS = \frac{1}{2} \left( \overline{U}_{i,j}^{ib} - \overline{u}_{i-1,j} \right) \left( \frac{1}{2} v_{i,j} + \frac{1}{2} v(x_{i,j}^{ib}, y_j) \right), \]

\[ = \frac{1}{2} \left( u_{i,j}^{ib} \left[ \theta_{i,j} - \theta_{i,j-1} \right] \Delta y_j + v_{i,j}^{ib} \left[ \theta_{i-1,j} - \theta_{i,j} \right] \Delta x_i \right), \]

\[ \left( \frac{1}{2} v_{i,j} + \frac{1}{2} v(x_{i,j}^{ib}, y_j) \right) - \frac{1}{2} \overline{u}_{i-1,j} \left( \frac{1}{2} v_{i,j} + \frac{1}{2} v(x_{i,j}^{ib}, y_j) \right). \]

At this point the coefficients for the five-point structure can be deduced and are given as

\[ C[\overline{U}]_{W_L} (i, j) = 0, \] (8.77)

\[ C[\overline{U}]_{E_L} (i, j) = \frac{1}{4} \overline{u}_{i,j}, \] (8.78)

\[ C[\overline{U}]_{P_L} (i, j) = -\frac{1}{4} \left( \overline{u}_{i-1,j} - \overline{u}_{i,j} + \overline{v}_{i,j-1} + \overline{v}_{i,j} - \overline{U}_{i,j}^{ib} \right). \] (8.79)
CHAPTER 8. DISCRETIZATION OF THE CONVECTIVE FLUXES

\[ C[\overline{U}]_{\overline{V}_{L}}(i,j) = 0, \quad (8.80) \]
\[ C[\overline{U}]_{\overline{V}_{S_{L}}}(i,j) = -\frac{\overline{v}_{i-1,j} + \overline{v}_{i,j}}{4}, \quad (8.81) \]
\[ S_{ib,c}^{ib,c_{v}} = \frac{1}{4} \left( \overline{U}_{ib}^{ib} - \overline{u}_{i-1,j} \right) \cdot v(x_{i-1,j}, y_{j}). \quad (8.82) \]

Upper half of control volume

The construction for the upper half of the control volume \( \Omega_{i,j}^{b} \) in situation 2, see Fig. 8.9(b), will be treated now. The corresponding boundary integrals for the lower half of control volume \( \Omega_{i,j}^{l} \) are

\[ \int_{\Gamma_{e,s, i,j+1}} (\mathbf{v} \cdot \mathbf{n}) \, d\mathbf{s} = \int_{\Gamma_{e,s, i,j+1}} (\mathbf{v} \cdot \mathbf{n}) \, d\mathbf{s} + \int_{\Gamma_{e,n, i,j+1}} (\mathbf{v} \cdot \mathbf{n}) \, d\mathbf{s} + \int_{\Gamma_{ib,s, i,j+1}} (\mathbf{v} \cdot \mathbf{n}) \, d\mathbf{s} + \int_{\Gamma_{w,n, i,j+1}} (\mathbf{v} \cdot \mathbf{n}) \, d\mathbf{s}. \quad (8.83) \]

In this case, an extra boundary integral is introduced, due to fulfilling the skew-symmetric condition for convection. This introduced term \( \Gamma_{ib,s, i,j+1} \), is formulated as

\[ \int_{\Gamma_{ib,s, i,j+1}} (\mathbf{v} \cdot \mathbf{n}) \, d\mathbf{s} = \frac{1}{2} \overline{U}_{ib}^{ib} \left( \frac{1}{2} \overline{v}_{i,j} + \frac{1}{2} v(x_{i-1}, y_{i-1,j+1}) \right), \]
\[ = \frac{1}{2} \left( u_{i,j+1}^{ib} [\theta_{i-1,j+1}^{u} - \theta_{i,j+1}^{u}] \Delta y_{j+1} + v_{i,j+1}^{ib} [\theta_{i,j}^{v} - \theta_{i,j+1}^{v}] \Delta x_{i} \right), \]
\[ = \left( \frac{1}{2} v_{i,j} + \frac{1}{2} v(x_{i-1}, y_{i-1,j+1}) \right). \quad (8.84) \]

The components are formulated as

\[ C[\overline{U}]_{W_{U}}(i,j) = -\frac{1}{4} \overline{u}_{i-1,j+1}, \quad (8.85) \]
\[ C[\overline{U}]_{E_{U}}(i,j) = \frac{1}{4} \overline{u}_{i,j+1}, \quad (8.86) \]
\[ C[\overline{U}]_{P_{U}}(i,j) = -\frac{1}{4} \left( \overline{u}_{i-1,j+1} - \overline{u}_{i,j+1} - \overline{v}_{i,j} - \overline{v}_{i,j+1} - \overline{U}_{ib}^{ib} \right), \quad (8.87) \]
\[ C[\overline{U}]_{N_{U}}(i,j) = \frac{1}{4} (\overline{v}_{i,j} + \overline{v}_{i,j+1}), \quad (8.88) \]
\[ C[\overline{U}]_{S_{U}}(i,j) = 0, \quad (8.89) \]
\[ S_{ib,c}^{ib,c_{v}} = \frac{1}{4} \overline{U}_{ib}^{ib} \cdot v(x_{i-1}, y_{i-1,j+1}). \quad (8.90) \]
Chapter 9

Discretization of the viscous fluxes

9.1 General considerations

For the x-momentum equation, the viscous terms written in control volume $\Omega_{x,i,j}$ reads

$$\int_{\Gamma_{x,i,j}} \nabla u \cdot \text{n} \, dS = \int_{\Gamma_{x,i,j}} \frac{\partial u}{\partial x} e_x \cdot \text{n} \, dS + \int_{\Gamma_{x,i,j}} \frac{\partial u}{\partial y} e_y \cdot \text{n} \, dS.$$  \hspace{0.5cm} (9.1)

First and foremost, Cheny et al. make a distinction between normal stress fluxes, $\int_{\Gamma_{x,i,j}} \frac{\partial u}{\partial x} e_x \cdot \text{n} \, dS$ and the shear stress fluxes, $\int_{\Gamma_{x,i,j}} \frac{\partial u}{\partial y} e_y \cdot \text{n} \, dS$. We begin with the discretization of the normal stress fluxes.

9.1.1 Discretization of the normal stress fluxes

For the normal stress fluxes, a geometric based formula would consist in writing this term as the net flux through the east $\Gamma_{x,i,j}^{e}$ and the west $\Gamma_{x,i,j}^{w}$ faces, and then discretize each of these terms with a differential quotient

$$\int_{\Gamma_{x,i,j}^{w}} \frac{\partial u}{\partial x} e_x \cdot \text{n} \, dS \approx \Delta y_{x,i,j} \frac{u_{i,j} - u_{i-1,j}}{\Delta x_i},$$  \hspace{0.5cm} (9.2)

where the area $\Delta y_{x,i,j}^{w}$ is yet to be defined. Despite of all the efforts Cheny et al. made in this direction, they found out that this gave disappointing results in terms of numerical accuracy. The reason is that the LS-STAG mesh is not admissible in the sense of Eymard et al. [10] for the normal stresses in the cut-cells, i.e. the line joining the location of $u_{i-1,j}$ and $u_{i,j}$ is not orthogonal to the face $\Gamma_{x,i,j}^{w}$ in the trapezoidal cell of [8.3(b)]. They observed that this feature also occurs for pentagonal cut-cells, resulting in a non-consistent approximation of (9.2) and thus yielding to large numerical errors. In order to improve the consistency of this term, Cheny et al. use the fact that the discrete normal stress should be consistent with the discrete pressure, as both are contributing to the diagonal.
part of the Cauchy stress tensor. The derivation of the normal stress is stated in §7.4. The discretization of the normal stress for the u and v momentum are, see (7.29) and (7.30)

\[
\frac{\partial u}{\partial x}_{i,j} \approx \frac{\theta u_{i,j} - \theta u_{i-1,j}u_{i-1,j} + u_{i,j}^{ib} \left( \theta u_{i-1,j} - \theta u_{i,j} \right)}{V_{i,j}/\Delta y_j},
\]

(9.3)

\[
\frac{\partial v}{\partial y}_{i,j} \approx \frac{\theta v_{i,j} - \theta v_{i,j-1}v_{i,j-1} + v_{i,j}^{ib} \left( \theta v_{i,j-1} - \theta v_{i,j} \right)}{V_{i,j}/\Delta x_i},
\]

(9.4)

9.1.2 Discretization of the shear stress fluxes

In contrast, the discretization of the shear stress flux \( \int_{\Gamma_{i,j}} \frac{\partial u}{\partial y} e_y \cdot n \, dS \) may seem simpler because the LS-STAG mesh is admissible for this term in the sense of Eymard et al. [10]. Therefore, the shear stress term can be written as the net flux through the north and south faces, for example away from the immersed boundary

\[
\int_{\Gamma_{i,j}} \frac{\partial u}{\partial y} e_y \cdot n \, dS = \int_{\Gamma_{i,j}^{n,s}}\frac{\partial u}{\partial y} \, dx - \int_{\Gamma_{i,j}^{s,w}}\frac{\partial u}{\partial y} \, dx.
\]

(9.5)

Applying the midpoint rule for the north face gives...
\[ \int_{\Gamma_{i,j}^{n,e} \cup \Gamma_{i+1,j}^{n,w}} \frac{\partial u}{\partial y} \, dx \approx \left( \Delta x_{i,j}^{n,e} + \Delta x_{i+1,j}^{n,w} \right) \frac{\partial u}{\partial y} \bigg|_{i,j}, \]  

(9.6)

where for the purpose of local conservation of the fluxes, the areas \( \Delta x_{i,j}^{n,e} \) and \( \Delta x_{i+1,j}^{n,w} \) represent only the fluid part of the faces, i.e.

\[ \Delta x_{i,j}^{n,e} = \frac{1}{2} \theta_{i,j}^{u} \Delta x_{i,j}, \quad \Delta x_{i,j}^{n,e} = \frac{1}{2} \theta_{i,j+1}^{u} \Delta x_{i+1,j}. \]  

(9.7)

The quotient \( \frac{\partial u}{\partial y} \big|_{i,j} \), located at the upper right corner of cell \( \Omega_{i,j} \), is computed by differentiating the interpolation polynomial of \( u(x, \cdot) \) in the vertical direction

\[ \frac{\partial u}{\partial y} \bigg|_{i,j} = \frac{u_{i,j+1} - u_{i,j}}{\frac{1}{2} \theta_{i,j+1}^{u} \Delta y_{j+1} + \frac{1}{2} \theta_{i,j}^{u} \Delta y_{j}}. \]  

(9.8)

These formulae are valid if \( u_{i,j+1} \) exists in the fluid domain, i.e. \( \theta_{i,j+1}^{u} > 0 \). When the north face is solid and thus \( u_{i,j+1} \) does not exist, i.e. \( \theta_{i,j+1}^{u} = 0 \), these formulae have to be modified for taking into account the boundary conditions. In the case of a solid north face

\[ \int_{\Gamma_{i,j}^{n,e} \cup \Gamma_{i+1,j}^{n,w}} \frac{\partial u}{\partial y} \, dx \approx \left( \Delta x_{i,j}^{n,e} + \Delta x_{i+1,j}^{n,w} \right) \frac{\partial u}{\partial y} \bigg|_{i,j}, \]  

(9.9)

with the corresponding differential quotient

\[ \frac{\partial u}{\partial y} \bigg|_{i,j} = \frac{u(x_{i,j}^{b}, y_{j}) - u_{i,j}}{\frac{1}{2} \theta_{i,j}^{u} \Delta y_{j}}. \]  

(9.10)

The shear stress flux \( \int_{\Gamma_{i,j}^{e,x}} \frac{\partial v}{\partial x} x \cdot n \, dS \) is derived in a similar way. It can be written as the net flux through the east and west faces, for example far from the immersed boundary

\[ \int_{\Gamma_{i,j}^{e,x}} \frac{\partial v}{\partial x} \, dS = \int_{\Gamma_{i,j}^{e,n} \cup \Gamma_{i,j+1}^{e,s}} \frac{\partial v}{\partial x} \, dy - \int_{\Gamma_{i,j}^{e,n} \cup \Gamma_{i,j+1}^{e,s}} \frac{\partial v}{\partial x} \, dy. \]  

(9.11)

Applying the midpoint rule for the east face gives

\[ \int_{\Gamma_{i,j}^{e,n} \cup \Gamma_{i,j+1}^{e,s}} \frac{\partial v}{\partial x} \, dx \approx \left( \Delta y_{i,j}^{e,n} + \Delta y_{i,j+1}^{e,s} \right) \frac{\partial v}{\partial x} \bigg|_{i,j}, \]  

(9.12)

where for the purpose of local conservation of the fluxes, the areas \( \Delta y_{i,j}^{e,n} \) and \( \Delta y_{i,j+1}^{e,s} \) represent only the fluid part of the faces, i.e.

\[ \Delta y_{i,j}^{e,n} = \frac{1}{2} \theta_{i,j}^{u} \Delta y_{j}, \quad \Delta y_{i,j}^{e,n} = \frac{1}{2} \theta_{i,j+1}^{u} \Delta y_{j+1}. \]  

(9.13)

The quotient \( \frac{\partial v}{\partial x} \big|_{i,j} \), located at the upper right corner of cell \( \Omega_{i,j} \), is computed by differentiating the interpolation polynomial of \( v(\cdot, y_{j}) \) in the vertical direction.
CHAPTER 9. DISCRETIZATION OF THE VISCOUS FLUXES

\[ \frac{\partial v}{\partial x} \bigg|_{i,j} = \frac{1}{2} \theta^v_{i+1,j} \Delta x_{i+1} + \frac{1}{2} \theta^v_{i,j} \Delta x_i. \]  

(9.14)

These formulae are valid if \( v_{i+1,j} \) exists in the fluid domain, i.e. \( \theta^v_{i+1,j} > 0 \). When the east face is solid and thus \( v_{i+1,j} \) does not exist, i.e. \( \theta^v_{i+1,j} = 0 \), these formulae have to be modified for taking into account the boundary conditions. In the case of a solid east face

\[ \int_{\Gamma_{ib,n}^{i+1,j} \cup \Gamma_{ib,s}^{i+1,j}} \frac{\partial v}{\partial x} dy \approx \left( \Delta y_{i,j}^{ib,n} + \Delta y_{i,j+1}^{ib,s} \right) \frac{\partial v}{\partial x} \bigg|_{i,j}, \]  

(9.15)

with the corresponding one-sided differential quotient

\[ \frac{\partial v}{\partial x} \bigg|_{i,j} = \frac{v(x_{i,j}, y_j) - v_{i,j}}{\frac{1}{2} \theta^v_{i,j} \Delta x_i}. \]  

(9.16)

9.1.3 Five-point structure for viscous term

The discretation of the viscous term has a five-point structure and is given as

\[
\int_{\Gamma_{i,j}^u} \nabla u \cdot ndS \approx \mathcal{K}_W^u(i,j) u_{i-1,j} + \mathcal{K}_E^u(i,j) u_{i+1,j} + \mathcal{K}_P^u(i,j) u_{i,j} \\
+ \mathcal{K}_S(i,j) u_{i,j-1} + \mathcal{K}_N(i,j) u_{i,j+1} \]  

(9.17)

\[
\int_{\Gamma_{i,j}^v} \nabla v \cdot ndS \approx \mathcal{K}_W^v(i,j) v_{i-1,j} + \mathcal{K}_E^v(i,j) v_{i+1,j} + \mathcal{K}_P^v(i,j) v_{i,j} \\
+ \mathcal{K}_S(i,j) v_{i,j-1} + \mathcal{K}_N(i,j) v_{i,j+1} \]  

(9.18)

for the x and the y direction respectively.

9.2 Normal fluid cell

In this section we consider a normal fluid cell where the apertures, such as \( \theta^u_{i,j} \), are for all faces equal to one. Nevertheless, in all coming equation, the apertures will still be formulated. The reason is that the global structure becomes clear, especially when considering cut-cells.

9.2.1 u-momentum

The viscous fluxes corresponding for \( \Omega_{i,j}^u \) are formulated as

\[
\int_{\Gamma_{i,j}^u} \nabla u \cdot ndS = \int_{\Gamma_{x,i,j}^u} \frac{\partial u}{\partial x} e_x \cdot ndS + \int_{\Gamma_{y,i,j}^u} \frac{\partial u}{\partial y} e_y \cdot ndS, \]  

(9.19)
where the normal stress fluxes are discretized according to (7.29)

\[
\int_{\Gamma_{i,j}} \frac{\partial u}{\partial x} e_x \cdot \mathbf{n} dS \approx \theta_{i,j}^u \Delta y_j \left( \frac{\partial u}{\partial x}_{i+1,j} - \frac{\partial u}{\partial x}_{i,j} \right),
\]

and the discrete shear stress flux follow from (9.8)

\[
\int_{\Gamma_{i,j}} \frac{\partial u}{\partial y} e_y \cdot \mathbf{n} dS = \int_{\Gamma_{i,j}} n, e_{i,j} \big| n, w_{i+1,j} \frac{\partial u}{\partial y} dx - \int_{\Gamma_{i,j}} s, e_{i,j} \big| s, w_{i+1,j} \frac{\partial u}{\partial y} dx \approx \left( \Delta x_{i,j}^{n,e} + \Delta x_{i+1,j}^{n,w} \right) \frac{u_{i,j+1} - u_{i,j}}{\frac{1}{2} \theta_{i,j+1} \Delta y_{j+1} + \frac{1}{2} \theta_{i,j} \Delta y_j} - \left( \Delta x_{i,j}^{s,e} + \Delta x_{i+1,j}^{s,w} \right) \frac{u_{i,j} - u_{i,j-1}}{\frac{1}{2} \theta_{i,j} \Delta y_j + \frac{1}{2} \theta_{i,j-1} \Delta y_{j-1}} \right). \tag{9.21}
\]

From the previous equations we can derive the components of the 5-points structure for the viscous fluxes. Like for the treatment of the convective fluxes in the cut-cells, the same concept is used for the viscous fluxes. First the left half of control volume \( \Omega_{i,j}^L \) will be formulated.

**Left half of control volume**

\[
\begin{align*}
K_{W_L}^u (i, j) &= \theta_{i,j}^u \Delta y_j \frac{\theta_{i-1,j}^u}{V_{i,j}/\Delta y_j}, \quad \tag{9.22} \\
K_{E_L}^u (i, j) &= 0, \quad \tag{9.23} \\
K_{P_L}^u (i, j) &= -\theta_{i,j}^u \Delta y_j \frac{\theta_{i-1,j}^u}{V_{i,j}/\Delta y_j} - \frac{\Delta x_{i,j}^{n,e}}{\frac{1}{2} \theta_{i,j+1} \Delta y_{j+1} + \frac{1}{2} \theta_{i,j} \Delta y_j}, \\
&\quad - \frac{\Delta x_{i,j}^{s,e}}{\frac{1}{2} \theta_{i,j} \Delta y_j + \frac{1}{2} \theta_{i,j-1} \Delta y_{j-1}}, \quad \tag{9.24} \\
K_{N_L}^u (i, j) &= \frac{\Delta x_{i,j}^{n,e}}{\frac{1}{2} \theta_{i,j+1} \Delta y_{j+1} + \frac{1}{2} \theta_{i,j} \Delta y_j}, \quad \tag{9.25}
\end{align*}
\]
CHAPTER 9. DISCRETIZATION OF THE VISCOUS FLUXES

\[
K_{SL}^{u}(i,j) = \frac{\Delta x_{i,j}^{s,e}}{2\theta_{i,j}^u \Delta y_j + \frac{1}{2}\theta_{i,j-1}^u \Delta y_{j-1}},
\]  
(9.26)

\[
S_{i,jL}^{ib,vu} = \theta_{i,j}^u \Delta y_j \left[ -\frac{\left( \theta_{i-1,j}^u - \theta_{i,j}^u \right) u_{i,j}^{ib}}{V_{i,j}/\Delta y_j} \right].
\]  
(9.27)

Right half of control volume

The coefficients coming from the right half of control volume are stated as

\[
K_{WR}^{u}(i,j) = 0,
\]  
(9.28)

\[
K_{ER}^{u}(i,j) = \theta_{i,j}^u \Delta y_j \frac{\theta_{i+1,j}^u}{V_{i+1,j}/\Delta y_j},
\]  
(9.29)

\[
K_{P_{R}}^{u}(i,j) = -\theta_{i,j}^u \Delta y_j \frac{\theta_{i+1,j}^u}{V_{i+1,j}/\Delta y_j} - \frac{\Delta x_{i+1,j}^{n,w}}{2\theta_{i,j}^u \Delta y_j + \frac{1}{2}\theta_{i,j-1}^u \Delta y_{j-1}},
\]  
(9.30)

\[
K_{NR}^{u}(i,j) = \frac{\Delta x_{i+1,j}^{n,w}}{2\theta_{i,j+1}^u \Delta y_{j+1} + \frac{1}{2}\theta_{i,j}^u \Delta y_j},
\]  
(9.31)

\[
K_{SR}^{u}(i,j) = \frac{\Delta x_{i+1,j}^{s,w}}{2\theta_{i,j}^u \Delta y_j + \frac{1}{2}\theta_{i,j-1}^u \Delta y_{j-1}},
\]  
(9.32)

\[
S_{i,jR}^{ib,vu} = \theta_{i,j}^u \Delta y_j \left[ -\frac{\left( \theta_{i,j}^u - \theta_{i+1,j}^u \right) u_{i+1,j}^{ib}}{V_{i+1,j}/\Delta y_j} \right].
\]  
(9.33)

The integration areas, needed for completing the shear stress fluxes discretization represents in the case of a normal fluid cell only the fluid part of the faces, for the purpose of local conservation of the fluxes. So

\[
\Delta x_{i,j}^{n,w} = \frac{1}{2}\theta_{i,j}^v \Delta x_i, \quad \Delta x_{i,j}^{n,e} = \frac{1}{2}\theta_{i,j}^v \Delta x_i, \\
\Delta x_{i,j-1}^{n,w} = \frac{1}{2}\theta_{i,j-1}^v \Delta x_i, \quad \Delta x_{i,j-1}^{n,e} = \frac{1}{2}\theta_{i,j-1}^v \Delta x_i.
\]  
(9.34)

9.2.2 v-momentum

For a control volume \(\Omega_{i,j}^v\), the corresponding discretization of viscous fluxes will be derived next.

\[
\int_{\Gamma_{i,j}^v} \nabla v \cdot \mathbf{n} \, dS = \int_{\Gamma_{i,j}^v} \frac{\partial v}{\partial x} \mathbf{e}_x \cdot \mathbf{n} \, dS + \int_{\Gamma_{i,j}^v} \frac{\partial v}{\partial y} \mathbf{e}_y \cdot \mathbf{n} \, dS.
\]  
(9.35)
The normal stress fluxes are discretized as

\[
\int_{\Gamma_{i,j}} \frac{\partial}{\partial y} e_y \cdot n \, dS \approx \theta_{i,j}^v \Delta x_i \left( \frac{\partial v}{\partial y}_{i,j+1} - \frac{\partial v}{\partial y}_{i,j} \right) \]

\[
\approx \theta_{i,j}^v \Delta x_i \left[ \frac{\theta_{i,j+1} v_{i,j+1} - \theta_{i,j} v_{i,j}}{V_{i,j+1}/\Delta x_i} - \frac{\theta_{i,j} v_{i,j} - \theta_{i,j-1} v_{i,j-1}}{V_{i,j}/\Delta x_i} \right].
\] (9.36)

And the discretization of the shear stress fluxes are stated as

\[
\int_{\Gamma_{i,j}} \frac{\partial}{\partial x} e_x \cdot n \, dS = \int_{\Gamma_{i,j}^{e,n} \cup \Gamma_{i,j+1}^{e,n}} \frac{\partial v}{\partial x} \, dy - \int_{\Gamma_{i,j}^{w,n} \cup \Gamma_{i,j+1}^{w,n}} \frac{\partial v}{\partial x} \, dy
\]

\[
\approx \left( \Delta y_{i,j}^{e,n} + \Delta y_{i,j+1}^{e,s} \right) \frac{\partial v}{\partial x}_{i,j} - \left( \Delta y_{i,j}^{w,n} + \Delta y_{i,j+1}^{w,s} \right) \frac{\partial v}{\partial x}_{i-1,j}
\]

\[
= \left( \Delta y_{i,j}^{e,n} + \Delta y_{i,j+1}^{e,s} \right) \left[ \frac{v_{i+1,j} - v_{i,j}}{\frac{1}{2} \theta_{i+1,j}^v \Delta x_{i+1} + \frac{1}{2} \theta_{i,j}^v \Delta x_i} \right]
\]

\[
- \left( \Delta y_{i,j}^{w,n} + \Delta y_{i,j+1}^{w,s} \right) \left[ \frac{v_{i,j} - v_{i-1,j}}{\frac{1}{2} \theta_{i,j}^v \Delta x_i + \frac{1}{2} \theta_{i-1,j}^v \Delta x_{i-1}} \right].
\] (9.37)

From these formulae the coefficients needed for the 5-points scheme can be derived. Just as in the case for the control volume \(\Omega_{i,j}^v\), the same concept of half control volumes are used for \(\Omega_{i,j}^v\). We begin with the formulation of the components of the lower half of control volume.

**Lower half of control volume**

\[
K_{WL}^v(i,j) = \frac{\Delta y_{i,j}^{w,n}}{\frac{1}{2} \theta_{i,j}^v \Delta x_i + \frac{1}{2} \theta_{i-1,j}^v \Delta x_{i-1}},
\] (9.38)

\[
K_{EL}^v(i,j) = \frac{\Delta y_{i,j}^{e,n}}{\frac{1}{2} \theta_{i+1,j}^v \Delta x_{i+1} + \frac{1}{2} \theta_{i,j}^v \Delta x_i},
\] (9.39)

\[
K_{RL}^v(i,j) = -\theta_{i,j}^v \Delta x_i \left[ \frac{\theta_{i,j-1}^v v_{i,j-1}}{V_{i,j}/\Delta x_i} - \frac{\theta_{i,j}^v \Delta y_{i,j}^{e,n}}{\frac{1}{2} \theta_{i,j}^v \Delta x_i + \frac{1}{2} \theta_{i-1,j}^v \Delta x_{i-1}} - \frac{\theta_{i,j}^v \Delta y_{i,j}^{e,n}}{\frac{1}{2} \theta_{i,j}^v \Delta x_i + \frac{1}{2} \theta_{i+1,j}^v \Delta x_{i+1}} \right].
\] (9.40)

\[
K_{NL}^v(i,j) = 0,
\] (9.41)


\[ K_{SL}^v(i,j) = \theta_{i,j}^v \Delta x_i \left( \frac{\theta_{i,j-1}^v}{V_{i,j}/\Delta x_i} \right), \quad (9.42) \]

\[ S_{ib,v}^L(i,j) = \theta_{i,j}^v \Delta x_i \left( -\frac{(\theta_{i,j-1}^v - \theta_{i,j}^v) v_{ib}^{i,j}}{V_{i,j}/\Delta x_i} \right). \quad (9.43) \]

Upper half of control volume

Which leaves us with the determination of coefficients coming from the upper half of control volume \( \Omega_{i,j} \)

\[ K_{WU}^v(i,j) = \frac{\Delta y_{i,j+1}^{w,s}}{2 \theta_{i,j}^v \Delta x_i + \frac{1}{2} \theta_{i-1,j}^v \Delta x_{i-1}}, \quad (9.44) \]

\[ K_{EU}^v(i,j) = \frac{\Delta y_{i,j+1}^{e,n}}{2 \theta_{i+1,j}^v \Delta x_{i+1} + \frac{1}{2} \theta_{i,j}^v \Delta x_i}, \quad (9.45) \]

\[ K_{PU}^v(i,j) = -\theta_{i,j}^v \left( \frac{\theta_{i+1,j}^v}{V_{i,j+1}/\Delta x_i} - \frac{\Delta y_{i,j+1}^{w,s}}{2 \theta_{i,j}^v \Delta x_i + \frac{1}{2} \theta_{i-1,j}^v \Delta x_{i-1}} \right), \]

\[ -\frac{1}{2} \theta_{i-1,j}^v \Delta x_{i-1} + \frac{1}{2} \theta_{i,j}^v \Delta x_i, \quad (9.46) \]

\[ K_{NU}^v(i,j) = \theta_{i,j+1}^v \Delta x_i \left( \frac{\theta_{i,j}^v}{V_{i,j+1}/\Delta x_i} \right), \quad (9.47) \]

\[ K_{SU}^v(i,j) = 0, \quad (9.48) \]

\[ S_{ib,v}^{ib,v} = \theta_{i,j}^v \Delta x_i \left( -\frac{(\theta_{i-1,j}^v - \theta_{i,j+1}^v) v_{ib}^{i,j+1}}{V_{i,j+1}/\Delta x_i} \right). \quad (9.49) \]

For completing the shear stress fluxes discretization, we need to derive the integration areas. These represents in the case of a normal fluid cell only the fluid part of the faces, due to conservation of the fluxes. So

\[ \Delta y_{i,j}^{e,n} = \frac{1}{2} \theta_{i,j}^u \Delta y_j, \quad \Delta y_{i,j+1}^{e,n} = \frac{1}{2} \theta_{i,j}^u \Delta y_j, \]

\[ \Delta y_{i-1,j}^{w,n} = \frac{1}{2} \theta_{i-1,j}^u \Delta y_j, \quad \Delta y_{i,j+1}^{w,n} = \frac{1}{2} \theta_{i,j}^u \Delta y_j. \quad (9.50) \]

In the next sections, for 6 types of cut-cells the components of the 5-points structure for the viscous fluxes will be derived. Special interest is paid at the handling of the shear stress fluxes, because they depend on the type of cut-cell. We start with the south trapezoidal cut-cell. Further the south-east triangular cut-cell will be treated. And finally the south-east pentagonal cut-cell will be discussed. For the integration areas in the neighbourhood
of the immersed boundary, \(9.50\) and \(9.50\) cannot be used. Therefore they have to be defined, which will be done §9.6.

### 9.3 South trapezoidal cut-cell

#### u-momentum

The discrete shear stress fluxes for Fig. 9.2(a) are

\[
\int_{\Gamma_{i,j}} \frac{\partial u}{\partial y} \mathbf{e}_y \cdot n \, dS \approx \left( \Delta x_{i,j}^{s,e} + \Delta x_{i+1,j}^{s,w} \right) \left\{ \frac{u(x_i+y_{i,j}^{ib}) - u_{i,j}}{\Delta x_{i,j}^{s,e} + \Delta x_{i+1,j}^{s,w}} \right\} \left\{ \frac{\theta_{i,j}^{u} \Delta y_j}{2} \right\} \\
- \left( \Delta x_{i,j}^{s,e} + \Delta x_{i+1,j}^{s,w} \right) \left\{ \frac{u_{i,j} - u_{i,j-1}}{\Delta y_j} \right\} \left\{ \frac{\theta_{i,j}^{u} + \theta_{i,j-1}^{u} \Delta y_j}{2} \right\}.
\]

The normal stress remains for all cut-cells the same and can be found in \((7.29)\).

#### Left half of control volume

From the previous equations we can derive the components of the 5-points structure for the viscous fluxes corresponding with the left half of control volume \(\Omega_{i,j}^{u}\),

\[\text{Figure 9.2: South trapezoidal cut-cell.}\]
\[ K_{P_L}(i,j) = -\theta_{i,j}^u \Delta y_j V_{i,j}/\Delta y_j - \frac{\Delta x_{i,j}^{ib,e}}{2\theta_{i,j}^u \Delta y_j} - \frac{\Delta x_{i,j}^{s,e}}{2\theta_{i,j}^u \Delta y_j} + \frac{\theta_{i,j}^u}{4\theta_{i,j}^u \Delta y_j} \Delta x_{i,j}^{s,e} \] (9.52)

\[ K_{N_L}(i,j) = 0, \] (9.53)

\[ S_{i,j}^{ib,v} = \theta_{i,j}^u \Delta y_j \left[ -\frac{\theta_{i,j}^u - \theta_{i+1,j}^u}{V_{i+1,j}/\Delta y_j} u_{i+1,j}^{ib} - \frac{\Delta x_{i,j}^{ib,w}}{2\theta_{i,j}^u \Delta y_j} u(x_i, y_{i,j}) \right]. \] (9.54)

The remaining components \( K_{W_L}(i,j), K_{E_L}(i,j) \) and \( K_{S_L}(i,j) \) are given by (9.22), (9.23) and (9.26) respectively.

**Right half of control volume**

From the equations for shear stress we can derive the components of the 5-points structure for the viscous fluxes corresponding with the right half of control volume \( \Omega_{i,j}^u \),

\[ K_{P_R}(i,j) = -\theta_{i,j}^u \Delta y_j V_{i,j}/\Delta y_j - \frac{\Delta x_{i+1,j}^{ib,e}}{2\theta_{i,j}^u \Delta y_j} - \frac{\Delta x_{i+1,j}^{s,e}}{2\theta_{i,j}^u \Delta y_j} + \frac{\theta_{i,j}^u}{4\theta_{i,j}^u \Delta y_j} \Delta x_{i+1,j}^{s,e} \] (9.55)

\[ K_{N_R}(i,j) = 0, \] (9.56)

\[ S_{i,j}^{ib,v} = \theta_{i,j}^u \Delta y_j \left[ \frac{\theta_{i,j}^u - \theta_{i+1,j}^u}{V_{i+1,j}/\Delta y_j} u_{i+1,j}^{ib} + \frac{\Delta x_{i+1,j}^{ib,w}}{2\theta_{i,j}^u \Delta y_j} u(x_i, y_{i,j}) \right]. \] (9.57)

The untreated coefficients \( K_{W_R}(i,j), K_{E_R}(i,j) \) and \( K_{S_R}(i,j) \) are given by (9.28), (9.29) and (9.32) respectively.

**9.3.2 v-momentum**

The discretization of the shear stress fluxes for Fig. 9.2(b) is stated as

\[
\int_{\Gamma_{v_{i,j}}} \frac{\partial v}{\partial x} e_x \cdot n dS \cong \left( \Delta y_{i,j}^{\nu,v} + \Delta y_{i,j+1}^{s,v} \right) \frac{\partial v}{\partial x}_{i,j} \bigg|_{i,j} - \left( \Delta y_{i,j}^{w,v} + \Delta y_{i,j+1}^{w,v} \right) \frac{\partial v}{\partial x}_{i-1,j} \bigg|_{i-1,j}
\]

\[
= \left( \Delta y_{i,j}^{\nu,v} + \Delta y_{i,j+1}^{s,v} \right) \left[ \frac{v_{i-1,j} - v_{i,j}}{\frac{1}{2} \theta_{i+1,j}^v \Delta x_{i+1} + \frac{1}{2} \theta_{i,j}^v \Delta x_i} \right] - \left( \Delta y_{i,j}^{w,v} + \Delta y_{i,j+1}^{w,v} \right) \left[ \frac{v_{i,j} - v_{i-1,j}}{\frac{1}{2} \theta_{i,j}^v \Delta x_i + \frac{1}{2} \theta_{i-1,j}^v \Delta x_{i-1}} \right]. \] (9.58)

The discretization for the normal stress is defined in (7.30).
9.4 South-east triangular cut-cell

Upper half of control volume

From these formulae the coefficients needed for the 5-points scheme can be derived. Most of the coefficients are the same as formulated in §9.2.2, so \( K_{N_U}^v(i,j) \), \( K_{E_U}^v(i,j) \), \( K_{P_U}^v(i,j) \) and \( K_{S_U}^v(i,j) \) are given by (9.44), (9.45), (9.46) and (9.48) respectively.

\[
K_{N_U}^v(i,j) = 0, \tag{9.59}
\]

\[
K_{P_U}^v(i,j) = -\frac{\theta_{i,j}^u}{2}\Delta_1 + \frac{\theta_{i-1,j}^u}{2}\Delta_2 - \frac{\theta_{i,j+1}^u}{2}\Delta_3, \tag{9.60}
\]

\[
S_{i,j_U}^{ib,v} = \Delta x_i \left[ \frac{v_{i+1,j}^{ib}}{V_{i,j+1}/\Delta x_i} \right]. \tag{9.61}
\]

9.4 South-east triangular cut-cell

![Figure 9.3: South-east triangular cut-cell.](image)

9.4.1 u-momentum

The discrete shear stress fluxes for Fig. 9.3(a) are

\[
\int_{\Gamma_{i,j}^u} \frac{\partial u}{\partial y} e_y \cdot \mathbf{n} \, dS \approx \left( \Delta x_{i,j}^{ib,c} + \Delta x_{i+1,j}^{n,w} \right) \frac{\partial u}{\partial y}_{i,j} - \left( \Delta x_{i,j}^{s,e} + \Delta x_{i+1,j}^{s,w} \right) \frac{\partial u}{\partial y}_{i,j-1}
\]

\[
= \left( \Delta x_{i,j}^{ib,c} + \Delta x_{i+1,j}^{n,w} \right) \left[ \frac{u(x_i, y_{i,j}^{ib}) - u_{i,j}}{\frac{1}{2}\theta_{i,j}^u \Delta y_j} \right]
\]

\[
- \left( \Delta x_{i,j}^{s,e} + \Delta x_{i+1,j}^{s,w} \right) \left[ \frac{u_{i,j} - u_{i,j-1}}{\frac{1}{2}\theta_{i,j}^u \Delta y_j + \frac{1}{2}\theta_{i,j-1}^u \Delta y_{j-1}} \right]. \tag{9.62}
\]
For completing the viscous fluxes, the normal stress is also needed and is formulated in (7.29).

**Left half of control volume**

From the previous equations we can derive the components of the 5-points structure for the viscous fluxes, and are formulated as

\[ K_{W_L}^{u}(i,j) = 0, \]  
\[ K_{P_L}^{u}(i,j) = -\theta_{i,j}^{u}\Delta y_j \frac{\theta_{i-1,j}^{u}}{V_{i,j}/\Delta y_j} - \frac{\Delta x_{i,j}^{b,e}}{2\theta_{i,j}^{u}} \Delta y_j - \frac{\Delta x_{i,j}^{s,e}}{2\theta_{i,j-1}^{u}} \Delta y_{j-1}, \]  
\[ K_{N_L}^{u}(i,j) = 0, \]  
\[ S_{i,j_L}^{ib,v} = \theta_{i,j}^{u}\Delta y_j \left[ + \frac{\theta_{i,j}^{u}V_{i,j}/\Delta y_j}{2\theta_{i,j}^{u}} \Delta y_j \right] + \frac{\Delta x_{i,j}^{b,e}}{2\theta_{i,j}^{u}} \Delta y_j u(x_i, y_{i,j}). \]  

The remaining coefficients \( K_{E_L}^{u}(i,j) \) and \( K_{S_L}^{u}(i,j) \) are given by (9.23) and (9.26) respectively.

**9.4.2 v-momentum**

The discretization of the shear stress fluxes for Fig. 9.3(b) is stated as

\[ \int_{\Gamma_{i,j}} \frac{\partial v}{\partial x} e_x \cdot n dS \approx \left( \Delta y_{i,j}^{e,n} + \Delta y_{i,j+1}^{e,s} \right) \frac{\partial v}{\partial x} \bigg|_{i,j} - \left( \Delta y_{i,j}^{w,n} + \Delta y_{i,j+1}^{b,s} \right) \frac{\partial v}{\partial x} \bigg|_{i-1,j} \]

\[ = \left( \Delta y_{i,j}^{e,n} + \Delta y_{i,j+1}^{e,s} \right) \left[ \frac{v_{i+1,j} - v_{i,j}}{2\theta_{i+1,j}^{v}} \Delta x_{i+1} + \frac{1}{2\theta_{i,j}^{v}} \Delta x_{i} \right] \]

\[ - \left( \Delta y_{i,j}^{w,n} + \Delta y_{i,j+1}^{b,s} \right) \left[ \frac{v_{i,j} - v(x_{i-1,j}^{ib}, y_{j})}{2\theta_{i,j}^{v}} \Delta x_{i} \right]. \]

The normal stress, needed for completing the viscous fluxes, is formulated in (7.30).

**Upper half of control volume**

From these formulae the coefficients needed for the 5-points scheme can be derived and are given as
\[ K^u_{W_U}(i, j) = 0, \]  
\[ K^u_{P_U}(i, j) = -\theta^v_{i,j} \Delta x_i \frac{\theta^v_{i,j+1}}{V_{i,j+1}} \Delta y_{i,j+1}^s - \frac{\Delta y_{i,j+1}^s}{2 \theta^v_{i,j+1} \Delta x_i} - \frac{\Delta y_{i,j+1}^s}{2 \theta^v_{i,j} \Delta x_i}, \]  
\[ K^u_{N_U}(i, j) = 0, \]  
\[ S^i_{i,j} = \theta^v_{i,j} \Delta x_i \left[ \frac{\theta^v_{i,j+1}^{ib}}{V_{i,j+1}} \Delta x_i \right] + \frac{\theta^v_{i,j}^{ib}}{2 \theta^v_{i,j} \Delta x_i} \cdot v(x_{i-1}^{ib}, y_j). \]

The components which are not treated yet are \( K^v_{E_U}(i, j) \) and \( K^v_{S_U}(i, j) \) and these are given by (9.45) and (9.48) respectively.

## 9.5 South-east pentagonal cut-cell

(a) Control volume \( \Omega^u_{i,j} \).  
(b) Control volume \( \Omega^u_{i,j} \).  
(c) Control volume \( \Omega^v_{i,j} \).  
(d) Control volume \( \Omega^v_{i,j} \).

Figure 9.4: Control volumes for a south-east pentagonal cut-cell.

### 9.5.1 u-momentum

The discrete shear stress fluxes for Fig. 9.4(a)
\[ \int_{\Gamma_{i,j}} \frac{\partial u}{\partial y} e_y \cdot ndS \approx (\Delta x_{i,j}^{n,e} + \Delta x_{i+1,j}^{n,w}) \frac{\partial u}{\partial y} |_{i,j} - (\Delta x_{i,j}^{s,e} + \Delta x_{i+1,j}^{s,w}) \frac{\partial u}{\partial y} |_{i,j-1} \]

\[ = \left( \Delta x_{i,j}^{n,e} + \Delta x_{i+1,j}^{n,w} \right) \frac{u_{i,j+1} - u_{i,j}}{2 \theta_{i,j+1} \Delta y_j + \frac{1}{2} \theta_{i,j} \Delta y_j} \]

\[ - \left( \Delta x_{i,j}^{s,e} + \Delta x_{i+1,j}^{s,w} \right) \frac{u_{i,j} - u_{i,j-1}}{2 \theta_{i,j} \Delta y_j + \frac{1}{2} \theta_{i,j-1} \Delta y_j} , \quad (9.72) \]

and for Fig 9.4(b)

\[ \int_{\Gamma_{i,j}} \frac{\partial u}{\partial y} e_y \cdot ndS \approx (\Delta x_{i,j}^{i,b,e} + \Delta x_{i+1,j}^{n,w}) \frac{\partial u}{\partial y} |_{i,j} - (\Delta x_{i,j}^{s,e} + \Delta x_{i+1,j}^{s,w}) \frac{\partial u}{\partial y} |_{i,j-1} \]

\[ = \left( \Delta x_{i,j}^{i,b,e} + \Delta x_{i+1,j}^{n,w} \right) \frac{u(x_{i,j}, y_{i,j+1}) - u_{i,j}}{2 \theta_{i,j} \Delta y_j} \]

\[ - \left( \Delta x_{i,j}^{s,e} + \Delta x_{i+1,j}^{s,w} \right) \frac{u_{i,j} - u_{i,j-1}}{2 \theta_{i,j} \Delta y_j + \frac{1}{2} \theta_{i,j-1} \Delta y_j} , \quad (9.73) \]

where

\[ \Delta x_{i+1,j}^{n,w} = \Delta x_{i+1,j}^{s,w} + \Delta x_{i+1,j}^{e,w} . \quad (9.74) \]

The normal stress still holds for this situation, because the formulae (7.29) is derived in such a way that it is valid for every type of cut-cell.

### Left half of control volume

From the equations for the shear stress we can derive the components of the 5-points structure for the viscous fluxes corresponding with the left half of control volume \( \Omega_{i,j}^u \), see Fig. 9.4(a). It turns out that all of these components are already computed in section 9.2.1. Thus \( \kappa_{W_L}(i,j), \kappa_{E_L}(i,j), \kappa_{P_L}(i,j), \kappa_{N_L}(i,j) \) and \( \kappa_{S_L}(i,j) \) are given by (9.22), (9.23), (9.24), (9.25) and (9.26), respectively.

### Right half of control volume

From the equations for shear stress we can derive the components of the 5-points structure for the viscous fluxes corresponding with the right half of control volume \( \Omega_{i,j}^u \), see Fig. 9.4(b). It turns out that all of these components have already been computed in §9.2.1.
### 9.5. SOUTH-EAST PENTAGONAL CUT-CELL

\[ K_{PR}^u(i, j) = -\theta_{i,j}^u \Delta y_j \left( \frac{\theta_{i,j}^u}{V_{i+1,j}^u} \Delta \bar{y}_{i+1,j}^n + \frac{\theta_{i+1,j}^u}{2 \theta_{i,j}^u} \Delta \bar{y}_{i+1,j}^s \right), \quad (9.75) \]

\[ K_{NR}^u(i, j) = 0, \quad (9.76) \]

\[ S_{i,j}^{ib,u} = \theta_{i,j}^u \Delta y_j \left( \frac{\left( \theta_{i,j}^u - \frac{\theta_{i+1,j}^u}{2} \right) V_{i+1,j}^u}{2 \theta_{i,j}^u} \Delta \bar{y}_{i+1,j}^n + \frac{\Delta \bar{y}_{i+1,j}^s}{2 \theta_{i,j}^u} \Delta \bar{y}_{i+1,j}^s \right), \quad (9.77) \]

The remaining components \( K_{WR}^u(i, j) \), \( K_{ER}^u(i, j) \) and \( K_{SR}^u(i, j) \) are given by (9.28), (9.29) and (9.32), respectively.

#### 9.5.2 v-momentum

The discretization of the shear stress fluxes for Fig 9.4(c) is

\[ \int_{\Gamma_{i,j}^v} \frac{\partial v}{\partial x} e_x \cdot ndS \approx \left( \Delta y_{i,j}^e,n + \Delta y_{i,j+1}^e,s \right) \frac{\partial v}{\partial x} \bigg|_{i,j} - \left( \Delta \bar{y}_{i,j}^n + \Delta \bar{y}_{i,j+1}^{ib,s} \right) \frac{\partial v}{\partial x} \bigg|_{i-1,j} \]

\[ = \left( \Delta y_{i,j}^e,n + \Delta y_{i,j+1}^e,s \right) \left[ \frac{v_{i+1,j} - v_{i,j}}{2 \theta_{i+1,j}^v} \Delta x_{i+1} + \frac{v_{i,j} - v_{i,j+1}^{ib,b}}{2 \theta_{i,j}^v} \Delta x_i \right] \]

\[ - \left( \Delta \bar{y}_{i,j}^n + \Delta \bar{y}_{i,j+1}^{ib,s} \right) \left[ \frac{v_{i,j} - v(x_{i,j}^{ib,b}, y_j)}{2 \theta_{i,j}^v} \Delta x_i \right], \quad (9.78) \]

where

\[ \Delta \bar{y}_{i-1,j}^n = \Delta y_{i-1,j}^{ib,n} + \Delta y_{i,j}^n. \quad (9.79) \]

The discretization of the shear stress fluxes for Fig. 9.4(d) is

\[ \int_{\Gamma_{i,j}^v} \frac{\partial v}{\partial x} e_x \cdot ndS \approx \left( \Delta y_{i,j}^e,n + \Delta y_{i,j+1}^e,s \right) \frac{\partial v}{\partial x} \bigg|_{i,j} - \left( \Delta y_{i,j}^w,n + \Delta y_{i,j+1}^{w,s} \right) \frac{\partial v}{\partial x} \bigg|_{i-1,j} \]

\[ = \left( \Delta y_{i,j}^e,n + \Delta y_{i,j+1}^e,s \right) \left[ \frac{v_{i+1,j} - v_{i,j}}{2 \theta_{i+1,j}^v} \Delta x_{i+1} + \frac{v_{i,j} - v_{i,j+1}^{w,s}}{2 \theta_{i,j}^v} \Delta x_i \right] \]

\[ - \left( \Delta y_{i,j}^w,n + \Delta y_{i,j+1}^{w,s} \right) \left[ \frac{v_{i,j} - v_{i-1,j}}{2 \theta_{i,j}^v} \Delta x_i + \frac{v_{i,j} - v_{i,j+1}^{w,s}}{2 \theta_{i,j}^v} \Delta x_{i-1} \right]. \quad (9.80) \]

The normal stress are also needed for deriving components for the 5-points structure and these are formulated in (7.30).
Lower half of control volume

From these formulae the coefficients needed for the 5-points scheme can be derived. The components corresponding with the lower half of control volume $\Omega_{i,j}^v$, see Fig. 9.4(c).

\[
K_{W_L}^v(i,j) = 0, \quad (9.81)
\]

\[
K_{P_L}^v(i,j) = -\theta_{i,j}^v \Delta x_i \frac{\theta_{i,j}^v \Delta y_{i,j}}{V_{i,j}/\Delta x_i} - \frac{1}{2} \theta_{i,j}^v \Delta x_i - \frac{1}{2} \theta_{i,j}^v \Delta x_i + \frac{1}{2} \theta_{i,j}^v \Delta x_i, \quad (9.82)
\]

\[
S_{ib,v}^{i,j} = \theta_{i,j}^v \Delta x_i \left[ -\theta_{i,j-1}^v v_{i,j}^b \frac{V_{i,j}/\Delta x_i}{V_{i,j}/\Delta x_i} + \frac{\Delta y_{i,j}^{ib,s}}{2} \right], \quad (9.83)
\]

The remaining components $K_{E_L}^v(i,j)$, $K_{N_L}^v(i,j)$ and $K_{S_L}^v(i,j)$ are given by (9.39), (9.41) and (9.42), respectively.

Upper half of control volume

From these formulae the coefficients needed for the 5-points scheme can be derived. The components corresponding with the upper half of control volume $\Omega_{i,j}^v$, see Fig. 9.4(d).

It turns out that all of these components have already been computed in §9.2.2. Thus $K_{W_U}^v(i,j)$, $K_{P_U}^v(i,j)$, $K_{E_U}^v(i,j)$, $K_{N_U}^v(i,j)$ and $K_{S_U}^v(i,j)$ are given by (9.22), (9.23), (9.24), (9.25) and (9.26), respectively.

9.6 Integration areas needed for the shear stress

In this section, the integration areas needed for the shear stress will be determined. Before doing so, the quadrature of the shear stress used for discretizing the stress forces (9.92), (9.93) acting on the immersed boundary is needed.

9.6.1 Stress force

In this subsection the integration areas will be derived using discrete conservation of total momentum and computation of viscous terms.

The total momentum is given by

\[
P(t) = \int_{\Omega^t} \mathbf{v} d\mathbf{V}. \quad (9.84)
\]

Discretizing this term with the trapezoidal rule results

\[
P^h(t) = \mathbf{1}^T \mathbf{MU} + \mathbf{1}^T \mathbf{M}^{ib} \mathbf{U}^{ib}, \quad (9.85)
\]
where $\mathbf{1}$ is the constant vector. To formulate the conservation equation for $P^h(t)$, we start with differentiating above equation to time

$$\frac{dP^h}{dt} = \frac{d}{dt} \left( \mathbf{1}^T \mathcal{M} \mathbf{U} + \mathbf{1}^T \mathcal{M}^b U^b \right)$$

$$= \mathbf{1}^T \frac{d}{dt} (\mathcal{M} \mathbf{U}) + \mathbf{1}^T \frac{d}{dt} (\mathcal{M}^b U^b)$$

$$= \mathbf{1}^T \frac{d}{dt} (\mathcal{M} \mathbf{U}). \tag{9.86}$$

where Cheny et al. have assumed that the contribution of the boundary conditions, $\mathcal{M}^b U^b$, are steady. Rewriting the equation using the semi-discrete scheme (7.3) leads to

$$\frac{dP^h}{dt} = \mathbf{1}^T \left( -\mathcal{C}[\mathbf{U}]\mathbf{U} - \mathcal{G} P + \nu \mathcal{K} \mathbf{U} - \mathbf{S}^{ib,c} + \nu \mathbf{S}^{ib,v} \right)$$

$$= - \left[ \mathbf{1}^T \mathcal{C}[\mathbf{U}] \mathbf{U} + \mathbf{1}^T \mathbf{S}^{ib,c} \right] - \left[ \mathbf{1}^T \mathcal{G} P - \mathbf{1}^T \nu \left( \mathcal{K} \mathbf{U} + \mathbf{S}^{ib,v} \right) \right]. \tag{9.87}$$

This expression is the semi-discrete version of

$$\frac{dP}{dt} = - \int_{\Gamma^b} \mathbf{v} \cdot \mathbf{dS} - \mathbf{F}, \tag{9.88}$$

where this term is obtained by integration by parts of the volume integral and $\mathbf{F} = (F_x, F_y)$ represents the stress force acting on the immersed boundary such that

$$F_x = \int_{\Gamma^b} \left[ p - \nu \frac{\partial u}{\partial x} \right] \mathbf{e}_x \cdot \mathbf{n} dS - \int_{\Gamma^b} \nu \frac{\partial u}{\partial y} \mathbf{e}_y \cdot \mathbf{n} dS, \tag{9.89}$$

$$F_y = - \int_{\Gamma^b} \nu \frac{\partial v}{\partial x} \mathbf{e}_x \cdot \mathbf{n} dS + \int_{\Gamma^b} \left[ p - \nu \frac{\partial v}{\partial y} \right] \mathbf{e}_y \cdot \mathbf{n} dS. \tag{9.90}$$

The terms in the right-hand side of (9.87) correspond to the summation of the convective, pressure and viscous fluxes from all control volumes. Since the property of local conservatity of the fluxes holds at fluid faces, all terms cancel out except those appearing at solid boundary faces of the cut-cells. These remaining terms should correspond to the forces that act on the immersed boundary. The LS-STAG method is momentum conserving if the non-zero terms in the sum $\left[ \mathbf{1}^T \mathcal{G} P - \mathbf{1}^T \nu \left( \mathcal{K} \mathbf{U} + \mathbf{S}^{ib,v} \right) \right]$ correspond to the discretization of the stress forces (9.89),(9.90).

$$\left[ \mathbf{1}^T \mathcal{G} P - \mathbf{1}^T \nu \left( \mathcal{K} \mathbf{U} + \mathbf{S}^{ib,v} \right) \right] = \left[ \mathbf{1}^T \mathcal{G} P - \mathbf{1}^T \nu \left( \mathcal{K} \mathbf{U} + \mathbf{S}^{ib,v} \right) \right]_{x} = \tilde{F}^h_x$$

$$\left[ \mathbf{1}^T \mathcal{G} P - \mathbf{1}^T \nu \left( \mathcal{K} \mathbf{U} + \mathbf{S}^{ib,v} \right) \right]_{y} = \tilde{F}^h_y \tag{9.91}$$
In other words, the LS-STAG method is momentum conserving if $\tilde{F}_h^x = F_h^x$ and $\tilde{F}_h^y = F_h^y$.

The discretization of the stress forces is obtained by approximating the surface integrals in (9.89) and (9.90) respectively as

\[
F_h^x = \sum_{\text{Cut-cells } \Omega_{i,j}} \left[ n_x \Delta S_{i,j} \left( p_{i,j} - \nu \left| \frac{\partial u}{\partial x} \right|_{i,j} \right) - \nu \text{Quad}_{i,j}^b \left( \frac{\partial u}{\partial y} e_y \cdot n \right) \right], \quad (9.92)
\]

\[
F_h^y = \sum_{\text{Cut-cells } \Omega_{i,j}} \left[ -\nu \text{Quad}_{i,j}^b \left( \frac{\partial v}{\partial x} e_x \cdot n \right) + [n_y \Delta S_{i,j} \left( p_{i,j} - \nu \left| \frac{\partial v}{\partial y} \right|_{i,j} \right) \right], \quad (9.93)
\]

where Quad$^b_{i,j}()$ represents the quadrature of the shear stresses.

In these equations, the quadrature of the pressure and normal stress term has been performed by observing that these terms are constant in the cut-cells and when using the midpoint rule leads to the fact that the same formula is valid for all types of cut-cells. In contrast, the quadrature of the shear stresses has to be adapted to each type of cut-cells. This quadrature, based on the location of the shear stresses and the trapezoidal rule will be described in §9.6.2-9.6.4.

The stress part of the global momentum equation (9.87), whose contribution in the $x$ direction is given by

\[
\tilde{F}_x^h = \sum_{\text{Control volumes } \Omega_{i,j}^u} \int_{\Gamma_{i,j}^u} \left[ p - \nu \frac{\partial u}{\partial x} \right] e_x \cdot n dS - \int_{\Gamma_{i,j}^u} \nu \frac{\partial u}{\partial y} e_y \cdot n dS
\]

\[
= \sum_{\text{CVs } \Omega_{i,j}^u} \theta_{i,j}^u \Delta y_j (p_{i+1,j} - p_{i,j}) - \nu \int_{\Gamma_{i,j}^u} \frac{\partial u}{\partial x} e_x \cdot n + \frac{\partial u}{\partial y} e_y \cdot n dS
\]

\[
= \sum_{\text{CVs } \Omega_{i,j}^u} \theta_{i,j}^u \Delta y_j (p_{i+1,j} - p_{i,j}) - \nu \theta_{i,j}^u \Delta y_j \left( \frac{\partial u}{\partial x} \bigg|_{i+1,j} - \frac{\partial u}{\partial x} \bigg|_{i,j} \right)
\]

\[
- \nu \int_{\Gamma_{i,j}^u} \frac{\partial u}{\partial y} e_y \cdot n dS, \quad (9.94)
\]

should correspond to the drag force $F_x^h$ in (9.92). The easiest part to inspect is the contribution of the normal stresses, since a unique formula (namely, (7.13) and (7.29)) is valid for these terms in all computational cells

\[
\tilde{F}_{x,\text{normal}} = \sum_{\text{Control volumes } \Omega_{i,j}^u} \theta_{i,j}^u \Delta y_j \left( p_{i+1,j} - p_{i,j} - \nu \left[ \frac{\partial u}{\partial x} \bigg|_{i+1,j} - \frac{\partial u}{\partial x} \bigg|_{i,j} \right] \right). \quad (9.95)
\]
Using re-indexation, the sum is rewritten such that it becomes a sum over the computational cells. When examining this sum, we observe that the pressure and normal stresses cancel out in fluid cells where $\theta_{i,j}^u = \theta_{i-1,j}^u = 1$, and only the following terms in the cut-cells do remain

$$\tilde{F}_{x,normal}^h = \sum_{\text{Cut-cells } \Omega_{i,j}} \left( \theta_{i-1,j}^u - \theta_{i,j}^u \right) \Delta y_j \left( p_{i,j} - \nu \left. \frac{\partial u}{\partial x} \right|_{i,j} \right). \quad (9.96)$$

This sum is exactly the contribution of the normal stresses to the discrete drag force (9.92).

And now we do the same for the vertical force (9.93). The stress part of the global momentum equation (9.87), whose contribution in the $y$ direction is given by

$$\tilde{F}_y^h = \sum_{\text{Control volumes } \Omega_{i,j}^v} - \int_{\Gamma_{i,j}^v} \frac{\partial v}{\partial x} e_x \cdot n dS + \int_{\Gamma_{i,j}^v} \left( p - \nu \frac{\partial v}{\partial y} \right) e_y \cdot n dS \approx \sum_{\text{CVs } \Omega_{i,j}^v} \theta_{i,j}^v \Delta x_i \left( p_{i,j+1} - p_{i,j} \right) - \nu \left. \frac{\partial v}{\partial y} \right|_{i,j+1} - \left. \frac{\partial v}{\partial y} \right|_{i,j} \Delta x_i \left( \frac{\partial v}{\partial y} \right)_{i,j+1} - \left( \frac{\partial v}{\partial y} \right)_{i,j} \right)$$

$$\approx - \nu \int_{\Gamma_{i,j}^v} \frac{\partial v}{\partial x} e_x \cdot n dS \quad (9.97)$$

which should correspond to the lift force $F_y^h$ in (9.93). Again, the easiest part to inspect is the contribution of the normal stresses, since a unique formula (namely (7.14) and (7.30)) is valid for these terms in all computational cells

$$\tilde{F}_{y,normal}^h = \sum_{\text{Control volumes } \Omega_{i,j}^v} \theta_{i,j}^v \Delta x_i \left( p_{i,j+1} - p_{i,j} \right) - \nu \left[ \left. \frac{\partial v}{\partial y} \right|_{i,j+1} - \left. \frac{\partial v}{\partial y} \right|_{i,j} \right] \quad (9.98)$$

Again using re-indexation, the sum is rewritten such that it becomes a sum over the computational cells. When taking a closer look at this sum, we observe that the pressure and normal stresses cancel out in fluid cells such that $\theta_{i,j}^v = \theta_{i,j-1}^v = 1$, and only the following terms in the cut-cells do remain

$$\tilde{F}_{y,normal}^h = \sum_{\text{Cut-cells } \Omega_{i,j}} \left( \theta_{i,j-1}^v - \theta_{i,j}^v \right) \Delta x_i \left( p_{i,j} - \nu \left. \frac{\partial v}{\partial y} \right|_{i,j} \right). \quad (9.99)$$
This sum is exactly the contribution of the normal stresses to the discrete drag force (9.93). Summarizing what we have achieved so far, above derivation ensures that the normal stress contribution to the total momentum budget is recovered.

For the shear stress contribution, the fluxes at all fluid faces cancel out and only fluxes at the immersed boundary remains in sum (9.94) or in sum (9.97), thus

\[
\tilde{F}_{x,\text{shear}}^{h} \bigg|_{i,j} = -\nu \int_{\Gamma_{i,j}} \frac{\partial u}{\partial y} e_{y} \cdot n dS, 
\]

\[
\tilde{F}_{y,\text{shear}}^{h} \bigg|_{i,j} = -\nu \int_{\Gamma_{i,j}} \frac{\partial v}{\partial x} e_{x} \cdot n dS. 
\]

The shear stress in the the exact expression (9.92), (9.93), formulated as

\[
F_{x,\text{shear}}^{h} \bigg|_{i,j} = -\nu \text{Quad}_{i,j}^{ib} \left( \frac{\partial u}{\partial y} e_{y} \cdot n \right), 
\]

\[
F_{y,\text{shear}}^{h} \bigg|_{i,j} = -\nu \text{Quad}_{i,j}^{ib} \left( \frac{\partial v}{\partial x} e_{x} \cdot n \right), 
\]

depends on the type of cut-cell.

In the next sections, the integration areas which are aligned with the immersed boundary will be derived. For all the other integration areas (thus away from the boundary), they are just the same as the ones derived for the normal fluid cell.

### 9.6.2 South trapezoidal cut-cell

This situation is depicted in Fig. 9.2. The LS-STAG method is momentum conserving if \( \tilde{F}_{x}^{h} = F_{x}^{h} \) and \( \tilde{F}_{y}^{h} = F_{y}^{h} \) (thus (9.92) must be equal to (9.96) + (9.100) and (9.93) must be equal to (9.99) + (9.101)). In section 9.6.1 it has already been shown that \( \tilde{F}_{x,\text{normal}}^{h} = F_{x,\text{normal}}^{h} \) and \( \tilde{F}_{y,\text{normal}}^{h} = F_{y,\text{normal}}^{h} \) holds. The part that remains are the shear stresses. Before we can compare \( \tilde{F}_{x,\text{shear}}^{h} \) with \( F_{x,\text{shear}}^{h} \) and \( \tilde{F}_{y,\text{shear}}^{h} \) with \( F_{y,\text{shear}}^{h} \), the quadratures for the shear stress must be determined, because they are in (9.102) and (9.103) still undefined. The quadratures for the shear stress are formulated as

\[
\text{Quad}_{i,j}^{ib} \left( \frac{\partial u}{\partial y} e_{y} \cdot n \right) = \left[ n_{y} \Delta S_{i,j}^{ib} \left( \frac{1}{2} \frac{\partial u}{\partial y} \bigg|_{i-1,j} + \frac{1}{2} \frac{\partial u}{\partial y} \bigg|_{i,j} \right) \right],
\]

\[
\text{Quad}_{i,j}^{ib} \left( \frac{\partial v}{\partial x} e_{x} \cdot n \right) = \left[ n_{x} \Delta S_{i,j}^{ib} \left( \frac{1}{2} \frac{\partial v}{\partial x} \bigg|_{i-1,j} + \frac{1}{2} \frac{\partial v}{\partial x} \bigg|_{i,j} \right) \right].
\]
where \([n_x \Delta S_{ij}^{ib} = (\theta_{i-1,j}^u - \theta_{i,j}^u) \Delta y_j] \) and \([n_y \Delta S_{ij}^{ib} = (\theta_{i,j-1}^v - \theta_{i,j}^v) \Delta x_i] \). Thus combining (9.104) with (9.103) and (9.105) with (9.102) results in

\[ F_{x,shear}^{h,i,j} = -\nu \frac{\Delta x_i}{2} \left( \frac{\partial u}{\partial y}_{i,j} + \frac{\partial u}{\partial y}_{i,j} \right) \right), \quad (9.106) \]

\[ F_{y,shear}^{h,i,j} = -\nu \left( \theta_{i,j}^u - \theta_{i,j}^u \right) \Delta y_j \left( \frac{\partial v}{\partial x}_{i,j} + \frac{\partial v}{\partial x}_{i,j} \right) \right). \quad (9.107) \]

The shear stress in (9.100) and (9.101) can be formulated using Fig. 9.2

\[ F_{x,shear}^{h,i,j} = -\nu \left[ \Delta x_{i,j}^{ib,w} \frac{\partial u}{\partial y}_{i,j} + \Delta x_{i,j}^{ib,e} \frac{\partial u}{\partial y}_{i,j} \right] \right), \quad (9.108) \]

\[ F_{y,shear}^{h,i,j} = -\nu \left[ \Delta y_{i,j}^{e,s} \frac{\partial v}{\partial x}_{i,j} + \Delta y_{i,j}^{w,s} \frac{\partial v}{\partial x}_{i,j} \right] \right). \quad (9.109) \]

Now it is possible to compare \(F_{x,shear}^{h,i,j}\) with \(F_{x,shear}^{h,i,j}\) and \(F_{y,shear}^{h,i,j}\) with \(F_{y,shear}^{h,i,j}\). By requiring that (9.106) is equal to (9.108) and (9.107) is equal to (9.109), the integration areas in the neighbourhood of the immersed boundary can be derived. These are defined as

\[ \Delta x_{i,j}^{ib,w} = \frac{1}{2} \Delta x_i, \quad \Delta x_{i,j}^{ib,e} = \frac{1}{2} \Delta x_i, \quad (9.110) \]

and far away from the immersed boundary, the remaining integration areas are the same as defined for the 'normal' fluid cell, see (9.34), (9.50), thus

\[ \Delta x_{i,j}^{w,n} = \frac{1}{2} \theta_{i,j-1}^v \Delta x_i, \quad \Delta x_{i,j}^{w,s} = \frac{1}{2} \theta_{i,j}^v \Delta x_i, \]

\[ \Delta y_{i,j}^{w,n} = \frac{1}{2} \theta_{i,j-1}^v \Delta y_j, \quad \Delta y_{i,j}^{w,s} = \frac{1}{2} \theta_{i,j}^v \Delta y_j. \quad (9.111) \]

### 9.6.3 South-east triangular cut-cells

The geometry for this situation is shown in Fig. 9.3. Again the LS-STAG method is momentum conserving if \(\tilde{F}_{x,shear}^{h,i,j} = F_{x,shear}^{h,i,j}\) and \(\tilde{F}_{y,shear}^{h,i,j} = F_{y,shear}^{h,i,j}\) (thus (9.92) must be equal to (9.96) + (9.100) and (9.93) must be equal to (9.99) + (9.101) ). We have seen that in section 9.6.1 the relationship \(\tilde{F}_{x,normal}^{h,i,j} = F_{x,normal}^{h,i,j}\) and \(\tilde{F}_{y,normal}^{h,i,j} = F_{y,normal}^{h,i,j}\) holds. The part that remains for investigation are the shear stresses. Before we can compare \(\tilde{F}_{x,shear}^{h,i,j}\) with \(F_{x,shear}^{h,i,j}\) and \(\tilde{F}_{y,shear}^{h,i,j}\) with \(F_{y,shear}^{h,i,j}\), the quadratures for the shear stress must be determined, because they are in (9.102) and (9.103) still undefined. The quadratures for the shear stress are derived using Fig. 9.3 as
\[ \text{Quadr}_i^b \left( \frac{\partial u}{\partial y} e_y \cdot n \right) = [n_y \Delta S]_{i,j}^b \left( \frac{1}{2} \frac{\partial u}{\partial y} \bigg|_{i-1,j-1} + \frac{1}{2} \frac{\partial u}{\partial y} \bigg|_{i,j} \right), \quad (9.112) \]

\[ \text{Quadr}_i^b \left( \frac{\partial v}{\partial x} e_x \cdot n \right) = [n_x \Delta S]_{i,j}^b \left( \frac{1}{2} \frac{\partial v}{\partial x} \bigg|_{i-1,j-1} + \frac{1}{2} \frac{\partial v}{\partial x} \bigg|_{i,j} \right), \quad (9.113) \]

where \([n_x \Delta S]_{i,j}^b = -\theta_{i,j}^u \Delta y_j\) and \([n_y \Delta S]_{i,j}^b = \theta_{i,j-1}^v \Delta x_i\). Thus combining (9.112) with (9.103) and (9.113) with (9.102) results in

\[ F_{x,\text{shear}}^i \bigg|_{i,j} = -\nu \theta_{i,j}^v \Delta x_i \left( \frac{1}{2} \frac{\partial u}{\partial y} \bigg|_{i,j} + \frac{1}{2} \frac{\partial u}{\partial y} \bigg|_{i-1,j-1} \right), \quad (9.114) \]

\[ F_{y,\text{shear}}^i \bigg|_{i,j} = -\nu \left(-\theta_{i,j}^u \right) \Delta y_j \left( \frac{1}{2} \frac{\partial v}{\partial x} \bigg|_{i-1,j-1} + \frac{1}{2} \frac{\partial v}{\partial x} \bigg|_{i,j} \right). \quad (9.115) \]

The shear stress in (9.100) and (9.101) can be formulated using Fig. 9.3

\[ \tilde{F}_{x,\text{shear}}^i \bigg|_{i,j} = -\nu \left[ \Delta x_{i-1,j-1}^{x,w} \frac{\partial u}{\partial y} \bigg|_{i-1,j-1} + \Delta x_{i,j-1}^{x,e} \frac{\partial u}{\partial y} \bigg|_{i,j} \right], \quad (9.116) \]

\[ \tilde{F}_{y,\text{shear}}^i \bigg|_{i,j} = -\left(-\nu \right) \left[ \Delta y_{i,j}^{y,n} \frac{\partial v}{\partial x} \bigg|_{i,j} + \Delta y_{i-1,j}^{y,s} \frac{\partial v}{\partial x} \bigg|_{i-1,j-1} \right]. \quad (9.117) \]

At this point it is possible to compare \(\tilde{F}_{x,\text{shear}}^i\) with \(F_{x,\text{shear}}^i\) and \(\tilde{F}_{y,\text{shear}}^i\) with \(F_{y,\text{shear}}^i\). To accomplish this, we require that (9.114) is equal to (9.116) and (9.115) is equal to (9.117). From this comparison, the integration areas in the vicinity of the immersed boundary can be derived and are defined as

\[ \Delta x_{i,j}^{x,w} = \frac{1}{2} \theta_{i,j-1}^v \Delta x_i, \quad \Delta x_{i,j}^{x,e} = \frac{1}{2} \theta_{i,j-1}^v \Delta x_i, \quad \Delta y_{i-1,j}^{y,n} = \frac{1}{2} \theta_{i,j}^u \Delta y_j, \quad \Delta y_{i-1,j}^{y,s} = \frac{1}{2} \theta_{i,j}^u \Delta y_j. \quad (9.118) \]

and far away from the immersed boundary, the remaining integration areas are the same as defined for the 'normal' fluid cell, see (9.34), (9.50).

\[ \Delta x_{i,j-1}^{x,w} = \frac{1}{2} \theta_{i,j-1}^v \Delta x_i, \quad \Delta x_{i,j-1}^{x,e} = \frac{1}{2} \theta_{i,j-1}^v \Delta x_i, \quad \Delta y_{i,j}^{y,n} = \frac{1}{2} \theta_{i,j}^u \Delta y_j, \quad \Delta y_{i,j}^{y,s} = \frac{1}{2} \theta_{i,j}^u \Delta y_j. \quad (9.119) \]
9.6.4 South-east pentagonal cut-cell

A sketch of this situation can be found in Fig. 9.4. The LS-STAG method is momentum conserving if \( \tilde{F}_h^x = F_h^x \) and \( \tilde{F}_h^y = F_h^y \) (thus (9.92) must be equal to (9.96) + (9.100) and (9.93) must be equal to (9.99) + (9.101)). In section 9.6.1, we have derived that the relationship \( \tilde{F}_h^x,_{normal} = F_h^x,_{normal} \) and \( \tilde{F}_h^y,_{normal} = F_h^y,_{normal} \) holds. The remaining part for investigation are the shear stresses. Before we can compare \( \tilde{F}_h^x,_{shear} \) with \( F_h^x,_{shear} \) and \( \tilde{F}_h^y,_{shear} \) with \( F_h^y,_{shear} \), the quadratures for the shear stress must be determined, because they are still undefined in (9.102) and (9.103). Using Fig. 9.4, the quadratures for the shear stress are derived as

\[
\text{Quad}_{ib}^{i,j} \left( \frac{\partial u}{\partial y} \right) = \left[ n_y \Delta S_{ib}^{i,j} \right]_{i-1,j} \left( \frac{\partial u}{\partial y} \right)_{i-1,j}, \quad (9.120)
\]

\[
\text{Quad}_{ib}^{i,j} \left( \frac{\partial v}{\partial x} \right) = \left[ n_x \Delta S_{ib}^{i,j} \right]_{i-1,j} \left( \frac{\partial v}{\partial x} \right)_{i-1,j}, \quad (9.121)
\]

where \( [n_x \Delta S]^{i,j}_{ib} = \left( \theta_{i-1,j}^u - 1 \right) \Delta y_j \) and \( [n_y \Delta S]^{i,j}_{ib} = \left( 1 - \theta_{i,j}^v \right) \Delta x_i \).

Thus combining (9.120) with (9.103) and (9.121) with (9.102) results in

\[
F_h^x,_{shear} \bigg|_{i,j} = -\nu \left( 1 - \theta_{i,j}^v \right) \Delta x_i \left( \frac{\partial u}{\partial y} \right)_{i-1,j}, \quad (9.122)
\]

\[
F_h^y,_{shear} \bigg|_{i,j} = -\nu \left( \theta_{i-1,j}^u - 1 \right) \Delta y_j \left( \frac{\partial v}{\partial x} \right)_{i-1,j}, \quad (9.123)
\]

The shear stress in (9.100) and (9.101) can be formulated using Fig. 9.4

\[
\tilde{F}_h^x,_{shear} \bigg|_{i,j} = -\nu \left[ \Delta x_{ib}^{i,j} \partial u \left( \frac{\partial u}{\partial y} \right)_{i-1,j} + \Delta y_{ib}^{i,j} \partial u \left( \frac{\partial u}{\partial y} \right)_{i,j} \right], \quad (9.124)
\]

\[
\tilde{F}_h^y,_{shear} \bigg|_{i,j} = -\left( -\nu \right) \left[ \Delta y_{ib}^{i,j} \partial v \left( \frac{\partial v}{\partial x} \right)_{i-1,j} + \Delta y_{ib}^{i,j} \partial v \left( \frac{\partial v}{\partial x} \right)_{i,j} \right]. \quad (9.125)
\]

Now we arrived at the point, where it is possible to compare \( \tilde{F}_h^x,_{shear} \) with \( F_h^x,_{shear} \) and \( \tilde{F}_h^y,_{shear} \) with \( F_h^y,_{shear} \). By requiring that (9.122) is equal to (9.124) and (9.123) is equal to (9.125), the integration areas in the neighbourhood of the immersed boundary can be derived. These terms are defined as
$\Delta x_{i,j-1}^{ib,w} = (1 - \theta_{i,j}^v) \Delta x_i, \quad \Delta x_{i,j}^{ib,e} = 0,$
$\Delta y_{i,j}^{ib,n} = (1 - \theta_{i-1,j}^u) \Delta y_j, \quad \Delta y_{i,j}^{ib,s} = 0.$  \hfill (9.126)

Remember that we have to formulate $\Delta x_{i,j}^{n,w}$, which is defined in (9.74). The same holds
for $\Delta y_{i,j}^{n,n}$, which can be found in (9.79). Thus combining the derived integration areas
with the ‘normal’ integration areas (see (9.34), (9.50) ) leads to

$\Delta x_{i,j-1}^{n,w} = (1 - \theta_{i,j}^v) \Delta x_i + \frac{1}{2} \theta_{i,j}^v \Delta x_i, \quad \Delta x_{i,j}^{n,e} = 0 + \frac{1}{2} \theta_{i,j}^v \Delta x_i,$
$\Delta y_{i,j}^{w,n} = (1 - \theta_{i-1,j}^u) \Delta y_j + \frac{1}{2} \theta_{i,j}^u \Delta y_j, \quad \Delta y_{i,j}^{w,s} = 0 + \frac{1}{2} \theta_{i-1,j}^u \Delta y_j.$  \hfill (9.127)

and far away from the immersed boundary, the remaining integration areas are the same
as defined for the ‘normal’ fluid cell, see (9.34), (9.50) .

$\Delta x_{i,j}^{s,w} = \frac{1}{2} \theta_{i,j}^v \Delta x_i, \quad \Delta x_{i,j}^{s,e} = \frac{1}{2} \theta_{i,j-1}^v \Delta x_i,$
$\Delta y_{i-1,j}^{e,n} = \frac{1}{2} \theta_{i,j}^u \Delta y_j, \quad \Delta y_{i-1,j}^{e,s} = \frac{1}{2} \theta_{i,j}^u \Delta y_j.$  \hfill (9.128)

If we investigate all of these derived integration areas then it is interesting to know if these
values agree with what you would be expecting when looking at the corresponding figures.
It turns out that it does hold. Therefore, it is nice that this whole derivation supports our
intuition.
Chapter 10

Numerical results

In this section the numerical experiments are presented. The LS-STAG method will be validated against the theoretical solution. In all of the experiments water flows through a channel, whereby the channel is placed under different angles and also shifted a couple of times. The idea behinds this is that more and more different types of cut-cells will be used to build a geometry and resulting in better testing of the LS-STAG method.

10.1 Analytical

The parameters that will be used for the experiments are defined as follows. The dynamic viscosity, $\mu = 10 \text{Ns/m}^2$, kinematic viscosity $\nu = 0.01 \text{m}^2/\text{s}$, density $\rho = 1.0 \cdot 10^3 \text{kg/m}^3$. For the inflow, we use a parabolic velocity profile, whereby the average velocity, $u_{\text{avg}}$, is $1 \text{m/s}$ and the maximum velocity, $u_{\text{max}}$, is $1.5 \text{m/s}$. Using a parabolic velocity profile a Poiseuille flow will develop much faster, than when using a prescribe constant velocity. Another advantage of creating a Poiseuille flow is that the analytical outcome is known. So we can compare the numerical results of the LS-STAG method and also the ComFLOW method with it. The pressure difference between inlet and outlet, depending on the length of the channel, is given by

$$p_1 - p_2 = \frac{12\mu L}{\rho h^3} Q, \quad (10.1)$$

where $Q$ is the mass flux through the channel which can be computed as

$$Q = u_{\text{avg}} \rho h. \quad (10.2)$$

The pressure gradient can also be analytically determined and is formulated as

$$\frac{\partial p}{\partial x} = \frac{-12\mu u_{\text{avg}}}{h^2}. \quad (10.3)$$
The Reynolds number is given by \( Re = \frac{u_{avg} L}{\nu} \).

For the outflow we use a Neumann boundary condition for \( u \) and \( v \) and a Dirichlet boundary condition for \( p \). To be more precise, the pressure at the outflow will be set to zero.

10.2 Horizontal channel

10.2.1 Channel aligned with grid

The first test situation will be a channel aligned with the grid. The size of the opening for the inflow, \( h \), is set to 0.5 meter. \( p_1 - p_2 = 480 \), \( \frac{\partial p}{\partial x} = -480 \) and \( Re = 100 \). The length of the channel is 1 meter. The reason behind this is that this configuration is easy for comparison with ComFLOW. It looks like that for this situation only normal fluid cells are used, due to the fact that all \( \theta_{u,i,j} = 1 \) in the channel. However, the normal fluid cells are not really normal when they are adjacent to the immersed boundary, following the philosophy of Cheny et al. These cells are treated as a trapezoidal cut-cells. For instance, in the case when looking at the top of the channel, these cut-cells are south-trapezoidal cut-cells, see Fig. 9.2.

Taking a look at the test results for the velocity, see Fig. 10.1, it can be noticed that the horizontal velocity acts like you would expect it to be. The LS-STAG method and the ComFLOW method reach in the middle of the channel both the theoretical maximum velocity. Also the vertical velocity is almost zero. For the LS-STAG method the latter values are smaller than for ComFLOW. When considering the horizontal velocity profile halfway the channel, they both look like a normal parabola. However, in the endpoints of the velocity profile for LS-STAG a small kink appears. Maybe this has to do with the post-processing of Matlab, but it can also be something different. It is not clear at this moment, what is going on. From the pressure results, it is immediately clear that there are some big differences in the result of LS-STAG and ComFLOW. The method of Cheny et al. closely follows the analytical pressure, whereas ComFLOW does not. In Fig. 10.2 the pressure is computed at three different streamwise cross sections along the flow through the channel. Nevertheless all three have almost the same values and therefore they lie on each other. The reason why ComFLOW has different results in comparison with the LS-STAG method will be explained next.

The discretization of convection in the LS-STAG method is the same as the ComFLOW method, when considering non-moving boundaries. Therefore we investigate the discretization of the viscous fluxes. The ComFLOW method uses a central discretization
10.2. HORIZONTAL CHANNEL

(a) Horizontal velocity LS-STAG.

(b) Horizontal velocity ComFLOW.

(c) Vertical velocity LS-STAG.

(d) Vertical velocity ComFLOW.

(e) Horizontal velocity LS-STAG.

(f) Horizontal velocity ComFLOW.

Figure 10.1: Channel aligned with grid, mesh is 50 x 50.
(a) Pressure LS-STAG.

(b) Pressure ComFLOW.

(c) Pressure LS-STAG.

(d) Pressure ComFLOW.

Figure 10.2: Channel aligned with grid.
for diffusion

\[ V_{W,C}^u(i,j) = \frac{1}{2} \frac{\Delta x_i + \Delta x_{i+1}}{\Delta x_i} \Delta x_i, \]
\[ V_{E,C}^u(i,j) = \frac{1}{2} \frac{\Delta x_i + \Delta x_{i+1}}{\Delta x_{i+1}} \Delta x_{i+1}, \]
\[ V_{N,C}^u(i,j) = \frac{1}{2} \frac{\Delta y_j + \Delta y_{j+1}}{\Delta y_j} \Delta y_j, \]
\[ V_{S,C}^u(i,j) = \frac{1}{2} \frac{\Delta y_{j-1} + \Delta y_j}{\Delta y_j} \Delta y_j, \]
\[ V_{P,C}^u(i,j) = -V_{W,C}^u(i,j) - V_{E,C}^u(i,j) - V_{N,C}^u(i,j) - V_{S,C}^u(i,j). \]

LS-STAG uses a different discretization for diffusion, which is described in Chapter 9. To compare with the ComFLOW discretization, the coefficients for the viscous fluxes are multiplied with \( M^{-1} \). The mass matrix \( M \) is defined as

\[ M^x(i,j) = \frac{1}{2} V_{i,j} + \frac{1}{2} V_{i+1,j} = \frac{1}{2} \frac{(\Delta x_i \Delta y_j + \Delta x_{i+1} \Delta y_j)}{\Delta x_i} = \frac{1}{2} \frac{(\Delta x_i + \Delta x_{i+1}) \Delta y_j}{\Delta x_i}, \]
\[ M^y(i,j) = \frac{1}{2} V_{i,j} + \frac{1}{2} V_{i,j+1} = \frac{1}{2} \frac{(\Delta x_i \Delta y_j + \Delta x_i \Delta y_{j+1})}{\Delta y_j} = \frac{1}{2} \frac{(\Delta y_j + \Delta y_{j+1}) \Delta x_i}{\Delta y_j}, \]

for the \( x \)-direction and \( y \)-direction, respectively. Here \( V_{i,j} = \Delta x_i \Delta y_j \). The coefficients for the viscous fluxes multiplied with \( M^{-1} \) are denoted as \( V_{i,j}^u = M^{-1} K^u(i,j) \)

\[ V_{W,L}^u(i,j) = \frac{1}{2} \frac{\Delta x_i + \Delta x_{i+1}}{\Delta x_i} \Delta y_j \frac{\theta_{i,j}^u \Delta y_j}{V_{i,j}^u / \Delta y_j} \frac{\theta_{i-1,j}^u}{\Delta x_i} = \frac{\theta_{i,j}^u \theta_{i-1,j}^u}{\frac{1}{2} \frac{(\Delta x_i + \Delta x_{i+1}) \Delta x_i}{\Delta x_i}}, \]
\[ V_{E,L}^u(i,j) = \frac{1}{2} \frac{\Delta x_i + \Delta x_{i+1}}{\Delta x_{i+1}} \Delta y_j \frac{\theta_{i,j}^u \Delta y_j}{V_{i+1,j}^u / \Delta y_j} \frac{\theta_{i+1,j}^u}{\Delta x_{i+1}} = \frac{\theta_{i,j}^u \theta_{i+1,j}^u}{\Delta x_i + \Delta x_{i+1} \Delta x_{i+1}}, \]
\[ V_{N,L}^u(i,j) = \frac{1}{2} \frac{\Delta x_i + \Delta x_{i+1}}{\Delta x_{i+1}} \Delta y_j \frac{\theta_{i,j}^u \Delta y_j}{V_{n,e,i,j+1}^u + \Delta x_{i+1}^u} = \frac{\theta_{i,j}^u \theta_{i+1,j}^u \Delta x_i + \Delta x_{i+1}^u}{\Delta x_i + \Delta x_{i+1} \Delta x_{i+1}}, \]
\[ = \frac{\Delta x_i + \Delta x_{i+1}}{2} \Delta y_j \left( \frac{\theta_{i,j}^u \theta_{i+1,j}^u \Delta y_{j+1} + \frac{1}{2} \theta_{i,j}^u \Delta y_j}{\Delta x_i + \Delta x_{i+1}} \right). \]
The discretization for a south trapezoidal cut-cell is formulated in §9.3 and is given by

\[
\begin{align*}
\nu_{S,L}^u(i,j) &= \frac{1}{2} \left( \frac{\Delta x_{i,j}^{s,e}}{\Delta x_{i+1,j}^{s,w}} + \frac{\Delta x_{i,j}^{s,w}}{\Delta x_{i+1,j}^{s,e}} \right), \\
\nu_{W,L}^u(i,j) &= \frac{1}{2} \left( \frac{\Delta y_{i,j}^{s,w}}{\Delta y_{i,j+1}^{s,e}} + \frac{\Delta y_{i,j+1}^{s,e}}{\Delta y_{i,j}^{s,w}} \right), \\
\nu_{E,L}^u(i,j) &= \frac{1}{2} \frac{\Delta x_{i,j}^{s,e}}{\Delta x_{i+1,j}^{s,w}}, \\
\nu_{N,L}^u(i,j) &= \frac{1}{2} \frac{\Delta y_{i,j}^{s,w}}{\Delta y_{i,j+1}^{s,e}}, \\
\nu_{S,L}^u(i,j) &= \frac{1}{2} \left( \frac{\Delta y_{i,j}^{s}}{\Delta y_{i,j+1}^{s}} \right) - 1, \\
\nu_{P,L}^u(i,j) &= \frac{1}{2} \left( \frac{\Delta x_{i,j}^{s}}{\Delta x_{i+1,j}^{s}} \right) - 1
\end{align*}
\]

where (from (9.34))

\[
\begin{align*}
\Delta x_{i,j}^{n,e} &= \frac{1}{2} \theta_{i,j}^u \Delta x_{i,j}, \\
\Delta x_{i+1,j}^{n,w} &= \frac{1}{2} \theta_{i+1,j}^u \Delta x_{i+1,j}, \\
\Delta x_{i,j}^{s,e} &= \frac{1}{2} \theta_{i,j}^u \Delta x_{i,j}, \\
\Delta x_{i+1,j}^{s,w} &= \frac{1}{2} \theta_{i+1,j}^u \Delta x_{i+1,j}.
\end{align*}
\]

If we compare these formulas with the ones used for the discretization in ComFLOW, we see that they are the same when considering a normal fluid cell. Due to the fact that all apertures (cell fractions which are open to fluid) are equal to one. However, some coefficients for the viscous fluxes are different when considering a normal fluid cell adjacent to the immersed boundary. Actually, the normal fluid cell is then not really a normal fluid cell anymore, in the philosophy of Cheny et al.. This cell is treated as a south trapezoidal cut-cell, if we for instance consider a computational cell at the top of the channel, see Fig. 9.2. The discretization for a south trapezoidal cut-cell is formulated in §9.3 and is given by
where all the apertures are equal to one, except for $\theta_{i,j+1}^u$. This aperture is equal to zero. Here $\Delta x_{i,j}^{n,e}$ is replaced by $\Delta x_{i,j}^{ib,e}$ and $\Delta x_{i+1,j}^{n,w}$ is replaced by $\Delta x_{i+1,j}^{ib,w}$. The introduced integration areas are defined as, see (9.110)

$$\Delta x_{i,j}^{ib,e} = \frac{1}{2} \Delta x_i, \quad \Delta x_{i+1,j}^{ib,w} = \frac{1}{2} \Delta x_{i+1}.$$ 

If we again compare these formulas with the ones used for the discretization in ComFLOW, then it is clear that $V_{u,N,L}^n(i,j) \neq V_{u,N,C}^n(i,j)$ and $V_{u,P,L}^n(i,j) \neq V_{u,P,C}^n(i,j)$. That the coefficient $V_{u,N,\cdot}^n(i,j)$ is different is not a problem, because this coefficient will be multiplied with the horizontal velocity, $u_{i,j+1}$, in the discretization of the momentum equation. And $u_{i,j+1} = 0$, because it lies in the solid region and therefore this contribution is zero in the momentum equation. The real difference between LS-STAG and ComFLOW is the coefficient $V_{u,P,\cdot}^n(i,j)$. In this situation, $V_{P,L}^n(i,j) \neq -V_{W,L}^n(i,j) - V_{E,L}^n(i,j) - V_{N,L}^n(i,j) - V_{S,L}^n(i,j)$, whereas this still holds for the discretization in the ComFLOW method. The same situation arises when taking a look at the normal fluid cell (actually a north trapezoidal cut-cell) adjacent to the immersed boundary at the bottom of a channel. This is the reason why the LS-STAG results for the pressure are different compared to the pressure results from the ComFLOW method.

10.2.2 Aligned with grid and shifted half a cell

In the next test situation, the channel which was aligned with grid in the previous case is now shifted down by half a cell. This configuration is a better test example, because now we test the LS-STAG method with ‘real’ cut cells. All the features qua sizes and parameters from the previous case still hold for this situation.

When shifting the channel half a cell down, the results for the horizontal and vertical velocity are similar to the ones where the channel is aligned with the grid, see Fig. [10.3]. The LS-STAG methods shows much better results than ComFLOW. Considering the horizontal velocity profile, the difference is at the begin and endpoints of the parabola, where LS-STAG shows a kink and ComFLOW does not. When taking a look at the results for the pressure, which are plotted in Fig. [10.4] the LS-STAG methods performs just like in the case when the channel is not shifted. However, for the ComFLOW method, it even performs worse than in the previous case.

10.2.3 Grid refinement

For a grid refinement study, we use a channel which is aligned with the grid and also the same situation but then shifted a bit, see Fig. [10.5]. For both geometries, we start with a 10x10 mesh and end up with a 160x160 mesh.

The norm which will be used to determine the convergence rate of the velocity and of the pressure is $L^\infty$. This norm is defined like
Figure 10.3: Channel shifted a half cell down in comparison with previous test case, mesh is 50 x 50.
(a) Pressure LS-STAG.
(b) Pressure ComFLOW.
(c) Pressure LS-STAG.
(d) Pressure ComFLOW.

Figure 10.4: Channel shifted a half cell down in comparison with previous test case.
\[
L^\infty = \max_i |a_i - x_i|, \tag{10.4}
\]
where \(a_i\) is the analytical velocity/pressure and \(x_i\) is the computed velocity/pressure.

For the pressure gradient a difference will be used
\[
\frac{\partial p_a}{\partial x} - \frac{\partial p_c}{\partial x}, \tag{10.5}
\]
where \(\frac{\partial p_a}{\partial x}\) is the analytical pressure gradient and \(\frac{\partial p_c}{\partial x}\) is the computed pressure gradient. To determine the pressure gradient for the LS-STAG method as well as for the ComFLOW method, we have to do some extra work. First, we consider only the part where the pressure slope is more or less a straight line in Fig. 10.4. The values between 0.3m and 0.8m will be used for the determination of the pressure slope. Then for this part, the pressure gradient is computed using a least squares method. This computed pressure gradient is then used to compute the difference between the analytical and the computed pressure gradient.

In the case of the channel aligned with the grid, the LS-STAG method shows for the horizontal velocity, the pressure as well as for the pressure gradient a convergence rate of almost second order, see Fig. 10.5. When considering the ComFLOW method, it performs much worse. The convergence rate for the horizontal velocity, the pressure and also for the pressure gradient is around first order.

For a more complicated situation, the one where the channel is aligned with the grid and then shifted a bit, where 'real' apertures are used, both methods perform almost the same as in the previous test case, see Fig. 10.6. Thus the LS-STAG method shows nearly second order convergence rate for the velocity as well as for the pressure. And the ComFLOW method performs less, with first order convergence rate for the velocity as well as for the pressure. There were some stability problems with the LS-STAG method for computing this situation. The reason is that diffusion is iterated explicitly. Yet, by decreasing the time step, e.g. adjusting the CFL condition, the flow through the channel can be computed in a stable way.

10.3 Channel placed oblique to the grid

The position of inflow and outflow is now different compared to a horizontal channel. The inflow and outflow do not stand perpendicular with respect to the axis of a channel. Therefore the solution will not result in a perfect Poiseuille flow. Especially in the neighbourhood of the inlet and outlet some disturbance in the flow can be expected.

10.3.1 Channel placed under an angle of \(\frac{\pi}{4}\)

The next test case is one where the channel is placed oblique to the grid under an angle of \(\frac{\pi}{4}\). This situation is also not very complicated, because only triangular cells and normal fluid
10.3. CHANNEL PLACED OBLIQUE TO THE GRID

Figure 10.5: Channel aligned with the grid.
CHAPTER 10. NUMERICAL RESULTS

(a) Starting point

(b) Pressure convergence.

(c) Horizontal velocity convergence.

Figure 10.6: Channel shifted a bit in comparison with previous test case.
10.3. CHANNEL PLACED OBLIQUE TO THE GRID

cells are used for this construction. The height of the channel is \( h = \cos(\arctan(1/1)) \approx 0.4243\, \text{m} \). The length of the channel is \( L = \sqrt{1^2 + 1^2} \approx 1.414\, \text{m} \). The corresponding Reynolds number, \( Re \), is approximated as 141. Using these sizes the theoretical pressure difference between inlet and outlet can be calculated and is \( p_1 - p_2 \approx 942.81 \). The pressure gradient is analytically computed as \( \frac{\partial p}{\partial x} \approx -666.67 \).

The results for the horizontal and vertical velocity, shown in Fig. 10.7, are roughly in accordance with the values expected from the analytical solution. In this situation, the LS-STAG results do not show kinks any more in the endpoints of the velocity profile. The plots of the pressure, see Fig. 10.8 reveal that LS-STAG performs almost the same as ComFLOW. They both have a slope which deviates somewhat from the analytical slope. When using a finer grid, the LS-STAG results are depicted in Fig. 10.9, the computed pressure matches the analytical slope better. As we already predicted in the beginning of this section, Figs. 10.8 and 10.9 show some inflow and outflow effects.

10.3.2 Channel shifted half a cell upward

The channel is again placed oblique to the grid under an angle of \( \frac{\pi}{4} \), but now it is also shifted half a cell upward. This situation is more complicated than the previous situation, because now not only triangular cells and normal fluid cells are used, but also pentagonal cut-cells are required.

When we shift the channel from the previous situation with half a cell upward, the results for the velocity, see Fig. 10.10, remain roughly the same. Comparing Fig. 10.11 with 10.8 the pressure gradient is slightly worse for the LS-STAG method. ComFLOW seems to suffer a bit more from the shifted geometry. Again we have computed the flow through the channel on a finer grid. This time a grid of 110x209 is used. The computed pressure slope from the LS-STAG method, plotted in Fig. 10.12 match the theoretical pressure slope better.

10.3.3 Grid refinement

For a grid refinement study, we use a channel which is placed oblique to the grid under an angle of \( \frac{\pi}{4} \). For this geometry, we start with a 10x18 mesh and end up with a 160x288 mesh.

A difference will be used to determine the convergence rate of the pressure gradient, which is defined as

\[
\frac{\partial p_a}{\partial x} - \frac{\partial p_c}{\partial x},
\]

where \( \partial p_a/\partial x \) is the analytical pressure gradient and \( \partial p_c/\partial x \) is the computed pressure gradient. The pressure gradient from the LS-STAG method as well as from the ComFLOW method is computed in the same way as is has been done in §10.2.3. Thus, we consider
(a) Absolute velocity LS-STAG.  
(b) Absolute velocity ComFLOW.  
(c) Horizontal velocity LS-STAG.  
(d) Horizontal velocity ComFLOW.  
(e) Vertical velocity LS-STAG.  
(f) Vertical velocity ComFLOW.  

Figure 10.7: Channel placed oblique to the grid under an angle of $\frac{\pi}{4}$, mesh is 50 x 90.
10.3. CHANNEL PLACED OBLIQUE TO THE GRID

Figure 10.8: Channel placed oblique to the grid under an angle of $\frac{\pi}{4}$.

Figure 10.9: Channel placed oblique to the grid under an angle of $\frac{\pi}{4}$, mesh is 100x180.
(a) Absolute velocity LS-STAG.  
(b) Absolute velocity ComFLOW.  
(c) Horizontal velocity LS-STAG.  
(d) Horizontal velocity ComFLOW.  
(e) Vertical velocity LS-STAG.  
(f) Vertical velocity ComFLOW.

Figure 10.10: Channel shifted a half cell up in comparison with previous test case, mesh is 50 x 95.
10.3. CHANNEL PLACED OBLIQUE TO THE GRID

Figure 10.11: Channel shifted a half cell up in comparison with previous test case.

Figure 10.12: Channel shifted a half cell up in comparison with previous test case, mesh is 110x209.
only the part where the pressure slope is more or less a straight line in Fig. 10.8. In this situation we use the values of the pressure between 0.4m and 1.0m.

![Pressure convergence](image)

Figure 10.13: Pressure gradient convergence.

The convergence rates for the pressure gradient from the LS-STAG method as well as from the ComFLOW method are first order, see Fig. 10.13. This is quite a surprise for the LS-STAG method, because the convergence rate has dropped from second order to first order in comparison with the situation where the channel was placed horizontal. The reason for this effect is under investigation.

10.3.4 Channel placed under an angle of arctan $\frac{1}{2}$

This time, the channel is placed oblique to the grid under an angle of arctan $\frac{1}{2}$. For this configuration, 4 types of cut-cells are needed: north-east/south-west triangular cut-cell and south/north trapezoidal cut-cells. In complexity, it is similar to the situation in the previous test case. The height of opening perpendicular to the channel is $h = \cos(\arctan(0.5/1)) \approx 0.5367m$ and the length $L = \sqrt{1^2 + 0.5^2} \approx 1.118m$. The corresponding Reynolds number can be calculated, $Re \approx 112$. The theoretical pressure difference between the left and right of the channel is computed as $p_1 - p_2 \approx 465.85$. The pressure gradient, $\frac{\partial p}{\partial x}$, is approximated as $-416.67$.

First we begin with investigating the horizontal and vertical velocity, which are shown in Fig. 10.14. There is not much difference in the results between LS-STAG and ComFLOW. They both show some kinks in the velocity profiles. When considering the pressure gradient, Fig. 10.15 shows that the LS-STAG slope is almost equal to the analytical slope. However, the ComFLOW method gives a slope which is clearly less than the theoretical slope.

10.3.5 Channel slightly shifted

The final test situation is the one, where a channel is placed oblique to grid under an angle of arctan $\frac{1}{2}$ and then shifted up half a cell. This test example is the most complicated
10.3. CHANNEL PLACED OBLIQUE TO THE GRID

(a) Absolute velocity LS-STAG.  
(b) Absolute velocity ComFLOW.

(c) Horizontal velocity LS-STAG.  
(d) Horizontal velocity ComFLOW.

(e) Vertical velocity LS-STAG.  
(f) Vertical velocity ComFLOW.

Figure 10.14: Channel placed oblique to the grid under an angle of $\tan^{-1}\frac{1}{2}$, mesh is 50 x 65.
(a) Pressure LS-STAG.

(b) Pressure ComFLOW.

(c) Pressure LS-STAG.

(d) Pressure ComFLOW.

Figure 10.15: Channel placed oblique to the grid under an angle of $\tan^{-1}\frac{1}{2}$. 
one. All types of cut-cells are used, i.e. triangular, trapezoidal and pentagonal cut-cells are required for building this configuration. Further, all features qua sizes and parameters are the same as the ones computed in the previous situation.

The numerical results for the horizontal and vertical velocity, depicted in Fig. 10.16, are almost the same as in the previous case. Also the pressure results show similarly, when comparing Fig. 10.17 with Fig. 10.15. Again, the LS-STAG method predicts the pressure gradient fairly well, whereas the ComFLOW results are way off.

10.3.6 Grid refinement

For a channel which is placed oblique to the grid, we do another grid refinement study. This time, the channel is placed under an angle of \( \arctan \frac{\pi}{2} \). For such a geometry, we start with a 10x13 mesh and end up with a 160x208 mesh.

Again a difference will be used to determine the convergence rate of the pressure gradient. This difference is defined as

\[
\frac{\partial p_a}{\partial x} - \frac{\partial p_c}{\partial x},
\]

where \( \frac{\partial p_a}{\partial x} \) is the analytical pressure gradient and \( \frac{\partial p_c}{\partial x} \) is the computed pressure gradient. The pressure gradient from the LS-STAG method as well as from the ComFLOW method is computed in the same way as has been done in §10.2.3. Thus, we consider only the part where the pressure slope is more or less a straight line in Fig. 10.15. For this case, we take the values of the pressure between 0.3 m and 0.75 m.

In Fig. 10.18 the convergence rate for the pressure gradient is shown. For the LS-STAG method this is almost first order. Again this is a bit disappointing, because the convergence rate has dropped from second order to first order when compared to the situation where the channel was placed horizontal. The reason for causing this effect is still unknown. When considering the ComFLOW method, it is even worse than the LS-STAG method. This time, the convergence rate is less than first order.
Figure 10.16: Channel shifted a half cell up in comparison with previous test case, mesh is 50 x 70.
10.3. CHANNEL PLACED OBLIQUE TO THE GRID

(a) Pressure LS-STAG.
(b) Pressure ComFLOW.

(c) Pressure LS-STAG.
(d) Pressure ComFLOW.

Figure 10.17: Channel shifted a half cell up in comparison with previous test case, mesh is 50 x 70.

Figure 10.18: Pressure gradient convergence.
Chapter 11

Discussion and conclusion

11.1 Discussion

We have tested the LS-STAG method with 6 different geometries, two horizontal channels and four channels placed oblique to the grid. The reason for choosing a channel is that a Poiseuille flow can be generated and therefore analytical solutions for the velocity and pressure are available. The results for the horizontal and vertical velocity of the LS-STAG method matches in all situations the analytical solutions. The pressure gradient has also been computed for these configurations, and in most of the situations the LS-STAG method closely follows the theoretical pressure slope.

For two horizontal channels, a grid refinement study has been done. In both situations, the convergence rate of the velocity and of the pressure shows second order for the LS-STAG method. Bear in mind that this is a simple test case. Therefore another grid refinement study has been made and this time a channel placed oblique to the grid, under two different angles, has been investigated. For both configurations, the convergence rate for the pressure (gradient) drops to first order. The reason why this has happened is still unknown.

The discretization of the diffusive term by Cheny et al. differs at certain points from the central discretization used in ComFLOW. This difference appears in all test situations; systematically the LS-STAG method gives better results than the ComFLOW method.

For computing the flow through a channel with small apertures leading to small cut-cell volumes, some stability problems arise. The reason, diffusion is explicitly solved and this can result in a stringent time step limit. By decreasing the time step, e.g. adjusting the CFL condition, the LS-STAG can be made stable again. Thus, explicitly discretizing the diffusion can result in more computing time, but this is not a problem here. Our purpose is to compute a flow through a channel and then analyse the numerical results. If you want to compute it more efficiently, then the discretization of diffusion can be solved implicitly.
11.2 Conclusion and further action

From the results we may conclude that the LS-STAG method performs in almost every situation better than the ComFLOW method. Thus the discretization of the LS-STAG method looks a very promising method for improving the discretization in the existing ComFLOW method. Nevertheless, further research is required, such as

A study is needed to understand why the convergence rate for the pressure (gradient) is first order, in the situation where the channel is placed oblique to the grid.

The LS-STAG method has to be extended with the other 6 types of cut-cells. Then every (complex) geometry in two dimensions can be tested.

Research must be done to included moving bodies.

Configurations with high Reynolds number flows must be simulated with the LS-STAG method, to test whether the method can handle these flows or not.

The LS-STAG method must be tested if it can work properly with stretched grids.

The LS-STAG method has to be extended to three dimensions.
Bibliography


